

# 七牛云对象存储使用指南

## 1 对象存储服务简介

七牛云海量存储系统（KODO）是自主研发的非结构化数据存储管理平台，支持中心和边缘存储。平台经过多年大规模用户验证已跻身先进技术行列，并广泛应用于海量数据管理的各类场景。



详细介绍参考官网：<https://www.qiniu.com/products/kodo>

### 1.1 产品优势

#### 1. 高可靠

业界领先的纠删码存储方案，能够提供高达 11 个 9 的数据可靠性。跨数据中心的副本冗余，能够保障服务的超高可用性。

#### 2. 低成本

无需前期投入。七牛云对象存储按需使用、按需付费的便捷性，能够有效避免存储及带宽资源的闲置浪费。

#### 3. 存储加速

边缘存储可充分利用可用链路带宽，数据在边缘节点上传和下载可平均提速 60% 以上。

#### 4. 易扩展

利用七牛云对象存储，您的存储空间无上限的同时也无需担心扩容问题。您能够实现存储需求的弹性伸缩，从而提高业务灵活性。

#### 5. 数据智能化

与七牛云其他产品紧密协同，提供标准 HDFS 访问方式，为大数据和机器学习的海量高速读写场景进行了大量优化。

#### 6. 边缘计算

就近集成边缘计算及边缘缓存服务，边缘存储节点具备本地数据处理能力。

### 1.2 核心功能及服务

#### 1. 多媒体数据处理

基于对象存储，您可以一站式地完成图片处理和音视频处理

## 2. 镜像存储

镜像存储适用于迁移原有业务系统的已有数据。可以帮助用户实现无缝数据迁移，迁移过程中并不影响原有业务系统的访问。

## 3. 上传/下载

针对不同的上传/下载场景，七牛云提供了丰富的API接口和工具供用户使用，同时支持服务端上传和客户端直传，并提供加速服务。

## 4. 灵活部署

同时支持在七牛云边缘节点和客户侧边缘节点部署边缘存储服务。

## 5. 多级备份

提供边缘和中心多级服务备份。

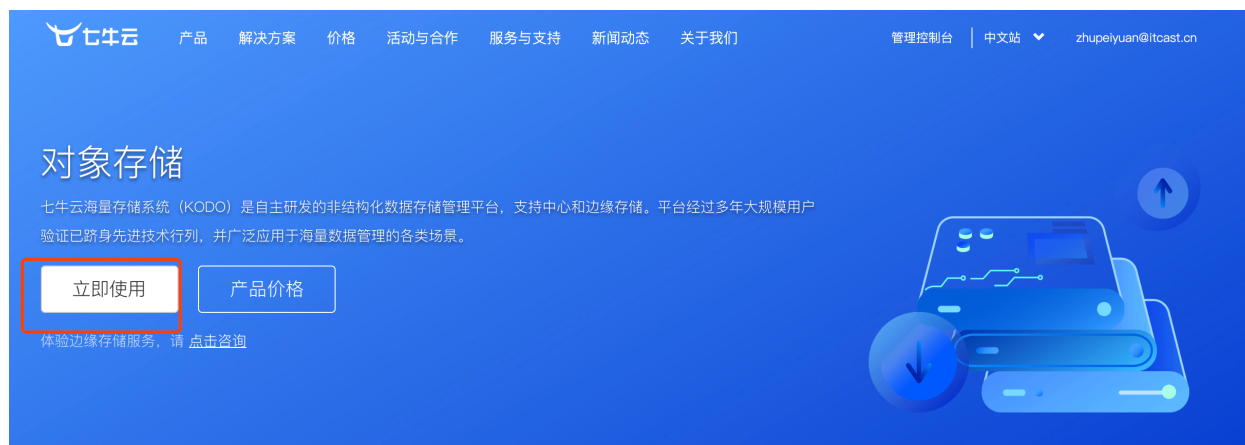
## 6. 边缘安全

通信加密，及时脱敏，边缘存储数据具备网银级安全可靠。

# 2 创建对象存储空间

1. 注册七牛云账号：<https://portal.qiniu.com/signup>

2. 开通对象存储服务：<https://www.qiniu.com/products/kodo>



3. 新建存储空间：<https://portal.qiniu.com/bucket/create>

选择服务器所在区域，访问控制方式。

公开空间可实现无需认证即可下载空间中的文件。

产品主页

邀请消息文档工单费用

体验新版

新建存储空间

搜索存储空间

存储空间列表

华南

华东

跨区域同步管理

存储空间名称

存储空间名称不允许重复，遇到冲突请更换名称。名称由 3 ~ 63 个字符组成，可包含小写字母、数字和短划线，且必须以小写字母或者数字开头和结尾。

存储区域

北美区域尚未支持自定义数据处理服务，一旦创建区域无法修改，请谨慎选择。

华东

华北

华南

北美

东南亚

华南区域相关文档

访问控制

公开和私有仅对 Bucket 的读文件生效，修改、删除、写入等对 Bucket 的操作均需要拥有者的授权才能进行操作。

公开空间

私有空间

#### 4. 创建成功，获取融合 CDN 测试域名：下载文件时需要使用此域名

产品主页

邀请消息文档工单费用

体验新版

新建存储空间

搜索存储空间

存储空间列表

华南

华东

xc-bucket01

跨区域同步管理

shanjupay-m1

空间概览

数据统计

内容管理

镜像存储

绑定域名

样式分隔符设置

图片样式

转码样式

空间授权

生命周期

空间设置

400 KB

200 KB

0 B

12月20日

12月22日

12月24日

12月26日

API 请求

GET

PUT

4次

2次

融合 CDN 测试域名

