# S3 File Upload and Integrity Validation Using SHA-256 Checksum

## Contents

# Complete step-by-step documentation

## 1. Objective :

- Upload a file from a local machine to an Amazon S3 bucket
- Validate the **data integrity** of the uploaded file
- Understand **checksum (SHA-256)** and how it is used to detect corruption
- Confirm that the file uploaded to S3 matches the original file exactly

This is a critical cloud engineering skill used in backups, migrations, and secure data transfer.

## 2. Checksum

Checksum = the proof that the file was not changed.

It verifies:

- No corruption
- No modification
- No partial upload
- No network error
- No tampering

### 2.1 HOW checksum integrity check works (concept)

Step A — Calculate checksum on your local file
You run:

CertUtil -hashfile myfile.txt SHA256
Local checksum might be:

AAA111BBB222CCC333...
Step B — Get checksum from S3
Two ways:

*1. If S3 stores checksum metadata*
(using head-object):

ChecksumSHA256 = "AAA111BBB222CCC333..."
If the checksums match → integrity = OK.

---

*2. If S3 does NOT store checksum (like your case)*
Then do this:

Download the file:

aws s3 cp s3://bucket/myfile.txt downloaded.txt
Calculate checksum of downloaded file:

CertUtil -hashfile downloaded.txt SHA256
Compare:

Local: AAA111BBB222CCC333

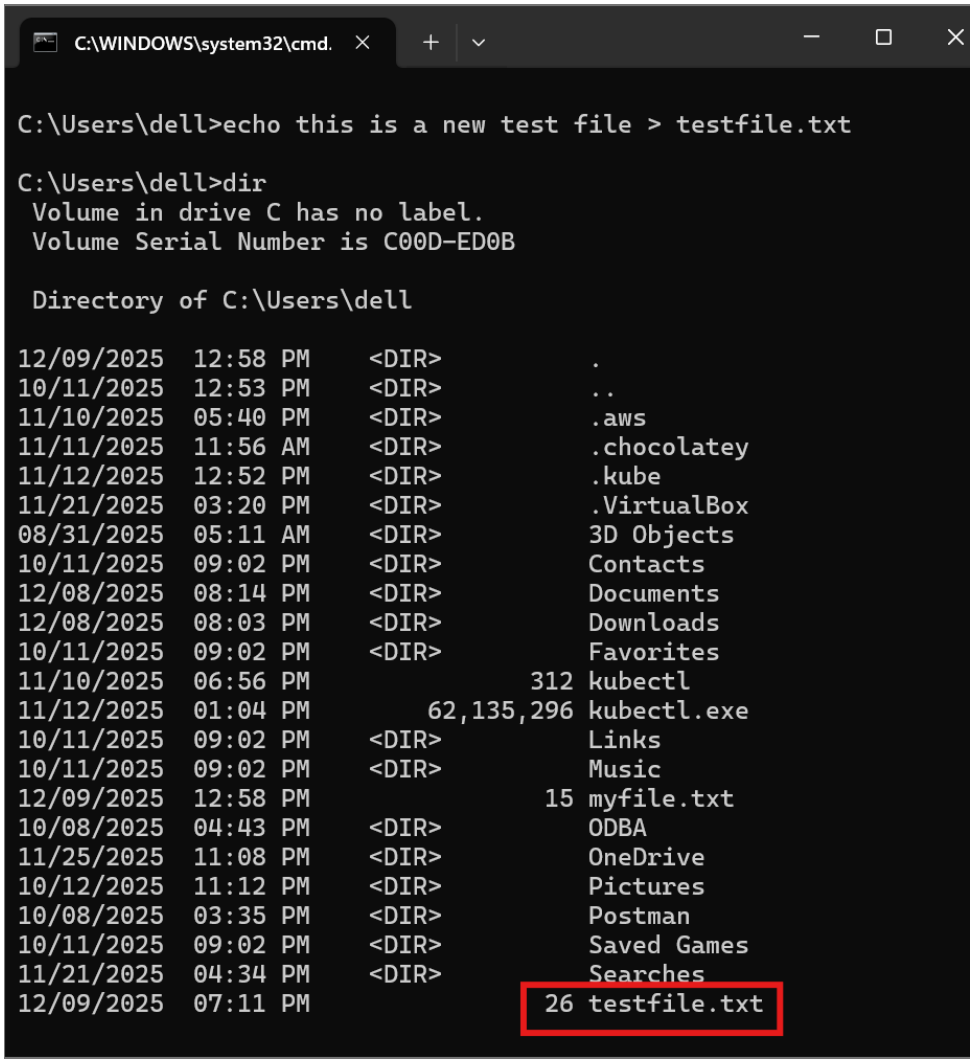
Downloaded: AAA111BBB222CCC333
If they match → **file is exactly identical**.

