

# 云存储开发人员手册

版本：v1.4

日期：2016-11-13

中国移动通信集团公司

总页数	52	正文	47	附录		生效日期	
编制				批准			

## 变更履历

修改编号	版本	修改内容	修改人	修改日期
1	v1.0	编制	李宏杰	2016-08-22
2	v1.1	全面纠错	李宏杰	2016-08-24
3	v1.2	调整章节顺序	李宏杰	2016-08.30
4	V1.3	删除管理员开发接口和2.4.1.2更新	余礼杨	2016-10.12
5	V1.4	添加云存储Rest API接口地址	董林鹏	2016-11.13

# 目 录

云存储开发人员手册.....	I
<b>1 概述.....</b>	<b>1</b>
1.1 产品简介与系统架构.....	1
1.2 oNest REST API 与本手册概览.....	2
1.3 oNest REST API 接口地址.....	2
<b>2 S3 API.....</b>	<b>3</b>
2.1 功能支持.....	3
2.2 一般实体.....	4
2.2.1 桶与主机名.....	4
2.2.2 一般请求头部.....	4
2.2.3 一般响应状态.....	5
2.3 认证与 ACLs.....	6
2.3.1 认证.....	6
2.3.2 访问控制列表（ACLs）.....	8
2.4 桶操作.....	8
2.4.1 桶的基本操作.....	8
2.4.1.1 列出桶.....	8
2.4.1.2 创建桶.....	9
2.4.1.3 删除桶.....	10
2.4.1.4 获取桶.....	10
2.4.1.5 获取桶位置.....	12
2.4.2 桶的 ACL 操作.....	13
2.4.2.1 获取桶的 ACL.....	13
2.4.2.2 创建桶的 ACL.....	14
2.4.3 桶的多块上传.....	15
2.4.3.1 列出桶的多块上传.....	15
2.4.4 桶的多版本.....	17
2.4.4.1 启用或挂起桶的版本功能.....	17
2.5 对象操作.....	18

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

2.5.1 对象的基本操作.....	18
2.5.1.1 创建对象.....	18
2.5.1.2 复制对象.....	19
2.5.1.3 移除对象.....	20
2.5.1.4 获取对象.....	21
2.5.1.5 获取对象信息.....	22
2.5.2 对象的 ACL 操作.....	23
2.5.2.1 获取对象 ACL.....	23
2.5.2.2 设置对象 ACL.....	23
2.5.3 对象的多块上传.....	24
2.5.3.1 初始化多块上传.....	24
2.5.3.2 多块上传部分.....	25
2.5.3.3 完成多块上传.....	26
2.5.3.4 放弃多块上传.....	28
2.6 SDKs.....	28
<b>3 SWIFT API.....</b>	<b>29</b>
<b>3.1 功能支持.....</b>	<b>29</b>
<b>3.2 认证.....</b>	<b>30</b>
3.2.1 认证操作.....	30
3.2.1.1 获取认证.....	30
<b>3.3 容器操作.....</b>	<b>31</b>
3.3.1 基本容器操作.....	32
3.3.1.1 列出所有容器.....	32
3.3.1.2 新建容器.....	33
3.3.1.3 列出容器内的对象.....	34
3.3.1.4 添加/更新容器元数据.....	36
3.3.1.5 删除一个容器.....	37
3.3.2 容器的 ACLs 操作.....	37
3.3.2.1 更新一个容器的 ACLs.....	37
<b>3.4 对象操作.....</b>	<b>38</b>
3.4.1 基本对象操作.....	39
3.4.1.1 创建/更新对象.....	39
3.4.1.2 复制对象.....	40
3.4.1.3 删除对象.....	41

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

---

3.4.1.4 获取对象.....	42
3.4.2 对象元数据操作.....	43
3.4.2.1 获取对象元数据.....	43
3.4.2.2 增加/更新对象元数据.....	43
<b>3.5 临时 URL 操作.....</b>	<b>44</b>
3.5.1 临时 URL 基本操作.....	44
3.5.1.1 POST TEMP-URL 密钥.....	44
3.5.1.2 获取 TEMP-URL 对象.....	45
<b>3.6 SDKs.....</b>	<b>46</b>

# 1 概述

## 1.1 产品简介与系统架构

oNest 是一个用于存储和管理海量对象的存储系统，其 6.0 版本是基于业界主流分布式文件系统 ceph 设计，设计的宗旨是为了给用户提供可扩展，高可靠，高性能的云存储平台，在任何时候从网络的任何地方存储和获取任何大小的数据；oNest 在 ceph 的基础上针对实际生产环境的应用进行了功能扩展，如智能部署、存储管理功能完善、性能调优等，充分满足企业未来业务需求，帮助其 IT 系统转型以便更快更好地应对日益激烈的竞争环境，实现与客户的共同成长。

oNest 由智能分布式存储（RADOS）集群、oNest 管理平台和驻留在应用服务系统中的程序或代码库组成，如图 1.1 所示：

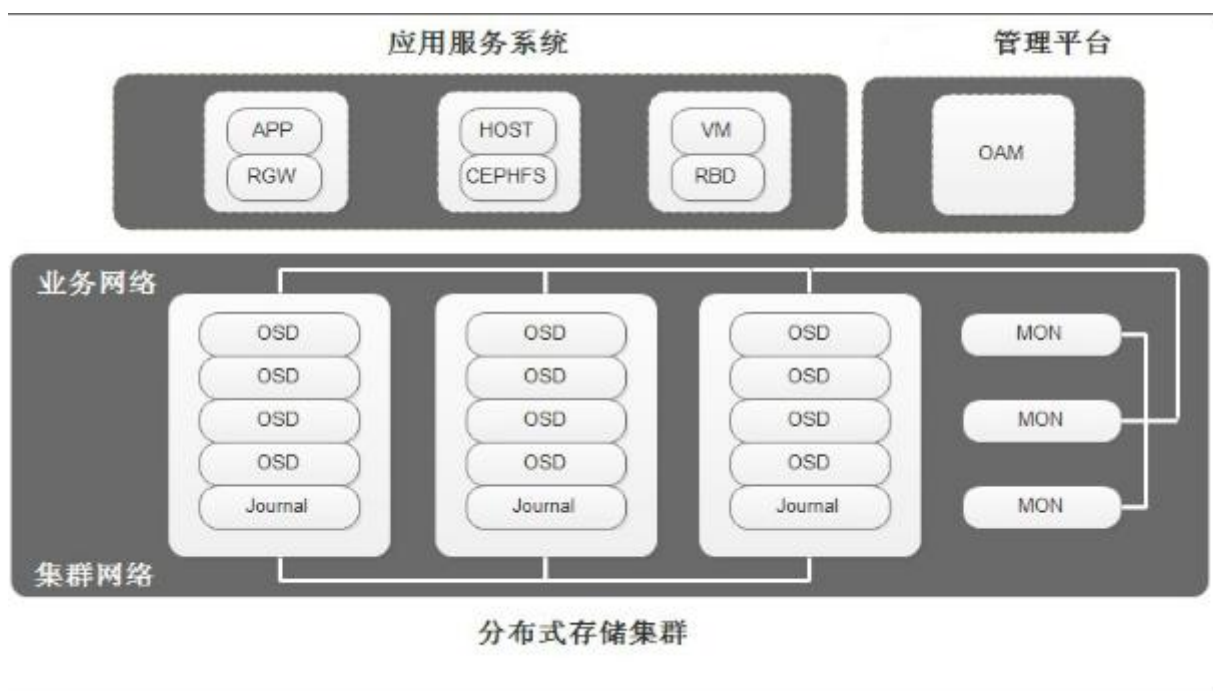


图 1.1 oNest 系统架构

其中智能分布式存储集群包含对象存储 OSD 和监控管理 MON 组件：

**OSD:** OSD 负责处理客户端读写请求。不管是通过哪种方式（对象、文件或者其他通过调用 librados 接口的 API）访问分布式存储集群，数据最后都转换成对象存储到集群中。一个本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