七牛融合 CDN 测试域名（以 clouddn.com/qiniucdn.com/qiniudn.com/qnssl.com/qbox.me 结尾），每个域名每日限总流量 10GB，每个测试域名自创建起 30 个自然日后系统会自动回收，仅供测试使用并且不支持 Https 访问，详情查看 [七牛测试域名使用规范](#)。点击域名可查看剩余回收时间。

q2cdf3lbb.bkt.clouddn.com

#### 5. 获取七牛云服务密钥：<https://portal.qiniu.com/user/key>

产品主页

邀请消息文档工单

个人信息

密钥管理

安全设置

提醒设置

操作日志

一个账号最多拥有两对密钥(Access/Secret Key)；更换密钥时，请创建第二个密钥；删除密钥前须停用；出于安全考虑，建议您周期性地更换密钥。您可以查看更多 [安全使用密钥建议](#)。

创建时间	AccessKey/SecretKey	状态	操作
2018-11-07	<div>AK: vG3km6ZU9j_KCIEsOW-76A3RxT...~kTDOf_</div> <div>SK: .....</div>	使用中	<div>停用</div> <div>显示</div>
2018-10-29	<div>AK: m44sYjPs2OX6H_DEutU-x8ILTTS...~MKK3IUJZ</div> <div>SK: .....</div>	使用中	<div>停用</div> <div>显示</div>

## 3 上传文件测试

官方提供了Java SDK来方便开发。

## 3.1 Java SDK简介

使用此 SDK 构建您的网络应用程序，能让您以非常便捷的方式将数据安全地存储到七牛云上。无论您的网络应用是一个网站程序，还是包括从云端（服务端程序）到终端（手持设备应用）的架构服务或应用，通过七牛云及其 SDK，都能让您应用程序的终端用户高速上传和下载，同时也让您的服务端更加轻盈。

Java SDK 属于七牛服务端SDK之一，主要有如下功能：

1. 提供生成客户端上传所需的上传凭证的功能
2. 提供文件从服务端直接上传七牛的功能
3. 提供对七牛空间中文件进行管理的功能
4. 提供对七牛空间中文件进行处理的功能
5. 提供七牛CDN相关的刷新，预取，日志功能

详细内容参考官网：<https://developer.qiniu.com/kodo/sdk/1239/java>

## 3.2 使用SDK上传文件

1. 在项目中引入依赖，在common工程添加以下依赖。

```
<dependency>
  <groupId>com.qiniu</groupId>
  <artifactId>qiniu-java-sdk</artifactId>
  <version>7.2.25</version>
</dependency>
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.2</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.47</version>
</dependency>
```

2. 在Common工程下编写七牛云的工具类QiniuUtils

```
package com.shanjupay.common.util;

import com.alibaba.fastjson.JSON;
import com.google.gson.Gson;
import com.qiniu.common.QiniuException;
import com.qiniu.http.Response;
import com.qiniu.storage.Configuration;
import com.qiniu.storage.Region;
import com.qiniu.storage.UploadManager;
import com.qiniu.storage.model.DefaultPutRet;
import com.qiniu.util.Auth;
```

```
public class QiniuUtils {  
  
}
```