---

### 3. Configure AWS using credentials via cli.


### 4. Create a new test file

run this command: echo this is a new test file > testfile.txt
verify it exists: >>>> `dir testfile.txt`



5. **Compute local checksum (original file)**

   Run : `CertUtil -hashfile testfile.txt SHA256`

```
C:\Users\dell>CertUtil -hashfile testfile.txt SHA256
SHA256 hash of testfile.txt:
898393ccb6f176cc783e9417ad1af2453e7f9972ba64330c17166f6c0f655c8d
CertUtil: -hashfile command completed successfully.
```

   This is the **digital fingerprint** of your file.

6. **Upload the file to AWS S3**

   Run : `aws s3 cp testfile.txt s3://preeti-demo-bucket-2025-12-09/`


you should see:

`upload: ./testfile.txt to s3://preeti-demo-bucket-2025-12-09/testfile.txt`

This means the upload succeeded.

7. **Download the same file back**

Run : `aws s3 cp s3://preeti-demo-bucket-2025-12-09/testfile.txt downloaded.txt`

```
C:\Users\dell>
C:\Users\dell>aws s3 cp s3://preeti-demo-bucket-2025-12-09/testfile.txt downloaded.txt
Completed 26 Bytes/26 Bytes (41 Bytes/s) with 1 file(s) remainingdownload: s3://preeti-demo
-bucket-2025-12-09/testfile.txt to .\downloaded.txt
```

Now you have a second copy called `downloaded.txt`, which came from S3.

8. **Compute the Checksum of download file**

Run : `CertUtil -hashfile downloaded.txt SHA256`

```
C:\Users\dell>CertUtil -hashfile downloaded.txt SHA256
SHA256 hash of downloaded.txt:
898393ccb6f176cc783e9417ad1af2453e7f9972ba64330c17166f6c0f655c8d
CertUtil: -hashfile command completed successfully.

C:\Users\dell>
```

You will get another long hash string.

## 9. Compare both checkums

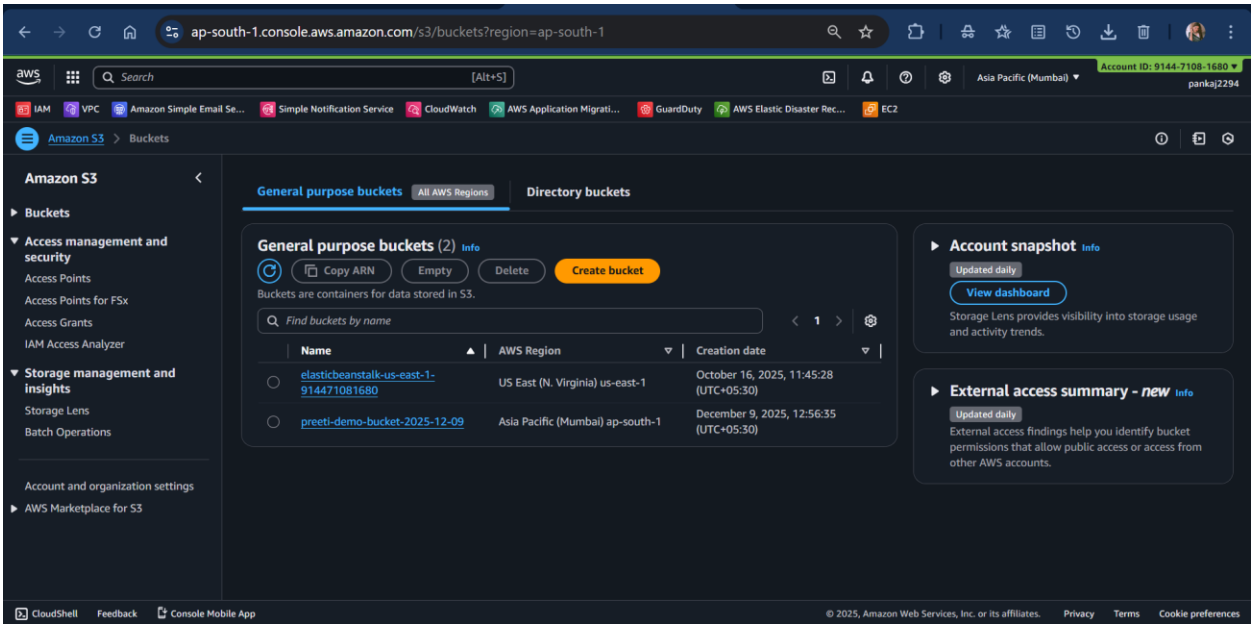
If both checksums are IDENTICAL: **Your original file == Your S3 file**

