

# Mastering Data Management: Unleash the Power of Amazon S3 for Cloud Storage

Organize & Secure Data, Configure Permissions, Data Backup, Versioning & Retrieval



AutOps · Feb 24, 2024 · 7 min read



## PRODUCTIVITY

With good data management, your company will be more organized and productive. Employees will have an easier time finding, understanding, and relaying information.



## COST EFFICIENCY

Data management can help your organization avoid unnecessary extra costs such as unneeded duplication. When data is easily accessible, You won't have to worry about employees conducting the same research over and over again.



## OPERATIONAL NIMBLENESS

Great data management makes it easy for companies to respond quickly to the world around them. This means companies can respond efficiently to market changes and react appropriately to competitors.

## Why Is Data Management Important?



### SECURITY RISKS

Proper data management helps ensure that your information stays secure and never ends up in the wrong hands. A strong data management system will help protect your information from theft and attacks.



### REDUCED DATA LOSS

With a data management plan in place, you greatly reduce the risk of losing vital company information. It also ensures your important information is backed up and retrievable in case something happens to the original copies.



### ACCURATE DECISIONS

Proper data management helps ensure all employees and workers view and analyze the same, most recent information. This helps ensure that your company will be making the most accurate decisions based on the most accurate information.

## Table of contents

## Introduction To Amazon S3



- > 1. General Configurations
- > 2. Object Ownership
- > 3. Block Public Access settings for this bucket
- > 4. Bucket Versioning
- > 5. Tags and Default encryption
  - > Tags
  - > Encryption type
  - > Bucket Key
- > 6. Advanced settings
  - > Object Lock
- > 7. Organizing Data In S3 Bucket
- > 8. Configure Access Permissions For Secure Data Storage
  - > I. Create a folder with access permissions
  - > II. Upload files with access permissions
- > 9. Implement Data Backup And Retrieve
  - > Backing Up Data in S3:
  - > Retrieving Backed Up Data from S3:
- > 10. Clean Up

Show less ^

**Author: Ujwal Pachghare**

# Introduction

\*AWS S3, or Amazon Simple Storage Service, functions similarly to a big cloud-based digital filing cabinet. Consider that you require a location to store a large number of assets, including papers, films, images, and more. They may be stored on your computer, but what would happen if it broke or ran out of space? S3 fills that role.

S3 can be compared to a massive, safe, and dependable hard drive located high in the sky. It allows you to upload and download files, as well as organize them into various folders (referred to as "buckets" by S3).\*

\*The fact that S3 is "scalable," or able to expand with you, is one of its interesting features. S3 is capable of handling millions of files, even if you just have ten. You just pay for what you use, too.

Essentially, Amazon Web Services' S3 storage solution allows you to store and retrieve enormous volumes of data via the Internet[ at any time and from any location. It functions similarly to a cloud-based personal storage unit!

## Introduction to Amazon S3



Storage

# Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

## 1. General Configurations

👉 Go to Amazon S3 Console → Click on **Create Bucket** → Choose your region  
→ Give your **S3 Bucket** a unique **name** → You can directly **copy** settings from **existing S3 bucket**

### General configuration

AWS Region

Asia Pacific (Mumbai) ap-south-1

Bucket name [Info](#)

my-s3-storage-bucket-2024

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

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

[Choose bucket](#)

Format: s3://bucket/prefix

## 2. Object Ownership

# Object ownership determinants with ACLs (Access Control Lists)

## ► ACLs Disabled

## ► ACLs Enabled

- **Bucket owner preferred:** The bucket owner owns and has full control over new objects that other accounts write to the bucket with the bucket-owner-full-control canned ACLs.
- **Object writer:** The AWS account that uploads an object owns the object, has full control over it and can grant other users access to it through ACLs

### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**⚠️** We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

### Object Ownership

**Bucket owner preferred**

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

**Object writer**

The object writer remains the object owner.