3. 将七牛云的AccessKey/SecretKey和存储空间名配置到代码中：<https://portal.qiniu.com/user/key>

```
public static String accessKey = "";  
public static String secretKey = "";  
public static String bucket = "";
```

4. 调用SDK上传文件

参考官方提供的方法，这里测试字节数组上传方法：



```
01. //构造一个带指定 Region 对象的配置类  
02. Configuration cfg = new Configuration(Region.region0());  
03. //...其他参数参考类注释  
04.  
05. UploadManager uploadManager = new UploadManager(cfg);  
06. //...生成上传凭证，然后准备上传  
07. String accessKey = "your access key";  
08. String secretKey = "your secret key";  
09. String bucket = "your bucket name";  
10.  
11. //默认不指定key的情况下，以文件内容的hash值作为文件名  
12. String key = null;  
13.  
14. try {  
15.     byte[] uploadBytes = "hello qiniu cloud".getBytes("utf-8");  
16.     Auth auth = Auth.create(accessKey, secretKey);  
17.     String upToken = auth.uploadToken(bucket);  
18.  
19.     try {  
20.         Response response = uploadManager.put(uploadBytes, key, upToken);  
21.         //解析上传成功的结果  
22.         DefaultPutRet putRet = new Gson().fromJson(response.bodyString(), DefaultPutRet.class);
```

```
//上传测试  
private static void testUpload(){  
    //构造一个带指定 Region 对象的配置类  
    Configuration cfg = new Configuration(Region.huanan());  
    //...其他参数参考类注释  
    UploadManager uploadManager = new UploadManager(cfg);  
    //...生成上传凭证，然后准备上传  
    String accessKey = "vG3krn6ZU9j_KCJEsOW-76A3RxEBgCkN6tkTDOf_";  
    String secretKey = "p04hR8a1151UXAb0K3GpYznPxAGuSj_9LbegmJuS";  
    String bucket = "shanjupay-m1";  
    //默认不指定key的情况下，以文件内容的hash值作为文件名，这里建议由自己来控制文件名  
    String key = UUID.randomUUID()+".png";  
    FileInputStream fileInputStream = null;  
    try {  
        //通常这里得到文件的字节数组  
        fileInputStream = new FileInputStream(new  
File("F:\\develop\\shanjupay\\1.png"));  
        byte[] uploadBytes = IOUtils.toByteArray(fileInputStream);  
        //  
        byte[] uploadBytes = "hello qiniu cloud".getBytes("utf-8");  
        Auth auth = Auth.create(accessKey, secretKey);  
        String upToken = auth.uploadToken(bucket);
```



```
        try {
            Response response = uploadManager.put(uploadBytes, key, upToken);
            //解析上传成功的结果
            DefaultPutRet putRet = new Gson().fromJson(response.bodyString(),
DefaultPutRet.class);
            System.out.println(putRet.key);
            System.out.println(putRet.hash);
        } catch (QiniuException ex) {
            Response r = ex.response;
            System.err.println(r.toString());
            try {
                System.err.println(r.bodyString());
            } catch (QiniuException ex2) {
                //ignore
            }
        }
    } catch (IOException ex) {
        //ignore
    } finally {
        try {
            if(fileInputStream!=null){
                fileInputStream.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

#### 5. 编写测试方法上传文件

```
public static void main(String[] args) {
    //测试上传
    QiniuUtils.testUpload();
}
```

6. 上传测试完成，通过七牛云平台查看上传的文件：<https://portal.qiniu.com/bucket>，选择上传使用的存储空间查看内容

体验新版  
新建存储空间  
搜索存储空间

存储空间列表

华南

shanjupay-m1

华东

xc-bucket01

跨区域同步管理

shanjupay-m1

空间概览 数据统计 内容管理 镜像存储 绑定域名 样式分隔符设置 图片样式

空间设置

内容管理

帮助

上传文件

全选

刷新列表

共 1 个文件

共 162.23 KB 存储量

q2cdf3ibb.bkt.clouddn.com

保存默认域名

文件名

文件类型

存储类型

文件大小

94873f84-3d6f-49da-86d0-98ad4fa68613.png

image/png

标准存储

162.23 KB

## 4 下载文件测试

参考官方文档：

对象存储 >  
产品手册 >  
API Reference >  
SDK >  
Objective-C SDK  
Android SDK >  
Java SDK >  
Java SDK 历史文档  
PHP SDK >  
Python SDK  
Ruby SDK  
Node.js SDK >  
C# SDK >  
C/C++ SDK

下载文件

- 公开空间
- 私有空间

文件下载分为公开空间的文件下载和私有空间的文件下载。

公开空间

对于公开空间，其访问的链接主要是将空间绑定的域名（可以是七牛空间的默认域名或者是绑定的自定义域名）拼接上空间里面的文件名即可访问，标准情况下需要在拼接链接之前，将文件名进行 `urlencode` 以兼容不同的字符。

