

Lab - Check the Integrity of Data in Amazon S3 with Additional Checksums

Steps:

1. Create a S3 Bucket
2. Upload a file and specify the checksum algorithm
3. Verify checksum
4. Cleanup

Step 1: Create a S3 bucket:

- From Amazon Console search S3 and then click on the Create bucket.
- Enter a descriptive globally unique name for your Bucket name: s3checksum-objects
- The default Block Public Access setting is appropriate for this workload, so leave this.
- Leave the remaining options as defaults, navigate to the bottom of the page, and choose Create bucket.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
US West (Oregon) us-west-2

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

s3checksum-objects

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [Info](#)

Account snapshot - updated every 24 hours All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

[View Storage Lens dashboard](#)

General purpose buckets | Directory buckets

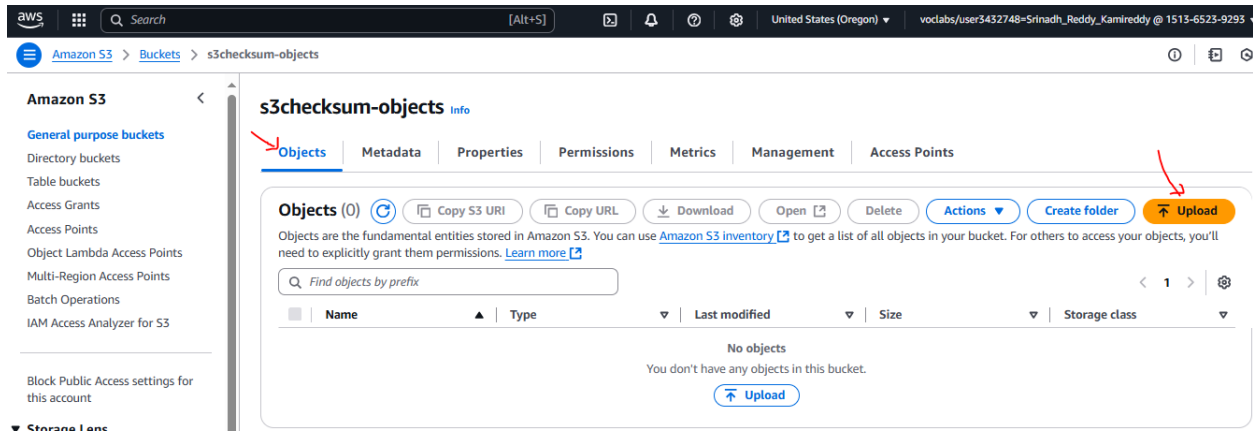
General purpose buckets (1) info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
s3checksum-objects	US West (Oregon) us-west-2	View analyzer for us-west-2	March 20, 2025, 17:35:23 (UTC+00:00)

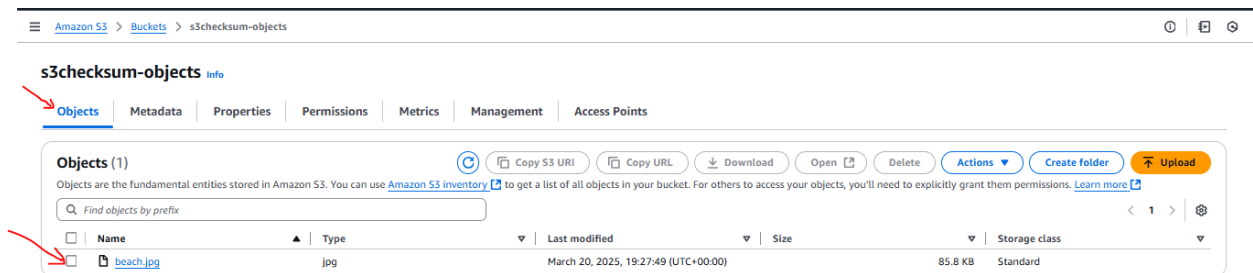
[illegible]

- Next, select the Objects tab. Then, from within the Objects section, choose the Upload button.



Add files

- Choose the Add files button and then select the file beach.jpg you would like to upload from your local desktop.



Expand properties

- Navigate down the page to find the Properties section. Then, select Properties and expand the section.

Select additional checksums

- Under checksums and checksum function choose SHA-256.

Amazon S3 > Buckets > s3checksum-objects > Upload

Checksums
Checksums are used for data integrity verification of new objects. [Learn more](#)

Checksum function
Checksum functions are used to calculate the checksum value. For objects smaller than 16 MB, only the full object checksum type is supported, for all checksum algorithms.

SHA256

Precalculated value - optional
When you provide a precalculated value for a single object, S3 compares it to the value it calculates using the selected checksum function. If the values don't match, the upload will fail. [Learn more](#)

Enter value

The precalculated value must be a Base64 encoded string. It must not exceed 128 characters, and can contain only letters (a-z, A-Z), numbers (0-9), forward slash (/), plus (+), or equals (=).

Tags - optional
You can use object tags to analyze, manage, and specify permissions for objects. [Learn more](#)

No tags associated with this resource.

Add tag

Metadata - optional
Metadata is optional information provided as a name-value (key-value) pair. [Learn more](#)

No metadata associated with this resource.

