

# 云存储解决方案-阿里云OSS

## 1. 阿里云OSS简介

阿里云对象存储服务（Object Storage Service，简称OSS）为您提供基于网络的数据存取服务。使用OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

阿里云OSS将数据文件以对象（object）的形式上传到存储空间（bucket）中。

您可以进行以下操作：

- 创建一个或者多个存储空间，向每个存储空间中添加一个或多个文件。
- 通过获取已上传文件的地址进行文件的分享和下载。
- 通过修改存储空间或文件的属性或元信息来设置相应的访问权限。
- 在阿里云管理控制台执行基本和高级OSS任务。
- 使用阿里云开发工具包或直接在应用程序中进行RESTful API调用执行基本和高级OSS任务

## 2. OSS开通

（1）打开<https://www.aliyun.com/>，申请阿里云账号并完成实名认证。

<p>扫码注册</p>  <p>使用阿里云APP/支付宝/钉钉 下载阿里云APP，上云更轻松</p>	<p>账号注册 <u>手机号注册</u></p> <p>未注册手机号验证通过后将自动注册</p> <p>+86 输入手机号</p> <p>输入验证码   获取验证码</p> <p><input type="checkbox"/> 我已阅读并同意<a href="#">服务条款</a>、<a href="#">隐私政策</a></p> <p>注册</p>	<p>注册成功</p> <p>温馨提示：您可以前往<a href="#">账号中心</a>修改登录名、设置密码、绑定邮箱、修改手机号等</p> <p>登录名: <input type="text"/></p> <p>手机号: <input type="text"/></p> <p>完成实名认证方可购买200+款云产品</p> <p>快速实名认证</p> <p>返回</p>
---	---	---

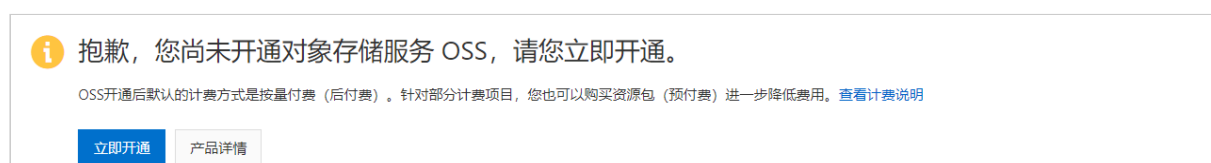
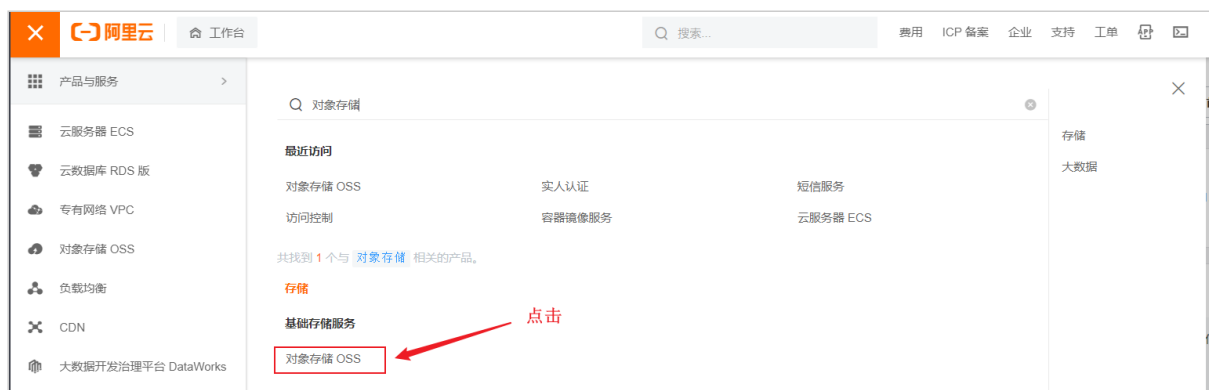
（2）充值 (可以不用做)

### （3）开通OSS

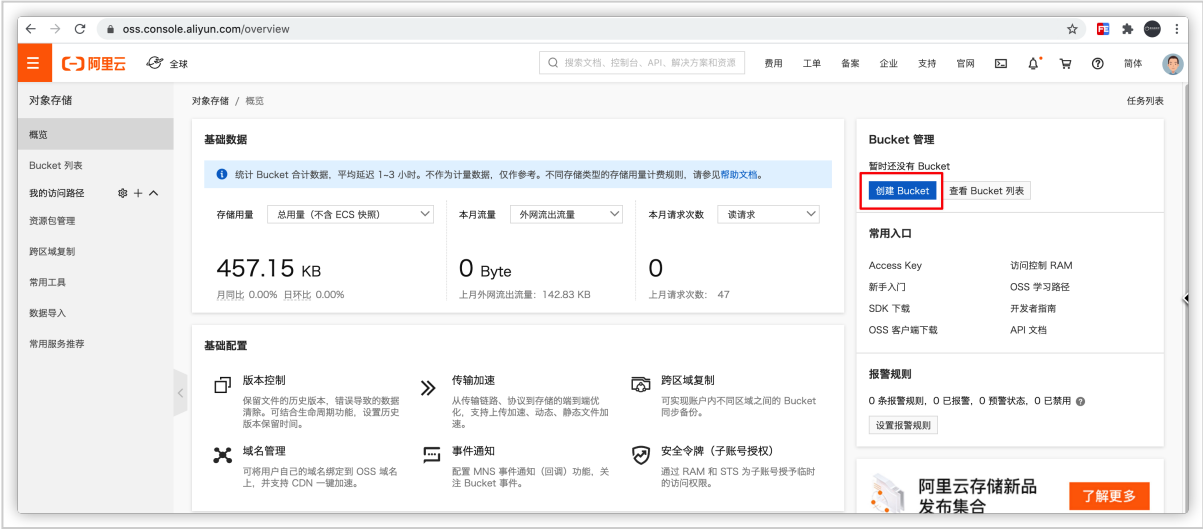
登录阿里云官网。点击右上角的控制台。



将鼠标移至产品，找到并单击对象存储OSS，打开OSS产品详情页面。在OSS产品详情页中的单击立即开通。



开通服务后，在OSS产品详情页面单击管理控制台直接进入OSS管理控制台界面。您也可以单击位于官网首页右上方菜单栏的控制台，进入阿里云管理控制台首页，然后单击左侧的对象存储OSS菜单进入**OSS管理控制台界面**。