**ℹ️** If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#) 

## 3. Block Public Access settings for this bucket

### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

#### Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

##### Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

##### Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

##### Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

##### Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

## 4. Bucket Versioning

### ► Disable

### ► Enable

- **Advantages Of Versioning:** Amazon S3 bucket versioning is a feature that allows you to keep multiple versions of an object in the same bucket. Versioning can help you recover objects from accidental deletions or overwrites.
- **Keep In mind:** Once you Enable it, you can't Disable it, but you can suspend it and, if you have three versions of an object stored, you are charged for three objects.

## Bucket Versioning

Versioning is a means of keeping every version of every object stored in your bucket. It helps you preserve, retrieve, and restore objects in both unintended user actions and application failures. [Learn more](#)

### Bucket Versioning

Disable

Enable

## 5. Tags and Default encryption

### Tags



*Tags in Amazon S3 are key-value pairs that allow you to categorize and classify your S3 buckets based on specific attributes such as project name, environment, or ownership. You can also use bucket tags to track storage costs and organize buckets.*

### Encryption type

#### *Server-side encryption with Amazon S3 managed keys (SSE-S3)*

- This encrypts data with S3-managed keys, which is the default option for S3.*

#### *Server-side encryption with Amazon KMS Keys (SSE-KMS)*

- We can encrypt data with Amazon KMS Keys by giving permissions that we want.*

#### *Dual-layer Server-side encryption with Amazon KMS Keys (DSSE-KMS)*

- We can apply **Dual-layer encryption with Amazon KMS**

## Bucket Key

1. The S3 Bucket Key feature is like a master key for your locker. Instead of having a separate key for each document, you have one master key that can open any document in the S3 bucket.
2. Every time you use it, it creates a temporary copy of itself (a “data key”) to open a document. This means that even if someone else gets hold of this temporary key, they can only access that one document, not the whole S3 bucket.
3. S3 Bucket Keys aren't supported for DSSE-KMS

**Tags - optional (0)**  
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

**Default encryption** [Info](#)  
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)  
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)  
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage tab](#) of the [Amazon S3 pricing page](#).

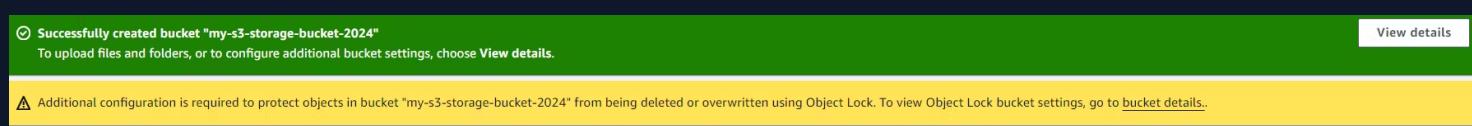
**Bucket Key**  
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable  
 Enable

## 6. Advanced

### Object Lock

- ▶ **Disable**
- ▶ **Enable**



## 7. Organizing Data In S3 Bucket

We can store data in a well-organized manner in S3 Bucket. For example, we can create folders for every different data type (image, video, music) and store sub-data in them.

**Ex:** We create an image folder for storing images, and then we create a sub-folder for storing different kinds of images (PNG, JPG, and GIF), and then we create a sub-folder for storing images for every month (Jan, Feb, and March), and we will store January month images in the January folder.

Objects (4) <a href="#">Info</a>						
<a href="#">Actions</a> <a href="#">Create folder</a> <a href="#">Upload</a>						
<input type="checkbox"/> <a href="#">Find objects by prefix</a> <input checked="" type="radio"/> <a href="#">Show versions</a> <a href="#">Previous</a> <a href="#">1</a> <a href="#">Next</a> <a href="#">Last</a>						
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class	⋮
<input type="checkbox"/>	Images/	Folder	-	-	-	⋮
<input type="checkbox"/>	Music/	Folder	-	-	-	⋮
<input type="checkbox"/>	Templates/	Folder	-	-	-	⋮
<input type="checkbox"/>	Videos/	Folder	-	-	-	⋮