OSD 对应一个存储磁盘。为了提高性能，每个 OSD 需要一个 SSD 盘或 SSD 盘分区作为其日志盘。一个集群可根据容量、性能、故障域需要，部署多个 OSD 字段。

**MON:** MON 负责处理集群管理元数据表。客户端向 OSD 发送读写请求之前，需要从 MON 中拿到最新的集群表，利用 CRUSH 算法计算出来数据存储位置后，直接将读写请求发送至对应的 OSD。一个存储集群可以只包含 1 个 MON 字段，但是考虑到 MON 字段失效后集群不可用的情况，实际生产环境中一个存储集群至少部署 3 个 MON 字段。

**RGW:** 对象存储网关，给用户提供了 REST API，并且兼容 S3 和 Swift。

## 1.2 oNest REST API 与本手册概览

oNest 提供 REST 风格的 API 供开发者使用，这些 API 能够让你从 Web 端随时随地存储、获取任意数据。

本手册包含三大章节，即兼容 Amazon S3 的 API，兼容 Swift 的 API，以及管理 API 三个部分。手册描述了如何进行各项管理操作，如何使用兼容 S3 和 Swift 的 API 管理你的资源。手册同时还描述了访问控制和认证过程。

本手册各小节采用统一形式编写，层次清晰，逻辑分明，便于开发者参阅手册进行开发。

## 1.3 oNest REST API 接口地址

云存储公网 Rest API 接口地址：

eos-beijing-1.cmecloud.cn

云存储内网 Rest API 接口地址：

112.35.22.101 或 112.35.22.102

## 2 S3 API

oNest 支持 RESTful API，兼容 Amazon S3 API 的基本数据访问模型。

### 2.1 功能支持

下表描述了目前所支持的 Amazon S3 功能。

功能	支持	备注
列出桶	√	
删除桶	√	
创建桶	√	标准 ACLs 的不同设置
桶生命周期	<b>X</b>	
桶与对象策略	<b>X</b>	支持 ACLs
桶网站	<b>X</b>	
桶 ACLs (Get, Put)	√	标准 ACLs 的不同设置
桶位置	√	
桶通知	<b>X</b>	
桶对象多版本	√	
获取桶信息 (HEAD)	√	
桶申请方付款	√	
创建对象	√	
删除对象	√	
获取对象	√	
对象 ACLs (Get, Put)	√	
获取对象信息 (HEAD)	√	
POST 对象	√	
复制对象	√	
多块上传	√	(不具备复制功能)

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。



下表描述了目前还不支持的请求头部。

字段	类型
x-amz-security-token	请求
Server	响应
x-amz-delete-marker	响应
x-amz-id-2	响应
x-amz-version-id	响应

## 2.2 一般实体

### 2.2.1 桶与主机名

这里有两种不同的访问桶的模式。第一种（推荐）方法将桶标识为 **URI** 的高层目录。

```
GET /mybucket HTTP/1.1
```

```
Host: cname.domain.com
```

第二种方法通过虚拟的桶主机名表示桶。例如：

```
GET / HTTP/1.1
```

```
Host: mybucket.cname.domain.com
```

提示：我们推荐第一种方法，因为第二种方法需要昂贵的域名认证和通配的 **DNS**。

### 2.2.2 一般请求头部

请求头部	描述
<b>CONTENT_LENGTH</b>	请求部分的长度
<b>DATE</b>	请求日期与时间（UTC）

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

HOST	主机名
AUTHORIZATION	认证令牌

### 2.2.3 一般响应状态

状态码	状态
100	Continue
200	Success
201	Created
202	Accepted
204	NoContent
206	Partial content
304	NotModified
400	InvalidArgument
400	InvalidDigest
400	BadDigest
400	InvalidBucketName
400	InvalidObjectName
400	UnresolvableGrantByEmailAddress
400	InvalidPart
400	InvalidPartOrder
400	RequestTimeout
400	EntityTooLarge
403	AccessDenied
403	UserSuspended
403	RequestTimeTooSkewed
404	NoSuchKey

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

404	NoSuchBucket
404	NoSuchUpload
405	MethodNotAllowed
408	RequestTimeout
409	BucketAlreadyExists
409	BucketNotEmpty
411	MissingContentLength
412	PreconditionFailed
416	InvalidRange
422	UnprocessableEntity
500	InternalError

## 2.3 认证与 ACLs

向 RADOS Gateway (RGW) 发起的请求可以是已认证或未认证的。RGW 假设未认证的请求是匿名用户发送的。RGW 支持标准 ACLs。

### 2.3.1 认证

请求认证需要在发送消息到 RGW 前，在请求中包括一个 **access key** 与一个基于哈希的消息认证码 (HMAC)。RGW 使用一个兼容 S3 的认证方法。下面是一个范例。

```
HTTP/1.1
```

```
PUT /buckets/bucket/object.mpeg
```

```
Host: cname.domain.com
```

```
Date: Mon, 2 Jan 2012 00:01:01 +0000
```

```
Content-Encoding: mpeg
```

```
Content-Length: 9999999
```

```
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

在例子中，使用你的 **access key** 填入 `{access-key}` 中，将头部字符串和 **secret key** 作为输入计算出哈希值，填入 `{hash-of-header-and-secret}` 中。

为了生成哈希值，你必须：

- 获得头部字符串；
- 将请求头部字符串转为标准形式；
- 使用 SHA-1 算法生成 HMAC；（可参考 RFC 2104 和维基百科“HMAC”条目）
- base-64 编码 HMAC。

为了将请求头部字符串转为标准形式：

- 获取所有以 `x-amz-` 开头的字段；
- 确保所有字段都是小写字母；
- 按照字典序对字段进行排序；
- 将同样字段的所有实例拼接成一个字段，并且使用逗号分隔开；
- 使用一个空格替代所有的换行与多个空格；
- 移除冒号后的空格；
- 在每一字段后面都加上一行空行；
- 将处理好的内容融回头部。

将 base-64 编码后的 HMAC 字符串填入 `{hash-of-header-and-secret}` 中。

## 2.3.2 访问控制列表 (ACLs)

RGW 支持兼容 S3 的 ACL 功能。ACL 是一个访问授权列表，指定了一个用户能够对一个桶或者一个用户使用什么操作。每一种授权在应用到桶上和对象上都会有不同的含义。

权限	桶	对象
READ	被授权者可以列出桶内的对象	被授权者可以读取对象
WRITE	被授权者可以写或删除桶内的对象	无
READ_ACP	被授权者可以读取桶 ACL	被授权者可以读取对象 ACL
WRITE_ACP	被授权者可以写桶 ACL	被授权者可以写对象 ACL
FULL_CONTROL	被授权者拥有对象与桶的所有权限	被授权者可以读写对象 ACL

## 2.4 桶操作

### 2.4.1 桶的基本操作

#### 2.4.1.1 列出桶

**GET** / 返回发出请求的用户的桶列表。 **GET** / 只返回一个已认证的用户创建的桶。你不能发出一个匿名请求。

##### ➤ 语法