#### （4）创建存储空间

新建Bucket，命名为 hmleadnews ，读写权限为 **公共读**

## 创建 Bucket



⚠ 注意: Bucket 创建成功后, 您所选择的 **存储类型**、**区域**、**存储冗余类型** 不支持变更。

\* Bucket 名称

web-framework01

15/63 ✓

\* 地域

华东1 (杭州)



相同区域内的产品内网可以互通; 订购后不支持更换区域, 请谨慎选择。

Endpoint

oss-cn-hangzhou.aliyuncs.com

所属资源组

请选择



存储类型

标准存储

低频访问存储

归档存储

冷归档存储

标准: 高可靠、高可用、高性能, 数据会经常被访问到。

[如何选择适合您的存储类型?](#)

HDFS服务 New



未开通

☐ 我已阅读《OSS-HDFS服务使用前须知》。

[前往授权](#)

HDFS服务说明请参考[HDFS服务](#) 开启后暂不支持关闭。

同城冗余存储 Hot



未开通

OSS 将您的数据以冗余的方式存储在同一个区域 (Region) 的 3 个可用区 (Zone) 中。提供机房级容灾能力, 能提高您的数据可用性, **同城冗余存储属性开启后, 将不支持关闭**。更多详情请参见 [同城冗余存储](#)。

⚠ 同城冗余存储将采用较高的计费标准。 [计费文档](#)

版本控制 Hot



未开通

开启版本控制后, 针对数据的覆盖和删除操作将会以历史版本的形式保存下来, 若不开启版本控制则数据删除或被覆盖将无法找回, 了解 [版本控制](#) 。当前未开启版本控制功能, 数据删除或被覆盖后将无法找回。

开启版本控制后, 对当前版本和所有历史版本的文件都会收取存储费用, 详情请查看[计费文档](#)

读写权限

私有

公共读

公共读写

公共读: 对文件写操作需要进行身份验证; 可以对文件进行匿名读。

服务端加密方式

☒ 无 ☐ OSS 完全托管 ☐ KMS

### 3. OSS快速入门

#### [参考文档官方](#)

##### (1) 创建测试工程，引入依赖

```
1 <dependency>
2     <groupId>com.aliyun.oss</groupId>
3     <artifactId>aliyun-sdk-oss</artifactId>
4     <version>3.15.1</version>
5 </dependency>
```

##### (2) 新建类和main方法

```
1 import org.junit.jupiter.api.Test;
2 import com.aliyun.oss.ClientException;
3 import com.aliyun.oss.OSS;
4 import com.aliyun.oss.OSSClientBuilder;
5 import com.aliyun.oss.OSSException;
6 import java.io.FileInputStream;
7 import java.io.InputStream;
8
9 public class AliOssTest {
10
11     @Test
12     public void testOss(){
13         // Endpoint以华东1（杭州）为例，其它Region请按实际情况填写。
14         String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
15         // 阿里云账号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RAM用户进行API访问或日常运维，请登录RAM控制台创建RAM用户。
16         String accessKeyId = "-----";
17         String accessKeySecret = "-----";
18         // 填写Bucket名称，例如examplebucket。
19         String bucketName = "-----";
20         // 填写Object完整路径，完整路径中不能包含Bucket名称，例如
        // exampledir/exampleobject.txt。
21         String objectName = "0001.jpg";
22         // 填写本地文件的完整路径，例如
        // D:\\localpath\\examplefile.txt。
```

```
23         // 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
        上传文件流。
24         String filePath=
        "C:\\Users\\Administrator\\Pictures\\Saved Pictures\\10.jpg";
25
26         // 创建OSSClient实例。
27         OSS ossClient = new OSSClientBuilder().build(endpoint,
        accessKeyId, accessKeySecret);
28
29         try {
30             InputStream inputStream = new
        FileInputStream(filePath);
31             // 创建PutObject请求。
32             ossClient.putObject(bucketName, objectName,
        inputStream);
33         } catch (OSSException oe) {
34             System.out.println("Caught an OSSException, which
        means your request made it to OSS, "
35             + "but was rejected with an error response
        for some reason.");
36             System.out.println("Error Message:" +
        oe.getErrorMessage());
37             System.out.println("Error Code:" +
        oe.getErrorCode());
38             System.out.println("Request ID:" +
        oe.getRequestId());
39             System.out.println("Host ID:" + oe.getHostId());
40         } catch (Exception ce) {
41             System.out.println("Caught an ClientException, which
        means the client encountered "
42             + "a serious internal problem while trying to
        communicate with OSS, "
43             + "such as not being able to access the
        network.");
44             System.out.println("Error Message:" +
        ce.getMessage());
45         } finally {
46             if (ossClient != null) {
47                 ossClient.shutdown();
48             }
49         }
50     }
```

## 4. 获取AccessKeyId