**Objects (3) Info**

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	GIF/	Folder	-	-	-
<input type="checkbox"/>	JPG/	Folder	-	-	-
<input type="checkbox"/>	PNG/	Folder	-	-	-

**Objects (3) Info**

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	February /	Folder	-	-	-
<input type="checkbox"/>	January /	Folder	-	-	-
<input type="checkbox"/>	March/	Folder	-	-	-

**Objects (4) Info**

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	IMG_20220802_152357.jpg	jpg	February 24, 2024, 16:37:27 (UTC+05:30)	469.1 KB	Standard
<input type="checkbox"/>	IMG_20220802_155430.jpg	jpg	February 24, 2024, 16:37:30 (UTC+05:30)	654.9 KB	Standard
<input type="checkbox"/>	IMG_20230612_111508.jpg	jpg	February 24, 2024, 16:37:35 (UTC+05:30)	2.1 MB	Standard
<input type="checkbox"/>	IMG_20230627_085218.jpg	jpg	February 24, 2024, 16:37:37 (UTC+05:30)	226.5 KB	Standard

## 8. Configure Access Permissions For Secure Data Storage

At point No 7 we have created an image folder and uploaded .JPG files to it. while creating folders and uploading files, we can configure permissions and secure our data storage. You just have to follow the respective points.

Prerequisite: KMS Key with permissions you want

### I. Create a folder with access permissions

- 👉 Go into the S3 bucket → Click on **Create Folder** → Give a name to folder → In the **Server-side encryption** Check the **Specify an encryption key option** → Then check to **Override bucket settings for default encryption option** -> Then check **Server-side encryption with AWS Key Management Service**

keys (SSE-KMS) | AWS KMS keys option →

Choose key | : | ... | Older

## Server-side encryption Info

Server-side encryption protects data at rest.

i The following encryption settings apply only to the folder object and not to sub-folder objects.

### Server-side encryption

Do not specify an encryption key

The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

Specify an encryption key

The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

### Encryption settings Info

Use bucket settings for default encryption

Override bucket settings for default encryption

### Encryption type Info

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

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

### AWS KMS key Info

Choose from your AWS KMS keys

Enter AWS KMS key ARN

### Available AWS KMS keys

arn:aws:kms:ap-south-1:814495875142:key/4d9...



Create a KMS key

Cancel

Create folder

## II. Upload files with access permissions

- 💡 Go into the S3 bucket → Click on Create Folder → Click on Upload file → Choose your desired files → In the Access control list section, Select Specify individual ACL permissions → Now you can choose whom to give read permissions → We enabled object locks so no one can overwrite

except the owner. You can grant access to other AWS accounts by entering its canonical ID.

## Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. [Learn more](#)

 AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#)

### Access control list (ACL)

- Choose from predefined ACLs
- Specify individual ACL permissions

Grantee	Objects	Object ACL
---------	---------	------------

Object owner (your AWS account)	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
---------------------------------	--	---

Canonical ID:

2b32f1000e195636e1281

bf080414fa2a2601e8fdcb12fe  
8dda39052d7de5e5a

Everyone (public access)	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input type="checkbox"/> Write
--------------------------	-------------------------------	---

Group:  
 http://acs.amazonaws.com/groups/global/AllUsers

Authenticated users group (anyone with an AWS account)	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input type="checkbox"/> Write
---	-------------------------------	---

Group:  
 http://acs.amazonaws.com/groups/global/AuthenticatedUsers

### Access for other AWS accounts

Grantee	Objects	Object ACL
---------	---------	------------

<input type="text" value="Enter canonical ID"/>	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input type="checkbox"/> Write	<a href="#">Remove</a>
---	-------------------------------	---	------------------------

[Add grantee](#)

 In the properties section, you can choose the storage class for your object.

**Properties**

Specify storage class, encryption, and more.