```
GET / HTTP/1.1
```

```
Host: cname.domain.com
```

```
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

##### ➤ 响应实体

字段	描述	类型
Buckets	包含桶列表的容器	Container

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

Bucket	包含桶信息的容器	Container
Name	桶的名字	String
CreationDate	桶被创建的 UTC 时间	Date
ListAllMyBucketsResult	一个包括结果的容器	Container
Owner	一个包含桶拥有者 ID 和 DisplayName 的容器	Container
ID	桶拥有者的 ID	String
DisplayName	桶拥有者的 display name	String

### 2.4.1.2 创建桶

创建一个新的桶。为了创建一个桶，你必须拥有一个 user ID 和一个合法的 AWS Access Key ID 来认证请求。你不能以匿名用户的身份创建桶。

**注意：在目前的释出版本中，我们并不支持 PUT /{bucket} 请求的请求实体**

#### ➤ 限制

一般情况下，桶的名字必须遵循以下限制：

- 桶名字必须唯一；
- 桶名字必须以小写字母开始和结束；
- 桶名字可以包括一个短横（-）。

#### ➤ 语法

```
PUT /{bucket} HTTP/1.1
```

```
Host: cname.domain.com
```

```
x-amz-acl: public-read-write
```

```
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

## ➤ 参数

**x-amz-acl**

描述： 标准 ACL

合法值： private, public-read, public-read-write, authenticated-read

必需： 否

## ➤ HTTP 响应消息

如果一个桶名字是唯一的，并且没有被使用，操作会成功。如果桶的名字已经存在并且用户是该重名桶的拥有者，操作也会成功。如果桶的名字已经存在并且被使用，操作将会失败。

状态码	状态	描述
409	BucketAlreadyExists	桶已被其它用户拥有

**2.4.1.3 删除桶**

删除一个桶。成功删除后你可以重新使用该桶名。

## ➤ 语法

DELETE /{bucket} HTTP/1.1

Host: cname.domain.com

Authorization: AWS {access-key}:{hash-of-header-and-secret}

## ➤ HTTP 响应消息

状态码	状态	描述
204	No Content	桶已被移除

**2.4.1.4 获取桶**

返回一个桶中对象的列表。

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

## ➤ 语法

```
GET /{bucket}?max-keys=25 HTTP/1.1
```

```
Host: cname.domain.com
```

## ➤ 参数

**prefix**

描述： 只返回包含指定前缀的对象

类型： String

**delimiter**

描述： 前缀和对象名剩余部分之间的分隔符

类型： String

**marker**

描述： 返回的对象列表的起始索引

类型： String

**max-keys**

描述： 最大返回的关键字的数量。默认为 1000

类型： Integer

## ➤ HTTP 响应消息

状态码	状态	描述
200	OK	已获取桶

## ➤ 桶的响应实体

GET /{bucket} 返回一个包含桶的相关信息的容器。

字段	描述	类型
ListBucketResult	包含对象列表的容器	Entity
Name	将要返回内容的桶的名字	String

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。



<b>Prefix</b>	对象的关键字的前缀	String
<b>Marker</b>	返回的对象列表的起始索引	String
<b>MaxKeys</b>	返回的最大关键字数量	Integer
<b>Delimiter</b>	如设置，拥有相同前缀的对象将会在 <b>CommonPrefixes</b> 列表中出现	String
<b>IsTruncated</b>	如为真，则只有桶的内容的子集被返回	Boolean
<b>CommonPrefixes</b>	如果多个对象拥有相同前缀，这里将会出现	Container

### ➤ 对象的响应实体

**ListBucketResult** 字段中包含的对象，每个对象都会被装在 **Contents** 容器内。

字段	描述	类型
<b>Contents</b>	包含对象的容器	Object
<b>Key</b>	对象的密钥	String
<b>LastModified</b>	对象的最后修改日期/时间	Date
<b>ETag</b>	对象的 MD5 值（实体标签）	String
<b>Size</b>	对象大小	Integer
<b>StorageClass</b>	应永远返回 <b>STANDARD</b>	String

### 2.4.1.5 获取桶位置

获取桶所处区域。用户必须是桶的拥有者才能执行此操作。一个桶可以通过在 **PUT** 请求中提供 **LocationConstraint** 参数来限定在一个区域内。

### ➤ 语法

通过下面的例子为桶资源加入 **location** 子资源。

```
GET /{bucket}?location HTTP/1.1
```

```
Host: cname.domain.com
```

```
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

➤ 响应实体

字段	描述	类型
LocationConstraint	桶所处的区域，默认区域则为空字符串	String

## 2.4.2 桶的 ACL 操作

### 2.4.2.1 获取桶的 ACL

获取桶的访问权限控制列表。用户必须是桶的拥有者或被授权了该桶的 **READ\_ACP** 权限。

➤ 语法

通过下面的例子为桶资源加入 **acl** 子资源。

```
GET /{bucket}?acl HTTP/1.1
```

```
Host: cname.domain.com
```

```
Authorization: AWS {access-key}:{hash-of-header-and-secret}
```

➤ 响应实体

字段	描述	类型
AccessControlPolicy	包含响应信息的容器	Container
AccessControlList	包含 ACL 信息的容器	Container
Owner	包含桶拥有者用户 ID 与 DisplayName 的容器	Container

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

ID	桶拥有者的 ID	String
DisplayName	桶拥有者的 display name	String
Grant	包含 Grantee 和 Permission 的容器	Container
Grantee	包含接受授权的用户的 ID 与 DisplayName 的容器	Container
Permission	给予 Grantee 的桶的权限	String

### 2.4.2.2 创建桶的 ACL

设置访问一个已经存在的桶的权限。用户必须是桶的拥有者或被授予了该桶的

WRITE\_ACP 权限。

#### ➤ 语法

通过下面的例子为桶资源加入 acl 子资源。

```
PUT /{bucket}?acl HTTP/1.1
```

#### ➤ 响应实体

字段	描述	类型
AccessControlPolicy	包含响应信息的容器	Container
AccessControlList	包含 ACL 信息的容器	Container
Owner	包含桶拥有者用户 ID 与 DisplayName 的容器	Container
ID	桶拥有者的 ID	String
DisplayName	桶拥有者的 display name	String
Grant	包含 Grantee 和 Permission 的容器	Container
Grantee	包含接受授权的用户的 ID 与 DisplayName 的容器	Container
Permission	给予 Grantee 的桶的权限	String