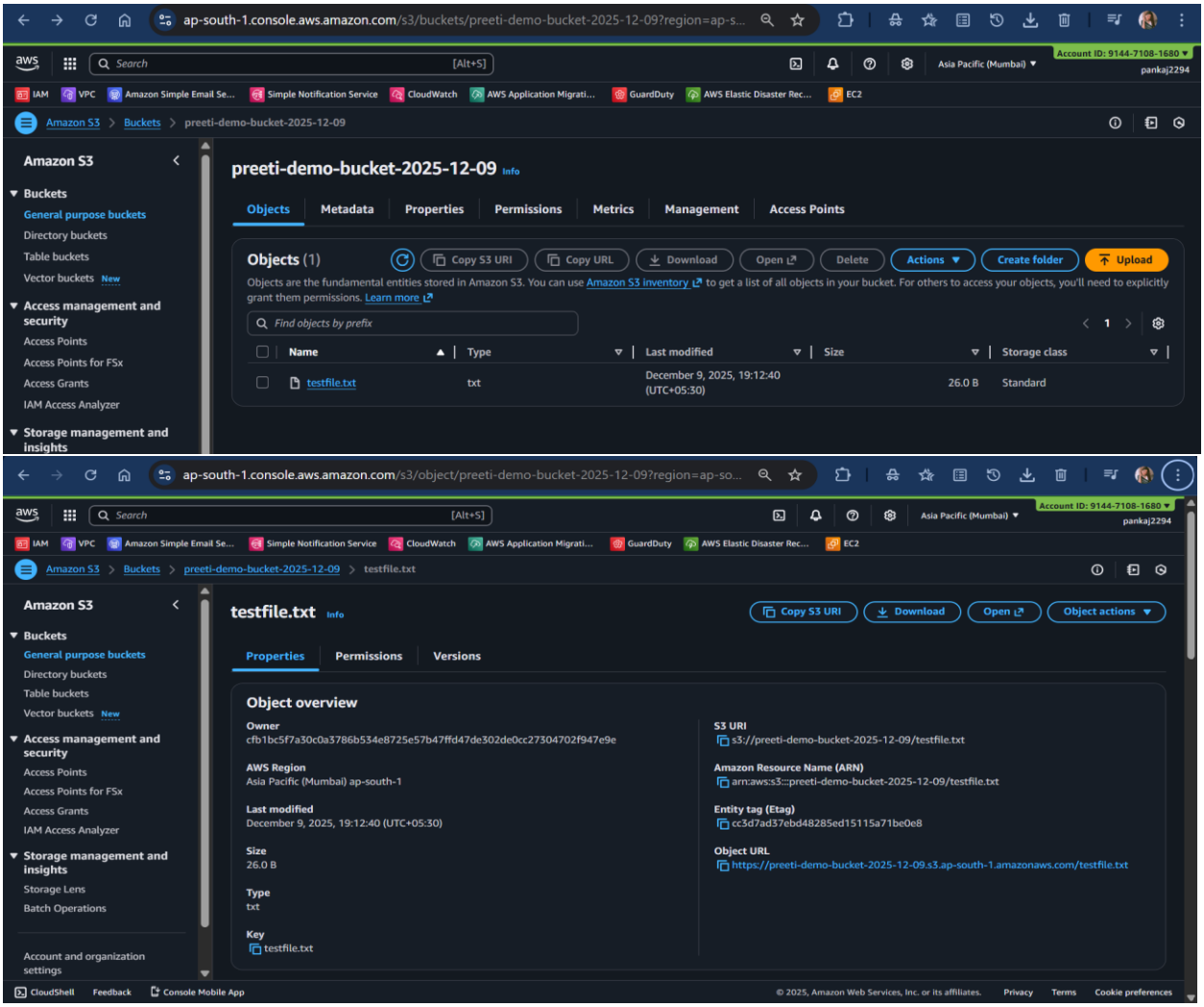
That means:

Integrity preserved
No corruption
No changes.

**If they are DIFFERENT:**

Something changed during upload/download.






## 10. Conclusion

A checksum is a digital fingerprint of a file
SHA-256 produces a unique hash.

Uploading to S3 did not modify or corrupt the file
The fingerprints were identical.

This proves the file was transferred correctly
No corruption
No loss
No extra bytes
No changes

You manually performed a data integrity validation workflow
This is exactly what cloud engineers do for:
backups
database dumps
logs

software releases
migrations
forensics
a full integrity check successfully.

Now you understand Step 3 completely:
✓ Calculate checksum locally
✓ Upload file
✓ Download file
✓ Recalculate
✓ Compare

If they match → Integrity OK

- Understand how to make S3 store checksum metadata automatically
- Learn MD5 vs SHA256
- Do the same test with a larger file
- Try a corrupt file scenario to see mismatch
- Learn how to automate this with a script