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

Storage class	Designed for	Bucket type	Availability Zones	
S3 Express One Zone	Single-digit millisecond response times for the most frequently accessed data.	Directory	1	-
<b>Standard</b>	Frequently accessed data (more than once a month) with milliseconds access	General purpose	≥ 3	-
Intelligent-Tiering	Data with changing or unknown access patterns	General purpose	≥ 3	-
Standard-IA	Infrequently accessed data (once a month) with milliseconds access	General purpose	≥ 3	-
One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	General purpose	1	-
Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	General purpose	≥ 3	-
Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	General purpose	≥ 3	-
Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	General purpose	≥ 3	1



In the **server-side encryption section** → you can do the same as we did for folder encryption → you can enable the **bucket key** only if you enabled it while creating your bucket.

## Server-side encryption

Server-side encryption protects your objects as they're stored and retrieved.

### Server-side encryption

- Do not specify an encryption key

The bucket settings for default encryption are used to encrypt objects when storing them in Amazon S3.

- Specify an encryption key

The specified encryption key is used to encrypt objects before storing them in Amazon S3.

#### Encryption settings | [Info](#)

- Use bucket settings for default encryption
- Override bucket settings for default encryption

#### Encryption type | [Info](#)

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage tab](#) of the [Amazon S3 pricing page](#).

#### AWS KMS key | [Info](#)

- Choose from your AWS KMS keys
- Enter AWS KMS key ARN

#### Available AWS KMS keys

arn:aws:kms:ap-south-1:814495875142:key/3a8...



Create a KMS key



#### Bucket Key is enabled for objects uploaded, modified, or copied in this bucket

Uploaded, modified, or copied objects inherit their Bucket Key settings from the bucket default encryption configuration unless they already have Bucket Key configured. [Learn more](#)

#### Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable

- Enable



You can't specify Object Lock from here → In **Additional Checksums**, you can secure your secret data by applying checksum formats to it like **SHA-1**, **SHA-256**, etc. → You can give **Tags** or **Metadata** to recognize your data easily → Finally, Click on **Upload**

## Additional checksums

Checksum functions are used to verify data integrity.

### Additional checksums

#### Off

Amazon S3 will use a combination of MD5 checksums and Etags to verify data integrity.

#### On

Specify a checksum function for additional data integrity validation.

### Checksum function

Choose the checksum function that should be used to calculate the checksum value.

SHA-256



### Precalculated value - optional

When you provide a precalculated value for a single object less than 16 MB, S3 compares it with the value it calculates using the selected checksum function. If the values don't match, the upload will not start. [Learn more](#)

Enter value

 You can provide a precalculated value when uploading a single object less than 16 MB. To use precalculated values with multiple objects, use the CLI or SDK.

### 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

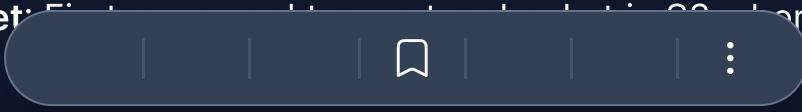
### Metadata - optional

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

No metadata associated with this resource.

## 9. Implement Data Backup And Retrieve

### Backing Up Data in S3:

- 
1. **Create a Bucket:** First, you need to create a bucket where you'll store your data.
  2. **Upload Data:** Next, you can upload your data (like files, images, etc.) to this bucket.
  3. **Enable Versioning:** To protect your data from accidental deletion or overwriting, you can enable versioning on your bucket. You can see
  4. **Use AWS Backup:** For additional protection, you can use AWS Backup, a service that automates backup tasks.

## Retrieving Backed Up Data from S3:

1. **Access the Bucket:** Open the Amazon S3 console and navigate to your bucket.
2. **Access the Object:** Navigate to the folder of the **deleted object**. Turn on **Show versions**.
3. **Find the File:** Choose the file that you want to open or download. Select the previous version of the object. Don't select the delete marker.
4. **Download the File:** Choose 'Actions', and then choose 'Open' or 'Download'.

---

## 10. Clean Up

 Navigate to the Bucket you want to delete but first we have to delete all data from it → Click on the **Empty button** → type **permanently delete in the box** and click on **Empty**

 Now we have an empty bucket, let's delete it now → Select the bucket → click on the **Delete button** → type the bucket name, and click on **Delete**.