## 2.4.3 桶的多块上传

### 2.4.3.1 列出桶的多块上传

`GET /?uploads` 返回一个当前正在进行的多块上传的列表。例如，应用初始化了一个多块上传，但是当前服务并没有完成所有上传。

#### ➤ 语法

```
GET /{bucket}?uploads HTTP/1.1
```

#### ➤ 参数

你可以为 `GET /{bucket}?uploads` 指定参数，但这之中并没有必填项。

##### prefix

**描述：** 返回一个正在进行的，关键字包含指定前缀的上传

**类型：** String

##### delimiter

**描述：** 前缀与其余对象名之间的分隔符

**类型：** String

##### key-marker

**描述：** 上传列表的起始标记

**类型：** String

##### max-keys

**描述：** 最大返回的正在上传的任务的数量。默认为 1000

**类型：** Integer

##### max-uploads

**描述：** 最大的多块上传数量，范围为 1-1000。默认值为 1000

**类型：** Integer

##### upload-id-marker

**描述：** 如果 `key-marker` 没有被指定，则可以忽略该值。指定第一个上传的 ID

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

或基于 ID 按照字典序列出。

类型: String

### ➤ 响应实体

字段	描述	类型
ListMultipartUploadsResult	包含结果的容器	Container
ListMultipartUploadsResult.Prefix	prefix 请求参数所指定的前缀（如有）	String
Bucket	将接受桶内容的桶	String
KeyMarker	key-marker 请求参数所指定的 key marker （如有）	String
UploadIdMarker	upload-id-marker 请求参数所指定的 marker（如有）	String
NextKeyMarker	用于随后的请求的 key 标记 （如果 IsTruncated 值为 True）	String
NextUploadIdMarker	用于随后的请求的上传 ID 标记 （如果 IsTruncated 值为 True）	String
MaxUploads	max-uploads 参数指定的最大上传数	Integer
Delimiter	如设置，则拥有相同前缀的对象将会出现 在 CommonPrefixes 列表中	String
IsTruncated	如为真，则只返回桶上传的内容的子集	Boolean
Upload	包含 Key, UploadId, InitiatorOwner, StorageClass, Initiated 元素的容器	Container
Key	当多块上传完成时对象的密钥	String
UploadId	标识多块上传的 ID	String
Initiator	包含初始化上传的用户的 ID 和 DisplayName 的容器	Container
DisplayName	初始化上传的用户的 display name	String

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

<b>ID</b>	初始化上传的用户的 ID	String
<b>Owner</b>	包含拥有上传好的对象的用户的 <b>ID</b> 和 <b>DisplayName</b> 的容器	Container
<b>StorageClass</b>	用于存储对象的方法，值为 <b>STANDARD</b> 或 <b>REDUCED_REDUNDANCY</b>	String
<b>Initiated</b>	用户初始化上传的日期和时间	Date
<b>CommonPrefixes</b>	若多个对象拥有相同前缀，则它们将会出现在这里	Container
<b>CommonPrefixes.Prefix</b>	在 <b>prefix</b> 请求参数指定前缀后被定义的密钥的子字符串	String

## 2.4.4 桶的多版本

### 2.4.4.1 启用或挂起桶的版本功能

**PUT /?versioning** 这一子资源设置一个已有的桶的版本控制状态。执行此操作必须是该桶的拥有者。你可以使用以下其中之一值来设置版本控制状态：

- Enabled**：启用桶中的对象的版本控制，所有在桶内的对象都会获得一个唯一的版本 ID；
- Suspended**：禁用桶内的对象的版本控制，所有加入该桶的对象将会收到一个空的版本 ID。

如果一个桶的版本控制状态从未被设定，则没有版本控制状态，一个 **GET** 版本请求将不会返回任何版本 ID 值。

## ➤ 语法

```
PUT /{bucket}?versioning HTTP/1.1
```

## ➤ 请求实体

**VersioningConfiguration**

描述： 请求容器  
类型： Container

**Status**

描述： 桶的版本控制状态。合法值：Suspended/Enabled  
类型： String

## 2.5 对象操作

### 2.5.1 对象的基本操作

#### 2.5.1.1 创建对象

为一个桶创建一个对象。你必须拥有桶的写权限。

## ➤ 语法

```
PUT /{bucket}/{object} HTTP/1.1
```

## ➤ 请求头部

**content-md5**

描述： 信令的一个 base64 编码的 MD5 值  
合法值： String，无默认值或限制值  
必需： 否

**content-type**

描述： 一个标准的 MIME 类型

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

合法值: 任何 MIME 类型。默认值: `binary/octet-stream`

必需: 否

#### **x-amz-meta-`<...>`**

描述: 用户元数据, 与对象存储在一起

合法值: 最大为 8kb 的 String。无默认值

必需: 否

#### **x-amz-acl**

描述: 一个标准 ACL

合法值: `private, public-read, public-read-write, authenticated-read`

必需: 否

### 2.5.1.2 复制对象

使用 PUT 并且指定一个目标桶和对象名来复制一个对象。

#### ➤ 语法

```
PUT /{dest-bucket}/{dest-object} HTTP/1.1
```

```
x-amz-copy-source: {source-bucket}/{source-object}
```

#### ➤ 请求头部

##### **x-amz-copy-source**

描述: 源的桶名字与对象名字

合法值: `{bucket}{obj}`

必需: 是

##### **x-amz-copy-if-modified-since**

描述: 仅在指定时刻后发生修改才复制

合法值: Timestamp

必需: 否

本文档的程序或内容受版权法的保护, 未经中国移动通信集团公司的书面许可, 不得擅自泄漏、拷贝或复制本文档资料的全部或部分。



### x-amz-copy-if-unmodified-since

描述： 仅在指定时刻后未发生修改才复制

合法值： Timestamp

必需： 否

### x-amz-copy-if-match

描述： 仅在对象 ETag 与给定 ETag 相符时复制

合法值： 实体标签（ETag）

必需： 否

### x-amz-copy-if-none-match

描述： 仅在对象 ETag 与给定 ETag 不相符时复制

合法值： 实体标签（ETag）

必需： 否

### x-amz-acl

描述： 一个标准 ACL

合法值： private, public-read, public-read-write, authenticated-read

必需： 否

#### ➤ 响应实体

字段	描述	类型
CopyObjectResult	包含响应元素的容器	Container
LastModified	源对象最后修改日期	Date
ETag	新对象的 ETag	String

