Assignment

TOPIC: Data Redundancy Removal in Cloud Storage

Abstract

In today's cloud-based storage systems, the problem of data redundancy increases storage costs and reduces efficiency. Duplicate files consume unnecessary space in cloud storage services, leading to higher costs and poor performance. This project implements a cloud-based **Data Redundancy Removal System** that scans files stored in an Amazon S3 bucket, detects duplicate files using cryptographic hash functions, and removes or manages them automatically. The system also generates a summary report for transparency and auditing.

Introduction

Cloud storage services such as AWS S3, Google Cloud Storage, and Azure Blob Storage have become integral for businesses and individuals. However, redundant data poses a challenge as it increases storage costs and affects scalability. Traditional manual deletion is not efficient in large-scale environments. Therefore, a system that can automatically detect and remove duplicate files is necessary.

This project focuses on leveraging **AWS S3 and Python (Boto3 library)** to automate duplicate detection and removal, ensuring optimized cloud storage usage.

Purpose of Work

- To identify and remove duplicate files in cloud storage.
- To minimize storage costs and improve efficiency.
- To automate the cleanup process with periodic execution.
- To provide users with **logs and reports** of removed duplicates for transparency.

Working Steps (Diagram)

Steps involved in the system:

1. Connect to AWS S3 bucket.
2. Scan files stored in the bucket.
3. Compute file hashes (SHA-256) to uniquely identify content.
4. Compare hashes to detect duplicates.
5. Remove duplicates and update logs.
6. Generate a **summary report**.

(You can add a simple block diagram here: User → AWS S3 → Script (Duplicate Detection) → Cleanup & Report)

System Architecture

Layers of the system:

- **User Layer:** Initiates the cleanup process.
- **Application Layer (Python Script):**
    - File scanning
    - Hash computation
    - Duplicate detection
    - File deletion & report generation
- **Cloud Storage Layer (AWS S3):** Stores files and responds to API calls.

(You can draw a simple architecture diagram: User → Python App → AWS S3 Service)

Hardware Requirements

- Processor: Intel i3 or above
- RAM: 4GB minimum
- Storage: 500MB free space
- Internet connection

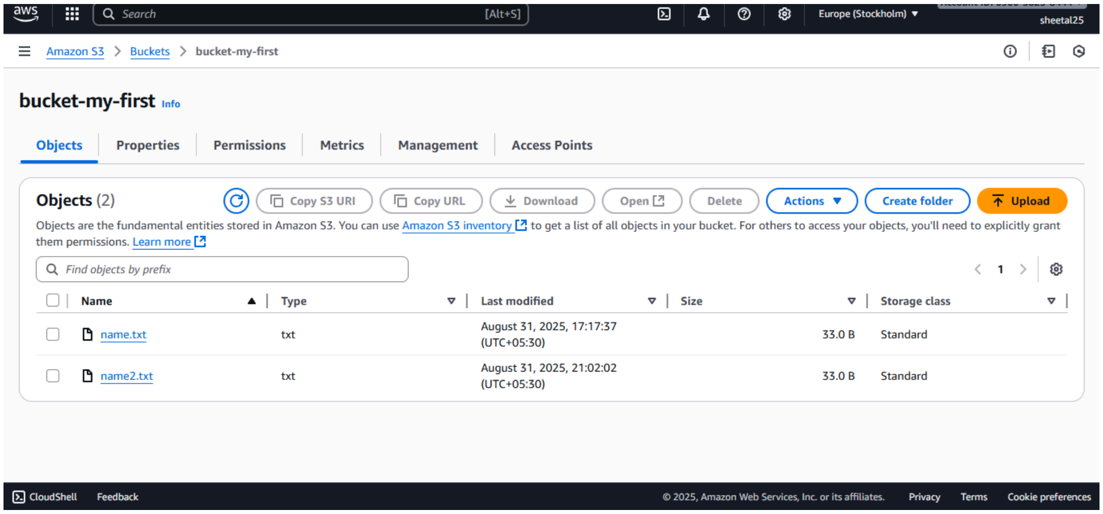
Software Requirements

- Operating System: Windows 11 / Linux / macOS
- Python 3.x installed
- AWS CLI configured
- Boto3 library installed
- AWS S3 bucket

Conclusion

The Data Redundancy Removal system provides a simple yet effective way to optimize cloud storage. By detecting and removing duplicate files automatically, it ensures reduced storage costs and efficient use of resources. The system can be further enhanced with scheduling, reporting dashboards, or integrating with multiple cloud providers.