Add metadata

Cancel Upload

If your object is less than 16 MB and you have already calculated the SHA-256 checksum (base64 encoded), you can provide it in the Precalculated value input box. To use this functionality for objects larger than 16 MB, you can use the CLI or SDK. When Amazon S3 receives the object, it calculates the checksum by using the algorithm specified. If the checksum values do not match, Amazon S3 generates an error and rejects the upload, as shown in the screenshot.

Click on Upload

- Navigate down the page and choose the Upload button.
- After your upload completes, choose the Close button.

Step 3: Verify checksum:

- Select the uploaded file by selecting the filename. This will take you to the Properties page.

beach.jpg Info

Copy S3 URI Download Open Object actions

Properties Permissions Versions

Object overview

Owner
aws:labs0w4843310t1668704714

AWS Region
US West (Oregon) us-west-2

Last modified
March 20, 2025, 19:27:49 (UTC+00:00)

Size
85.8 KB

Type
jpg

Key
beach.jpg

S3 URI
s3://s3checksum-objects/beach.jpg

Amazon Resource Name (ARN)
arn:aws:s3::s3checksum-objects/beach.jpg

Entity tag (Etag)
1c6defc638f71abd065d8dd2f450b207

Object URL
https://s3checksum-objects.s3.us-west-2.amazonaws.com/beach.jpg

Object management overview
The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Management configurations

- Locate the checksum value

Navigate down the properties page and you will find the Additional checksums section.

Amazon S3 Buckets s3checksum-objects beach.jpg

Expiration date
The object will be permanently deleted on this date.

Storage class
Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#).

Storage class
Standard

Server-side encryption settings Info
Server-side encryption protects data at rest.

Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)

Checksums
Checksums are used for data integrity verification of new objects. [Learn more](#)

Checksum function
SHA256

Checksum type
Full object

Checksum value
qt0Ja6bq0Da7NOGvgIKxzfUdAN8HsDKHkT2qk6FavzM=

- This section displays the base64 encoded checksum that Amazon S3 calculated and verified at the time of upload.

Compare

- To compare the object in your local computer, open a terminal window and navigate to where your file is.

- Use a utility like shasum to calculate the file. The following command performs a sha256 calculation on the same file and converts the hex output to base64: `shasum -a 256 image.jpg | cut -f1 -d\ | xxd -r -p | base64`
- When comparing this value, it should match the value in the Amazon S3 console.
- In windows laptop, Open cmd prompt and run below command

```
C:\Users\DELL\Desktop>CertUtil -hashfile beach.jpg SHA256
SHA256 hash of beach.jpg:
aadd096ba6ead036bb34e1af8085f3c45b8300df07b0328792d66a93a15abf33
CertUtil: -hashfile command completed successfully.

C:\Users\DELL\Desktop>
```

- Open Powershell run below command
Below is the Powershell script

Step 1: Set the SHA-256 hash value into a variable

```
$hashValue = "aadd096ba6ead036bb34e1af8085f3c45b8300df07b0328792d66a93a15abf33"
```

Step 2: Convert the hexadecimal hash string to a byte array manually

```
$bytes = @()
```

```
for ($i = 0; $i -lt $hashValue.Length; $i += 2) {
```

```
    $bytes += [Convert]::ToByte($hashValue.Substring($i, 2), 16)
```

```
}
```

Step 3: Convert the byte array to a Base64-encoded string

```
$base64 = [Convert]::ToBase64String($bytes)
```

Step 4: Output the Base64 result

```
$base64
```

```

PS C:\Users\DELL>
PS C:\Users\DELL> $hashValue = "aadd096ba6ead036bb34e1af8085f3c45b8300df07b0328792d66a93a15abf33"
PS C:\Users\DELL> $bytes = @()
PS C:\Users\DELL> for ($i = 0; $i -lt $hashValue.Length; $i += 2) {
>>     $bytes += [Convert]::ToByte($hashValue.Substring($i, 2), 16)
>> }
PS C:\Users\DELL> $base64 = [Convert]::ToBase64String($bytes)
PS C:\Users\DELL> $base64
qt0Ja6bq0Da7NOGvgIXzxFuDAN8HsDKHktZqk6FavzM=
PS C:\Users\DELL>

```


Checksum value is same now qt0Ja6bq0Da7NOGvgIXzxFuDAN8HsDKHktZqk6FavzM=

Checksums

Checksums are used for data integrity verification of new objects. [Learn more](#)

Checksum function
SHA256

Checksum type
Full object

Checksum value
 qt0Ja6bq0Da7NOGvgIXzxFuDAN8HsDKHktZqk6FavzM=

Clean up:

In the following steps, you clean up the resources you created in this project.

Delete test object

Navigate to the S3 console and select the Buckets menu option. First you will need to delete the test object from your test bucket. Select the name of the bucket you have been working with for this tutorial. Put a check mark in the checkbox to the left of your test object name, then choose the Delete button. On the Delete objects page, verify that you have selected the proper object to delete and enter permanently delete into the Permanently delete objects confirmation box. Then, choose the Delete object button to continue.

Delete test bucket

Finally, you need to delete the test bucket you have created. Return to the list of buckets in your account. Select the radio button to the left of the bucket you created for this tutorial, and then choose the Delete button. Review the warning message. If you desire to continue deletion of this bucket, enter the bucket name into the Delete bucket confirmation box, and choose Delete bucket.

Conclusion

Congratulations! You have done how to upload a file to Amazon S3, calculate additional checksums, and compare the checksum on Amazon S3 and your local file to verify data integrity.