### 2.5.1.3 移除对象

移除一个对象，需要拥有对包含该对象的桶的 WRITE 权限。

## ➤ 语法

```
DELETE /{bucket}/{object} HTTP/1.1
```

#### 2.5.1.4 获取对象

从 RADOS 中的桶获取一个对象。

## ➤ 语法

```
GET /{bucket}/{object} HTTP/1.1
```

## ➤ 请求头部

**range**

描述： 需要获取的对象的范围  
合法值： Range: bytes=beginbyte-endbyte  
必需： 否

**if-modified-since**

描述： 仅在指定时刻后发生修改才获取  
合法值： Timestamp  
必需： 否

**if-unmodified-since**

描述： 仅在指定时刻后未发生修改才获取  
合法值： Timestamp  
必需： 否

**if-match**

描述： 仅在对象 ETag 与给定 ETag 相符时获取  
合法值： 实体标签（ETag）  
必需： 否

**if-none-match**

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

**描述:** 仅在对象 ETag 与给定 ETag 不相符时获取

**合法值:** 实体标签 (ETag)

**必需:** 否

➤ 响应实体

字段	描述	类型
Content-Range	数据范围，如果范围头部在请求中被指定则返回该值	Container

### 2.5.1.5 获取对象信息

返回对象信息。这个请求将会返回与 Get Object 请求相同的信息头部，但仅会包括元数据，载荷不含对象实体。

➤ 语法

```
HEAD /{bucket}/{object} HTTP/1.1
```

➤ 请求头部

#### range

**描述:** 需要获取的对象的范围

**合法值:** Range: bytes=beginbyte-endbyte

**必需:** 否

#### if-modified-since

**描述:** 仅在指定时刻后发生修改才获取

**合法值:** Timestamp

**必需:** 否

#### if-unmodified-since

**描述:** 仅在指定时刻后未发生修改才获取

**合法值:** Timestamp

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

必需： 否

#### if-match

描述： 仅在对象 ETag 与给定 ETag 相符时获取

合法值： 实体标签（ETag）

必需： 否

#### if-none-match

描述： 仅在对象 ETag 与给定 ETag 不相符时获取

合法值： 实体标签（ETag）

必需： 否

## 2.5.2 对象的 ACL 操作

### 2.5.2.1 获取对象 ACL

#### ➤ 语法

```
GET /{bucket}/{object}?acl HTTP/1.1
```

#### ➤ 响应实体

字段	描述	类型
AccessControlPolicy	包含响应信息的容器	Container
AccessControlList	包含 ACL 信息的容器	Container
Owner	包含对象拥有者的 ID 与 DisplayName 的容器	Container
ID	对象拥有者的 ID	String
DisplayName	对象拥有者的 display name	String
Grant	包含 Grantee 与 Permission 的容器	Container
Grantee	包含接受授权的用户 ID 与 DisplayName 的容器	Container
Permission	给予 Grantee 对象的权限	String

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

### 2.5.2.2 设置对象 ACL

#### ➤ 语法

```
PUT /{bucket}/{object}?acl
```

#### ➤ 响应实体

字段	描述	类型
AccessControlPolicy	包含响应信息的容器	Container
AccessControlList	包含 ACL 信息的容器	Container
Owner	包含对象拥有者的 ID 与 DisplayName 的容器	Container
ID	对象拥有者的 ID	String
DisplayName	对象拥有者的 display name	String
Grant	包含 Grantee 与 Permission 的容器	Container
Grantee	包含接受授权的用户的 ID 与 DisplayName 的容器	Container
Permission	给予 Grantee 对象的权限	String

### 2.5.3 对象的多块上传

#### 2.5.3.1 初始化多块上传

初始化一个多块上传进程。

#### ➤ 语法

```
POST /{bucket}/{object}?uploads
```

#### ➤ 请求头部

##### content-md5

描述： 信令的一个 base64 编码的 MD5 值

合法值： String，无默认值或限制值

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

必需： 否

### content-type

描述： 一个标准的 MIME 类型

合法值： 任何 MIME 类型。默认值： `binary/octet-stream`

必需： 否

### x-amz-meta-`<...>`

描述： 用户元数据，与对象存储在一起

合法值： 最大为 8kb 的 String。无默认值

必需： 否

### x-amz-acl

描述： 一个标准 ACL

合法值： `private, public-read, public-read-write, authenticated-read`

必需： 否

### ➤ 响应实体

字段	描述	类型
<code>InitiatedMultipartUploadsResult</code>	包含响应信息的容器	Container
<code>Bucket</code>	将要接受对象的桶	String
<code>Key</code>	请求中指定的 key（如果有）	String
<code>UploadID</code>	<code>upload-id</code> 请求参数所指定的标识多块上传的 ID（如果有）	String

### 2.5.3.2 多块上传部分

#### ➤ 语法

```
GET /{bucket}/{object}?uploadId=123 HTTP/1.1
```

#### ➤ 响应实体

字段	描述	类型
ListPartsResult	包含响应信息的容器	Container
Bucket	将要接受对象的桶	String
Key	请求中指定的 <b>key</b> （如果有）	String
UploadID	upload-id 请求参数所指定的标识多块上传的 ID（如果有）	String
Initiator	包含初始化上传的用户的 <b>ID</b> 和 <b>DisplayName</b>	Container
ID	初始化上传的用户的 ID	String
DisplayName	初始化上传的用户的 display name	String
Owner	包含拥有上传的对象的用户的 <b>ID</b> 和 <b>DisplayName</b> 的容器	Container
StorageClass	用于存储对象的方法， <b>STANDARD</b> 或 <b>REDUCED_REDUNDANCY</b>	String
PartNumberMarker	若 <b>IsTruncated</b> 参数为 <b>true</b> ，则这是用在接下来的请求的分块部分标记，在列表的前面	String
NextPartNumberMarker	若 <b>IsTruncated</b> 参数为 <b>true</b> ，则这是用在接下来的请求的下一个分块部分标记，在列表的尾部	String
MaxParts	<b>max-parts</b> 请求参数指定的响应中允许的最大分块数	Integer
IsTruncated	若为 <b>true</b> ，则只返回对象上传的内容的子集	Boolean
Part	一个包含 <b>Key</b> , <b>Part</b> , <b>InitiatorOwner</b> , <b>StorageClass</b> ,	Container

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

	Initiated 元素的容器	
PartNumber	分块的 ID	Integer
ETag	分块的实体标签	String
Size	上传分块的大小	Integer

### 2.5.3.3 完成多块上传

组合上传分块并创建新对象，从而完成多块上传。

#### ➤ 语法

```
POST /{bucket}/{object}?uploadId= HTTP/1.1
```

#### ➤ 请求实体

##### CompleteMultipartUpload

描述： 包含一个或多个分块的容器

类型： Container

必需： 是

##### Part

描述： 一个包含 PartNumber 和 ETag 的容器

类型： Container

必需： 是

##### PartNumber

描述： 分块的 ID

类型： Integer

必需： 是

##### ETag

描述： 分块的实体标签

类型： String

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。



必需： 是

➤ 响应实体

字段	描述	类型
CompleteMultipartUploadResult	包含响应信息的容器	Container
Location	新对象的 URI	URI
Bucket	包含新对象的桶的名字	String
Key	对象的密钥	String
ETag	新对象的实体标签	String

#### 2.5.3.4 放弃多块上传

➤ 语法

```
DELETE /{bucket}/{object}?uploadId= HTTP/1.1
```

## 2.6 SDKs

API 提供多种语言 SDK，目前支持 C++，C#，Python，Java，Perl，PHP，Ruby。

详见 SDK 相关文档。

## 3 Swift API

oNest 支持 RESTful API，兼容 Swift API 的基本数据访问模型。

### 3.1 功能支持

下表描述了目前所支持的 Swift 功能。

功能	支持	备注
认证	√	
获取账户元数据	√	
Swift ACLs	√	支持 Swift ACLs 的子集
列出容器	√	
删除容器	√	
创建容器	√	
获取容器元数据	√	
更新容器元数据	√	
删除容器元数据	√	
列出对象	√	
静态网站	<b>X</b>	
创建对象	√	
创建大对象	√	
删除对象	√	
获取对象	√	
复制对象	√	
获取对象元数据	√	
更新对象元数据	√	
对象生命周期	√	
对象多版本	√	

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

CORS	<b>X</b>	
------	----------	--

## 3.2 认证

需要认证的 Swift API 请求必须在请求头里带上 **X-Storage-Token** 认证令牌。此令牌可以从 RADOS 网关、或别的认证器获取，要从 RADOS 网关获取的话需创建用户，例如：

```
sudo radosgw-admin user create --uid="{username}" --display-name="{Display Name}"
```