Screenshots:

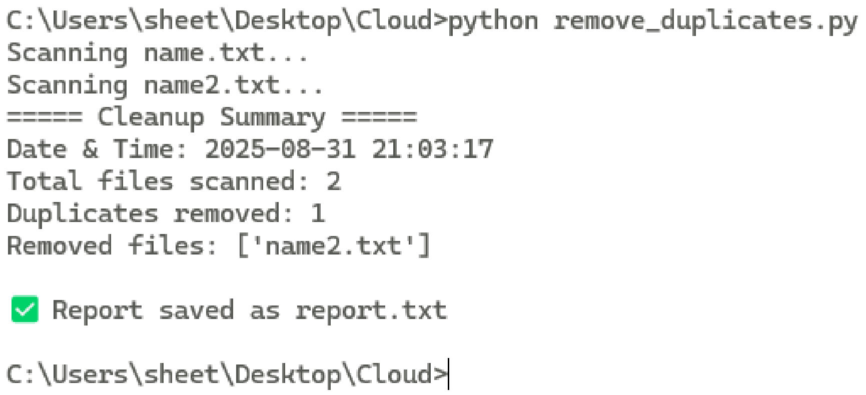


bucket-my-first

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

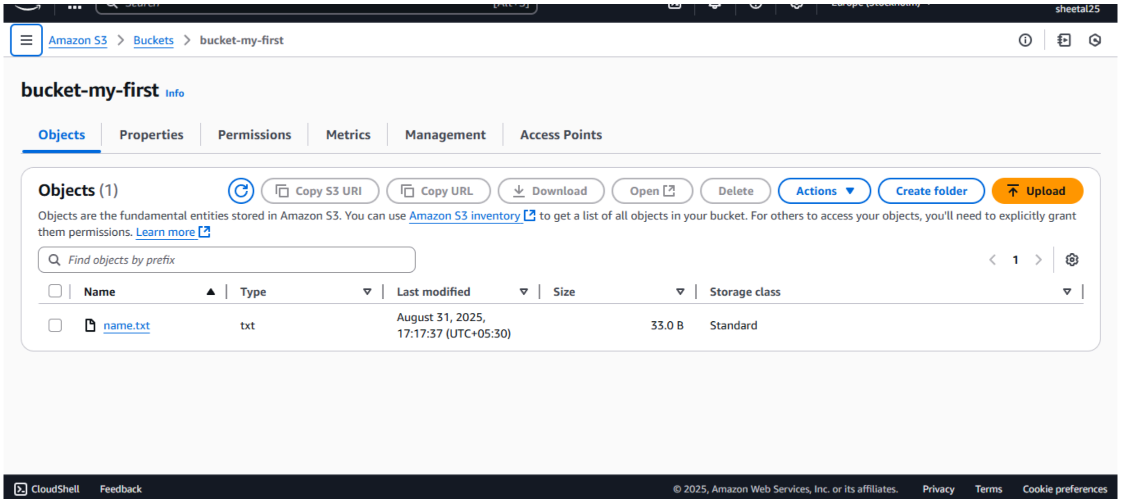
□	Name	Type	Last modified	Size	Storage class
□	name.txt	txt	August 31, 2025, 17:17:37 (UTC+05:30)	33.0 B	Standard
□	name2.txt	txt	August 31, 2025, 21:02:02 (UTC+05:30)	33.0 B	Standard



```
C:\Users\sheet\Desktop\Cloud>python remove_duplicates.py
Scanning name.txt...
Scanning name2.txt...
===== Cleanup Summary =====
Date & Time: 2025-08-31 21:03:17
Total files scanned: 2
Duplicates removed: 1
Removed files: ['name2.txt']

✅ Report saved as report.txt

C:\Users\sheet\Desktop\Cloud>
```



bucket-my-first

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	name.txt	txt	August 31, 2025, 17:17:37 (UTC+05:30)	33.0 B	Standard

2. Single-page detail (summary of project, screenshot of project, Technology used, Purpose of Project)

Title: Data Redundancy Removal in Cloud Storage

Summary of Project:
This project focuses on eliminating duplicate files from cloud storage (AWS S3) using Python and AWS SDK (Boto3). It scans all files, computes file hashes, detects duplicates, removes them, and generates a summary report. This helps organizations save costs and maintain efficient cloud usage.

Technology Used:

- Python 3.x
- AWS S3 (Cloud Storage)
- Boto3 (AWS SDK for Python)
- AWS CLI

Purpose of Project:

- Reduce cloud storage costs
- Remove redundant files automatically
- Improve efficiency and performance of cloud storage

Working Screenshot (Add here):
*(Paste the screenshot of your script output that shows "Duplicates removed: …")

System Architecture (Mini Diagram):
User → Python Script → AWS S3 → Cleanup & Report

3.Link to the project on GitHub
https://github.com/sheetal2506/cloudassignment.git