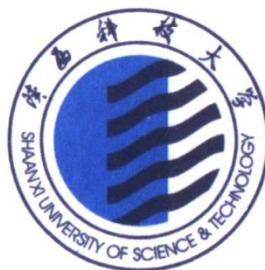


陕西科技大学

云计算实验报告



学 院: 电子信息与人工智能学院
专业名称: 计算机科学与技术
班 级: 计算机 232
学 号: 202307020122
姓 名: 马凌峰
任课老师: 齐 勇

实验六 利用对象存储服务（实验报告）

一、实验步骤与操作记录

1. 创建公有存储桶

- 在阿里云 OSS 控制台点击“创建 Bucket”。
- 名称：java-ai-test-010。
- 地域：华北 2（北京）。
- 读写权限：公共读。
- 存储类型：标准存储。

基本信息

存储空间名称	java-ai-test-010	存储类型	标准存储
地域	华北2 (北京)	创建时间	2025年11月25日 20:12
标签	未设置	版本控制	未开通
读写权限	公共读	归档直读	未开启
允许的 TLS 版本	未设置，全部允许	访问	文件可以被公共访问
存储冗余类型	同城冗余存储		

2. 上传测试文件

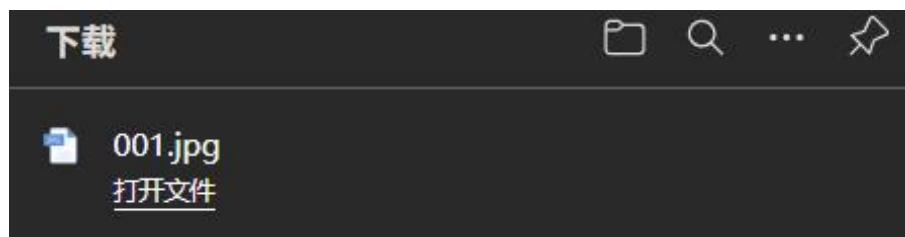
- 进入 Bucket，点击“上传文件”。
- 选择本地 001.jpg，上传。
- 上传成功后，在对象列表中可见该文件。

文件列表

□	文件名	文件大小	存储类型	更新时间	操作
□	2025/	未统计	标准存储	2025年12月7日 22:25:10	统计 彻底删除
□	001.jpg	90.881KB	标准存储	2025年12月7日 22:25:10	详情 :

3. 验证公网直接访问

- 复制对象 URL: <https://java-ai-test-010.oss-cn-beijing.aliyuncs.com/001.jpg>。
- 在浏览器新标签页打开，图片直接下载。



4. 集成到 Java Web 项目

- 按照阿里云官方 SDK 文档，集成到 java web 项目中。
- 实现头像上传接口：用户选择图片后，后端调用 `upload()` 方法将文件上传至 OSS。
- 前端页面通过后端提供的数据库查询接口得到 OSS URL 直接展示头像。
- 刷新页面后，头像能实时加载并显示，无需项目服务器存储文件。

员工管理

The screenshot shows a table with columns: 姓名 (Name), 性别 (Gender), 头像 (Avatar), 部门名称 (Department Name), 职位 (Position), 入职日期 (Joining Date), 最后操作时间 (Last Operation Time), and 操作 (Operations). There are two rows of data:

	姓名	性别	头像	部门名称	职位	入职日期	最后操作时间	操作
<input type="checkbox"/>	Marlin	男		财务部	班主任	2023-09-01	2025-11-29T02:43:15	<button>编辑</button> <button>删除</button>
<input type="checkbox"/>	hafds	男		教研部	班主任	2025-11-05	2025-11-25T23:56:17	<button>编辑</button> <button>删除</button>

二、实验结果

本次实验各项操作均顺利完成。

成功创建了名为 `java-ai-test-010` 的 Bucket，并将访问权限设置为“公共读”，确保上传的文件可通过公网直接访问。将本地准备的测试图片 `001.jpg` 成功上传至该存储桶的根目录。

通过浏览器直接访问该图片的 OSS URL，能够立即下载图片，验证了“公有读”权限已正确生效。

该 OSS 服务已成功集成到我的 Java Web 项目中：用户上传头像后，文件被存入 OSS，前端页面通过前端页面通过后端提供的数据库查询接口即可实时显示头像。

三、思考

1. 为什么用 OSS 存头像等资源？

传统做法是把图片存到服务器的 `static/upload/` 目录下，但这样不利于多台服务器需共享存储、浪费应用服务器带宽和磁盘，并且备份和扩容麻烦。而 OSS 天然支持高并发访问、自动冗余、按量计费，让 Web 应用更轻量化。

四、遇到的问题与解决方案

- 问题：一开始误将 Bucket 设为“私有”，前端出现 `403` 错误，加载失败。
解决：将 Bucket 修改为“公共读”。
- 问题：用户文件名相同时，文件被覆盖。
解决：文件名使用 `UUID + 后缀命名`，避免了重名覆盖。

五、实验结论

本次实验成功完成了阿里云 OSS 的创建、文件上传、公网访问验证，并将其集成到 Java Web 项目中实现用户头像功能。通过实践，我深刻体会到对象存储在 Web 开发中的价值：**解耦静态资源、降低服务器负担、提升系统可扩展性**。

未来在课程设计或实际项目中，我将优先考虑使用 OSS 存储图片、视频等静态内容，打造更健壮、更易维护的应用架构。