### 3.2.1 认证操作

#### 3.2.1.1 获取认证

要对用户进行身份认证，需要构建一个请求头中有 **X-Auth-User** 和 **X-Auth-Key** 的请求。

##### ➤ 语法

```
GET /auth HTTP/1.1
Host: swift.radosgwhost.com
X-Auth-User: johndoe
X-Auth-Key: R7UUOLFDI2ZI9PRCQ53K
```

##### ➤ 请求头部

#### **X-Auth-User**

**描述：** 用来认证的 RADOS 网关用户的用户名

**类型：** String

**必需：** 是

#### **X-Auth-Key**

**描述：** RADOS 网关用户的密钥

**类型：** String

**必需：** 是

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

### ➤ 响应头部

服务器的响应应该包含一个 **X-Auth-Token** 值。响应也可能包含一个 **X-Storage-Url**，它提供 **{api version}/{account}** 前缀，这是在整个 **Swift API** 文档的其它请求中指定的前缀。

字段	描述	类型
<b>X-Storage-Token</b>	在请求中 <b>X-Auth-User</b> 指定的用户的认证令牌	String
<b>X-Storage-Url</b>	URL 和为用户准备的 <b>{api version}/{account}</b> 路径	String

一个典型的响应消息如下：

```
HTTP/1.1 204 No Content
Date: Mon, 16 Jul 2012 11:05:33 GMT
Server: swift
X-Storage-Url: https://swift.radosgwhost.com/v1/ACCT-12345
X-Auth-Token: UOICCC8TahFKIWuv9DB09TWHF0nDjpPEIha0kAa
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

## 3.3 容器操作

要从我们的兼容 **Swift** 的服务中获取数据，你可以执行 **GET** 请求，在身份验证期间使用 **X-Storage-Url** 的值来获取。

一个容器是一种用来存储数据对象的机制。一个帐户可以有很多容器，但容器名称必须是唯一的。本 **API** 允许客户端创建一个容器，设置访问控制和元数据，检索一个容器的内容和删除一个容器。由于这个 **API** 的请求涉及到一个特定用户的帐户相关信息，因此在这个 **API** 中的所有请求都必须经过身份验证，除非一个容器的访问控制故意设置为公开访问（即允许匿名的请求）。

**Amazon S3 API** 使用 ‘bucket’ 这个词来描述一个数据容器。当你听到有人在 **Swift API**

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

中提到一个‘bucket’时，‘bucket’这个词可以被理解为相当于术语‘container’。

对象存储的一个特点就是它不支持分层路径或目录。相反，它支持由一个或者多个容器组成一个层级，其中每个容器包含若干对象。RADOS 网关的兼容 Swift API 支持

‘pseudo-hierarchical containers’的概念，就是一个使用对象的命名方式来模拟一个容器（或目录）层次以替代在存储系统具体实现。你可以使用 pseudo-hierarchical 名字来给对象命名（比如：photos/buildings/empire-state.jpg），但容器的名字不能包含一个正斜杠（/）字符。

### 3.3.1 基本容器操作

#### 3.3.1.1 列出所有容器

一个指定 API 版本和帐户的 GET 请求将返回特定 用户帐户的容器列表。因为这个请求返回一个特定用户的容器，所以该请求需要一个身份验证令牌。这种请求不能匿名。

##### ➤ 语法

```
GET /{api version}/{account} HTTP/1.1  
Host: {fqdn}  
X-Auth-Token: {auth-token}
```

##### ➤ 请求参数

###### limit

描述： 限制特定值的结果的数目  
类型： integer  
必需： 否

###### format

描述： 定义结果的格式  
类型： String  
有效值： json | xml  
必需： 否

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

**marker**

**描述:** 返回大于 **marker** 值的结果的列表

**类型:** String

**必需:** 否

**➤ 响应实体**

响应包含容器列表，或者返回一个 **204 HTTP** 响应代码。

字段	描述	类型
<b>account</b>	账户信息的列表	Container
<b>container</b>	容器列表	Container
<b>name</b>	容器名	String
<b>bytes</b>	容器空间大小	Integer

**3.3.1.2 新建容器**

要创建一个新容器，需要构建一个包含 **API** 版本、账户和新容器的名称的 **PUT** 请求。容器名称必须是唯一的，不能包含斜杠 (/) 字符，且应该小于 **256** 字节。在请求头信息中你也可以包括访问控制和元数据等。这个操作是幂等的，也就是说，如果你构建一个请求来创建一个已经存在的容器，它将返回一个 **HTTP 202** 返回代码，但不会创建另一个容器。

**➤ 语法**

```
PUT /{api version}/{account}/{container} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

```
X-Container-Read: {comma-separated-uids}
```

```
X-Container-Write: {comma-separated-uids}
```

```
X-Container-Meta-{key}: {value}
```

## ➤ 请求参数

### X-Container-Read

**描述：** 有该容器的读权限的用户的 IDs  
**类型：** 逗号分隔的用户 ID 字符串值  
**必需：** 否

### X-Container-Write

**描述：** 有该容器的写权限的用户的 IDs  
**类型：** 逗号分隔的用户 ID 字符串值  
**必需：** 否

### X-Container-Meta-{key}

**描述：** 用户定义元数据的 **key**，一个任意的字符串值  
**类型：** String  
**必需：** 否

## ➤ 响应实体

如果具有相同名称的容器已经存在，并且该用户是该容器的所有者，则这个操作会成功。  
否则操作将会失败。

状态码	状态	描述
409	BucketAlreadyExists	容器已经存在，且归一个不同的用户的所有

### 3.3.1.3 列出容器内的对象

要列出一个容器内的所有对象，需要构建一个包含 API 版本、账户和容器的名称的 **GET** 请求。您可以指定查询参数来过滤完整列表，或置空该参数以返回该容器内存储的前 10000 个对象的名称的列表。

## ➤ 语法

```
GET /{api version}/{container} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

## ➤ 请求参数

**format**

描述： 定义结果的格式

类型： String

有效值： json | xml

必需： 否

**prefix**

描述： 限制所有结果的对象名以这个指定的前缀开始

类型： String

必需： 否

**marker**

描述： 返回的结果的列表大于 marker 的值

类型： String

必需： 否

**limit**

描述： 限制返回的结果的列表为这个指定的值

类型： Integer

有效值： 0 - 10,000

必需： 否

**delimiter**

描述： 前缀和剩余对象名之间的分隔符

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。



类型: String

必需: 否

### path

描述: 对象的 pseudo-hierarchical 路径

类型: String

必需: 否

### ➤ 响应实体

字段	描述	类型
container	容器	Container
object	容器内的对象	Container
name	容器内的对象的名称	String
hash	对象内容的哈希码	String
last_modified	对象内容的最近修改时间	Date
content_type	对象内容的类型	String

#### 3.3.1.4 添加/更新容器元数据

要向容器添加元数据，需要构建一个包含 API 版本、账户和容器的名称的 POST 请求。你必须有对你想添加或者更新元数据的容器的写权限。

#### ➤ 语法

```
POST /{api version}/{account}/{container} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

```
X-Container-Meta-Color: red
```

```
X-Container-Meta-Taste: salty
```

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

### ➤ 请求头部

#### **X-Container-Meta-{key}**

**描述：** 用户定义的元数据的密钥，一个任意的字符串值

**类型：** String

**必需：** 否

### 3.3.1.5 删除一个容器

要删除一个容器，需要构建一个包含 API 版本、账户和容器的名称的 **DELETE** 请求。容器必须是空的。如果你想检查容器是否为空，可以对容器执行一个 **HEAD** 请求。一旦你确认成功删除了容器，就可以重用容器的名字。