```
01. String fileName = "公司/存储/qiniu.jpg";  
02. String domainOfBucket = "http://devtools.qiniu.com";  
03. String finalUrl = String.format("%s/%s", domainOfBucket, fileName);  
04. System.out.println(finalUrl);
```

拷贝

私有空间

对于私有空间，首先需要按照公开空间的文件访问方式构建对应的公开空间访问链接，然后再对这个链接进行私有授权签名。

```
01. String fileName = "公司/存储/qiniu.jpg";  
02. String domainOfBucket = "http://devtools.qiniu.com";  
03. String encodedFileName = URLEncoder.encode(fileName, "utf-8").replace("+", "%20");  
04. String publicUrl = String.format("%s/%s", domainOfBucket, encodedFileName);  
05.
```

拷贝

### 1、公开空间

对于公开空间，文件的访问链接是将空间绑定的域名加文件名称组成。

产品主页 邀请 消息 文档 工单 费用

体验新版 新建存储空间 搜索存储空间

存储空间列表

华南 shanjupay-m1

华东 xc-bucket01

跨区域同步管理

shanjupay-m1

空间概览 数据统计 内容管理 镜像存储 绑定域名 样式分隔符设置 图片样式 转码样式 空间授权 生命周期

空间设置

上传文件 全选 刷新列表 共 1 个文件 共 162.23 KB 存储量

q2cdf3ibb.bkt.clouddn.com 保存默认域名

文件名	文件类型	存储类型	文件大小	最后更新	操作
94873f84-3d6f-49da-86d0-98ad4fa68613.png	image/png	标准存储	162.23 KB	2019-12-26 18:27:20	<a href="#">下载文件</a> <a href="#">转为低频存储</a> <a href="#">复制外链</a>

链接地址为：<http://q2cdf3ibb.bkt.clouddn.com/94873f84-3d6f-49da-86d0-98ad4fa68613.png>

## 2、私有空间

私有空间主要存储一些不公开的文件，如果要下载文件则需要通过sdk生成文件访问地址。

首先需要按照公开空间的文件访问方式构建对应的公开空间访问链接，然后再对这个链接进行私有授权签名。

测试代码如下：

```
//获取文件url
private static void testGetFileUrl() throws UnsupportedOperationException {
    String fileName = "94873f84-3d6f-49da-86d0-98ad4fa68613.png";
    String domainOfBucket = "http://q2cdf3ibb.bkt.clouddn.com";
    String encodedFileName = URLEncoder.encode(fileName, "utf-8").replace("+", "%20");
    String publicUrl = String.format("%s/%s", domainOfBucket, encodedFileName);
    String accessKey = "vG3krn6ZU9j_KCJES0W-76A3RxEbgCkN6tkTDOF_";
    String secretKey = "p04hR8a1151UXAb0K3GpYznPxAGuSj_9LbegmJuS";
    Auth auth = Auth.create(accessKey, secretKey);
    long expireInSeconds = 3600; //1小时，可以自定义链接过期时间
    String finalUrl = auth.privateDownloadUrl(publicUrl, expireInSeconds);
    System.out.println(finalUrl);
}
```

### 1. 编写测试方法

```
public static void main(String[] args) throws UnsupportedOperationException {
    //测试上传
    // QiniuUtils.testUpload();
    //测试获取文件url
    QiniuUtils.testGetFileUrl();
}
```





2. 执行测试返回下载地址

[http://q2cdf3ibb.bkt.clouddn.com/FqgpZCqGhGKZOf9zmvsfP\\_tSbREk?e=1577355665&token=vG3krn6ZU9j\\_KCTEsOW-76A3RxEbgCkN6](http://q2cdf3ibb.bkt.clouddn.com/FqgpZCqGhGKZOf9zmvsfP_tSbREk?e=1577355665&token=vG3krn6ZU9j_KCTEsOW-76A3RxEbgCkN6)

3. 在浏览器访问下载地址，测试文件是否正确