### ➤ 语法

```
DELETE /{api version}/{account}/{container} HTTP/1.1  
Host: {fqdn}  
X-Auth-Token: {auth-token}
```

### ➤ 响应实体

状态码	状态	描述
204	NoContent	容器已经被删除

## 3.3.2 容器的 ACLs 操作

### 3.3.2.1 更新一个容器的 ACLs

当用户创建一个容器的时候，默认情况下用户有读和写这个容器的权限。要允许其他用户读一个容器的内容或者写入对象到一个容器，必须专门为该用户启用相关权限。你也可以指定 **X-Container-Read** 或 **X-Container-Write** 的值为 \*，这样可以有效地使所有用户都可以读或写这个容器。设置 \* 使得容器变为公开，即允许匿名用户读写容器。

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

## ➤ 语法

```
POST /{api version}/{account}/{container} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

```
X-Container-Read: *
```

```
X-Container-Write: {uid1}, {uid2}, {uid3}
```

## ➤ 请求头部

**X-Container-Read**

**描述:** 有该容器的读权限的用户的 IDs

**类型:** 逗号分隔的用户 ID 字符串值

**必需:** 否

**X-Container-Write**

**描述:** 有该容器的写权限的用户的 IDs

**类型:** 逗号分隔的用户 ID 字符串值

**必需:** 否

### 3.4 对象操作

对象是一个用于存储数据和元数据的容器。一个容器可以包含很多对象，但是对象的名字必须唯一。这个 API 允许客户端创建对象、设置访问权限及元数据、读取对象的数据和元数据、以及删除对象。因为此 API 发出的请求是与用户帐户信息相关的，所以此 API 内的所有请求都必须经过认证，除非容器或对象的访问控制权限被故意设置成了可公开访问（即允许匿名请求）。

### 3.4.1 基本对象操作

#### 3.4.1.1 创建/更新对象

要创建新对象，需发送带有 API 版本、帐户、容器名和新对象名称的 **PUT** 请求。还必须  
有对应容器的写权限才能创建或更新对象。容器内的对象名字必须唯一。**PUT** 请求不会预先  
检测名字唯一性，若使用已有名字，此请求就会更新该对象。你可以在对象名中用  
**pseudo-hierarchical** 来区分同名但位于不同 **pseudo-hierarchical** 目录的对象。你可以在请  
求头里加上访问控制和元数据信息。

##### ➤ 语法

```
PUT /{api version}/{account}/{container}/{object} HTTP/1.1  
Host: {fqdn}  
X-Auth-Token: {auth-token}
```

##### ➤ 请求头部

###### **ETag**

**描述：** 对象内容的 **MD5** 哈希值，建议设置  
**类型：** String  
**必需：** 否

###### **Content-Type**

**描述：** 对象所包含内容的类型  
**类型：** String  
**必需：** 否

###### **Transfer-Encoding**

**描述：** 用于标识此对象是否是更大的聚合对象的一部分  
**类型：** String  
**有效值：** **chunked**  
**必需：** 否

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

### 3.4.1.2 复制对象

对象复制方法允许你在服务器端创建对象副本，这样你就不必先下载再上传到另一个容器了。要把一对象的内容复制到另一对象，你可以发送包含 **API** 版本、帐户和容器名的 **PUT** 请求或 **COPY** 请求。发送 **PUT** 请求时，请求内容为目标容器和对象名。源容器和源对象名在请求头里设置。对于 **COPY** 请求，请求内容为源容器和源对象名，目标容器和对象在请求头里。要复制对象，你必须有写权限，目标对象名在其容器内也必须唯一。该请求不会预先检查名字唯一性，若使用已有名字，它就会更新重名对象。你可以在对象名中用 **pseudo-hierarchical** 语法来区分同名但位于不同 **pseudo-hierarchical** 目录的对象。你可以在请求头里加上访问控制和元数据头。

#### ➤ 语法

```
PUT /{api version}/{account}/{dest-container}/{dest-object} HTTP/1.1
X-Copy-From: {source-container}/{source-object}
Host: {fqdn}
X-Auth-Token: {auth-token}
```

或：

```
COPY /{api version}/{account}/{source-container}/{source-object} HTTP/1.1
Destination: {dest-container}/{dest-object}
```

#### ➤ 请求头部

##### **X-Copy-From**

**描述：** 用于在 **PUT** 请求中定义源容器或对象的路径

**类型：** String

**必需：** 使用 **PUT** 方法时必需

##### **Destination**

**描述：** 用于在 **COPY** 请求中定义目标容器或对象的路径

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄露、拷贝或复制本文档资料的全部或部分。

类型: String  
必需: 使用 COPY 方法时必需

#### If-Modified-Since

描述: 若从源对象的 last\_modified 属性记录的时间起修改过, 那就复制  
类型: Date  
必需: 否

#### If-Unmodified-Since

描述: 若从源对象的 last\_modified 属性记录的时间起未修改过, 那就复制  
类型: Date  
必需: 否

#### Copy-If-Match

描述: 请求中的 ETag 与源对象的 ETag 属性相同时才复制  
类型: ETag  
必需: 否

#### Copy-If-None-Match

描述: 请求中的 ETag 与源对象的 ETag 属性不相同时才复制  
类型: ETag  
必需: 否

### 3.4.1.3 删除对象

要删除对象, 可发送带有 API 版本、帐户、容器和对象名的 DELETE 请求。此帐户必须有容器的写权限才能删除其内的对象。成功删除对象后, 你就能重用对象名了。

## ➤ 语法

```
DELETE /{api version}/{account}/{container}/{object} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

### 3.4.1.4 获取对象

要获取一对象，需发出带有 API 版本、帐户、容器和对象名的 **GET** 请求，而且必须有此容器的读权限才能读取其内的对象。

## ➤ 语法

```
GET /{api version}/{account}/{container}/{object} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

## ➤ 请求头部

**range**

**描述：** 要获取某一对象内容的一部分，你可以指定字节范围

**类型：** Range: bytes=beginbyte-endbyte

**必需：** 否

**If-Modified-Since**

**描述：** 若从源对象的 **last\_modified** 属性记录的时间起修改过，那就复制

**类型：** Date

**必需：** 否

**If-Unmodified-Since**

**描述：** 若从源对象的 **last\_modified** 属性记录的时间起未修改过，那就复制

**类型：** Date

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

必需： 否

### Copy-If-Match

描述： 请求中的 ETag 与源对象的 ETag 属性相同时才复制

类型： ETag

必需： 否

### Copy-If-None-Match

描述： 请求中的 ETag 与源对象的 ETag 属性不相同才复制

类型： ETag

必需： 否

#### ➤ 响应实体

字段	描述	类型
Content-Range	此区间表示对象内容的子集。只有在请求头中有 range 字段时才会返回此字段。	Range

## 3.4.2 对象元数据操作

### 3.4.2.1 获取对象元数据

要查看一对象的元数据，可发送带有 API 版本、帐户、容器和对象名的 HEAD 头。你还必须有此容器的读权限才能读取其内对象的元数据。此请求会返回和获取对象本身时相同的头信息，只是不返回对象的数据而已。

#### ➤ 语法

```
HEAD /{api version}/{account}/{container}/{object} HTTP/1.1
Host: {fqdn}
X-Auth-Token: {auth-token}
```



### 3.4.2.2 增加/更新对象元数据

要给对象增加元数据需发送带有 API 版本、帐户、容器和对象名的 **POST** 请求。你还必须有父容器的写权限才能增加或更新元数据。

#### ➤ 语法

```
POST /{api version}/{account}/{container}/{object} HTTP/1.1
Host: {fqdn}
X-Auth-Token: {auth-token}
```

#### ➤ 请求头部

##### **X-Object-Meta-{key}**

**描述:** 一个用户定义的元数据关键字，其值为任意字符串

**类型:** String

**必需:** 否

## 3.5 临时 URL 操作

为做到无需共享凭据也能临时允许访问对象（比如用于 GET 请求），radosgw 末端的 swift 也支持临时 URL 功能。要用此功能，需设置 X-Account-Meta-Temp-URL-Key 或可选项 X-Account-Meta-Temp-URL-Key-2 的初始值。Temp URL 功能需要这些密钥的 HMAC-SHA1 签名。

### 3.5.1 临时 URL 基本操作

#### 3.5.1.1 POST TEMP-URL 密钥

向 swift 帐户发送一个带有所需 Key 的 **POST** 请求，就能给此帐户设置临时 URL 私钥，这样就可向其他帐户提供临时 URL 访问了。最多支持两个密钥，两个密钥的签名都要检查，如果没问题，密钥就可以继续使用，临时 URL 也不会失效。

本文档的程序或内容受版权法的保护，未经中国移动通信集团公司的书面许可，不得擅自泄漏、拷贝或复制本文档资料的全部或部分。

## ➤ 语法

```
POST /{api version}/{account} HTTP/1.1
```

```
Host: {fqdn}
```

```
X-Auth-Token: {auth-token}
```

## ➤ 请求头部

**X-Account-Meta-Temp-URL-Key**

描述： 用户定义的密钥，可以是任意字符串

类型： String

必需： 是

**X-Account-Meta-Temp-URL-Key-2**

描述： 用户定义的密钥，可以是任意字符串

类型： String

必需： 否

**3.5.1.2 获取 TEMP-URL 对象**

临时 URL 使用 HMAC-SHA1 签名，它包含下列要素：

- a. 请求方法的值，如 “GET” ；
- b. 过期时间，格式秒数，如 Unix 时间
- c. 从 “v1” 起的请求路径

上述条目将用新行来进行格式化，并使用之前上传的临时 URL 密钥通过 SHA-1 哈希算法生成一个 HMAC。

下面的 Python 脚本演示了上述过程：

```
import hmac
from hashlib import sha1
from time import time

method = 'GET'
host = 'https://objectstore.example.com'
duration_in_seconds = 300 # Duration for which the url is valid
expires = int(time() + duration_in_seconds)
path = '/v1/your-bucket/your-object'
key = 'secret'
hmac_body = '%s\n%s\n%s' % (method, expires, path)
hmac_body = hmac.new(key, hmac_body, sha1).hexdigest()
sig = hmac.new(key, hmac_body, sha1).hexdigest()
rest_uri =
"{host}{path}?temp_url_sig={sig}&temp_url_expires={expires}".format(
    host=host, path=path, sig=sig, expires=expires)
print rest_uri

# Example Output
#
https://objectstore.example.com/v1/your-bucket/your-object?temp_url_sig=ff465
7876227fc6025f04fcf1e82818266d022c6&temp_url_expires=1423200992
```

### 3.6 SDKs

SDK 示例遵循一个简单的基于容器的对象的生命周期。第一步为创建你的客户端与 RADOS 网关服务器的连接。随后你可以根据一个自然地容器和对象的生命周期进行，包括增加和获取对象元数据。如图 4.1 所示。

示例代码包括下列语言：Java, Python, Ruby。详见 SDK 相关文档。

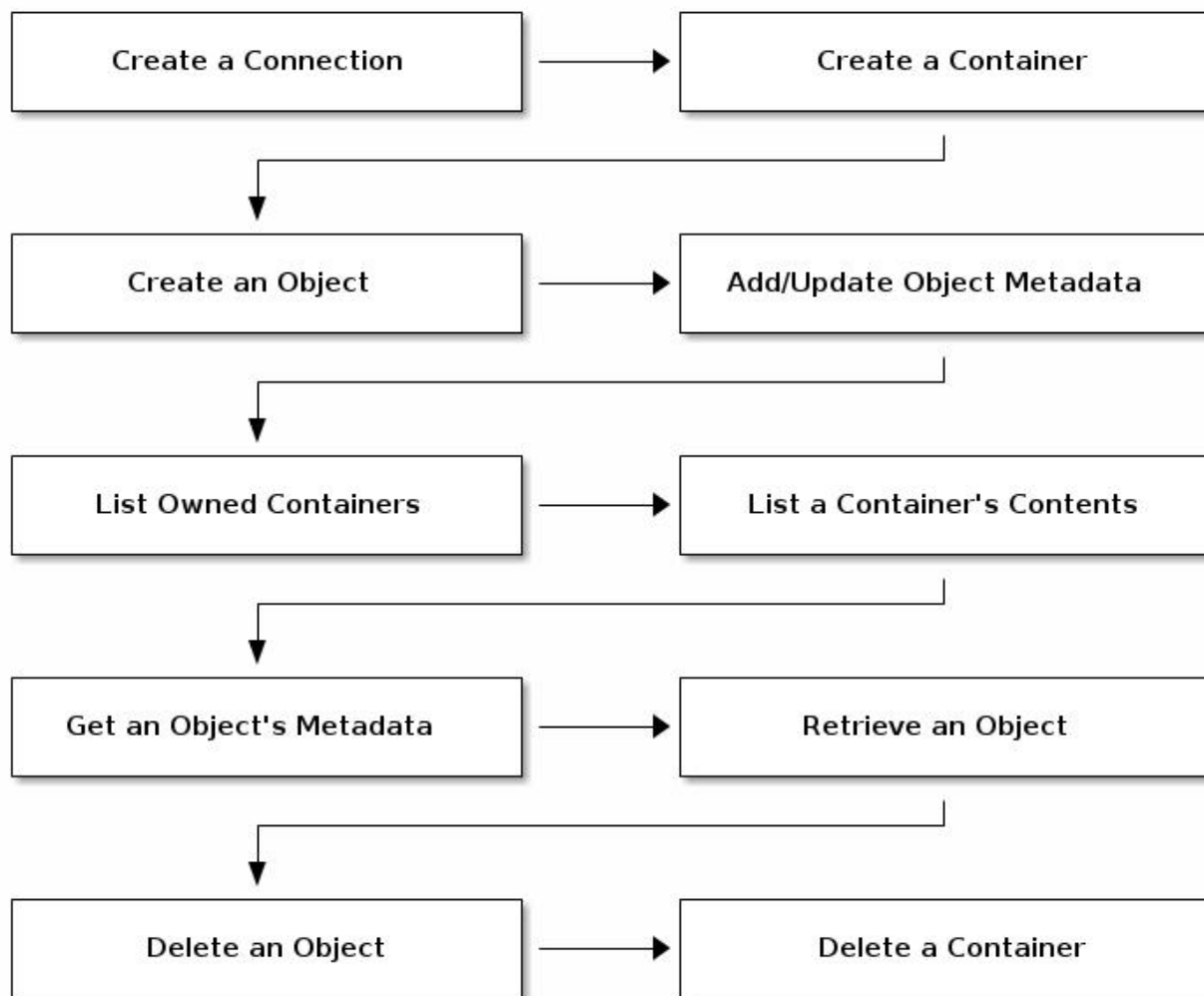


图 3.1 基于容器的对象的生命周期