Cloud storage and data management

Arihant Garg

-Intern at Extion Infotech

ACKNOLEDGEMENT

I would like to express my sincere gratitude to everyone who supported me throughout my internship and contributed to the successful completion of this project.

I am deeply grateful to Extion Infotech for providing me with the opportunity to intern with them and for offering a conducive learning environment. Their expertise and encouragement greatly contributed to my professional growth during this internship..

I would like to express my gratitude to my family and friends for their constant encouragement and support throughout this internship journey.

Thank you all for your unwavering support and for contributing to the successful completion of this project.

TABLE OF CONTENTS

* INTRODUCTION
* What is S3?
* Other Storage Services
* Why S3?
* CREATING A BUCKET
* General Configurations
* Object ownership
* Block public access to bucket
* Bucket Versioning
* Default Encryption
* Bucket Key

ADVANCED SETTINGS

* Object Lock
* ORGANIZING DATA IN BUCKET
* Creating a folder with SSE-KMS
* Specifying ACLs
* Storage Class
* Defining SSE-KMS for files
* Additional Checksums
* DATA BACKUP
* Bucket Versioning
* AWS Backup
* RETRIVING BACKUP DATA
* CLEANING UP
* REFERENCES
* **What is Simple Storage Service(S3)?**

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

* **Other Storage Services**

**Amazon EC2 Instance Store**

Instance store is ephemeral block storage. This is preconfigured storage that exists on the same physical server that hosts the EC2 instance and cannot be detached from Amazon EC2. You can think of it as a built-in drive for your EC2 instance. Instance store is generally well-suited for temporary storage of information that is constantly changing, such as buffers, caches, and scratch data. It is not meant for data that is persistent or long-lasting. If you need persistent long-term block storage that can be detached from Amazon EC2 and provide you more management flexibility, such as increasing volume size or creating snapshots, then you should use Amazon EBS.

**Amazon EBS**

Amazon EBS is meant for data that changes frequently and needs to persist through instance stops, terminations, or hardware failures. Amazon EBS has two different types of volumes: SSD-backed volumes and HDD-backed volumes. SSD-backed volumes have the following characteristics.

* Performance depends on IOPS (input/output operations per second).
* Ideal for transactional workloads such as databases and boot volumes.

HDD-backed volumes have the following characteristics:

* Performance depends on MB/s.
* Ideal for throughput-intensive workloads, such as big data, data warehouses, log processing, and sequential data I/O.

Why EBS is not generally preferred.

* It is block storage.
* EBS volumes are replicated across multiple servers in a single Availability Zone.

Most EBS volumes can only be attached to a single EC2 instance at a time.

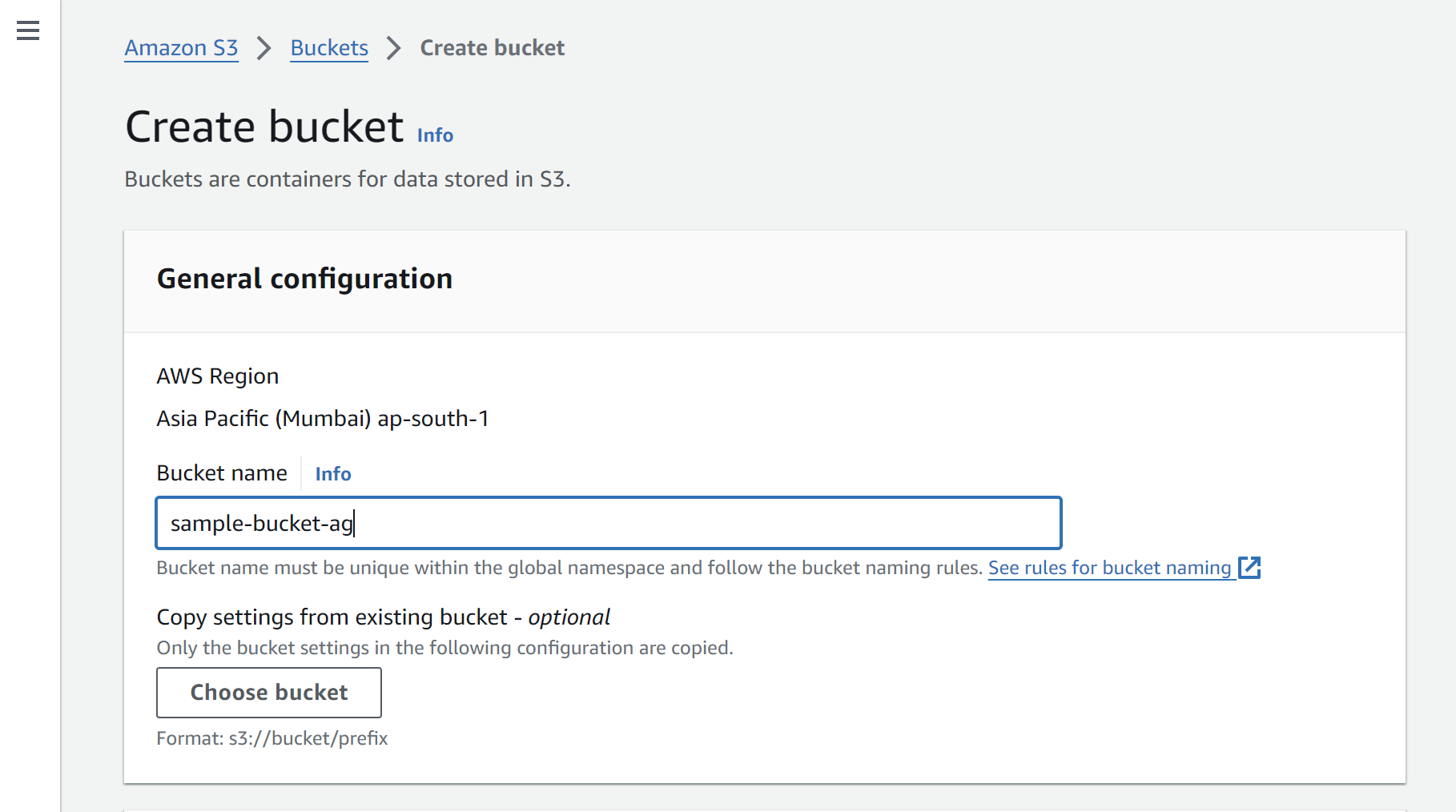
**Why S3?**

If your data doesn’t change that often, Amazon S3 might be a more cost-effective and scalable storage solution. S3 is ideal for storing static web content and media, backups and archiving, data for analytics, and can even be used to host entire static websites with custom domain names. Here are a few important features of Amazon S3 to know about when comparing it to other services.

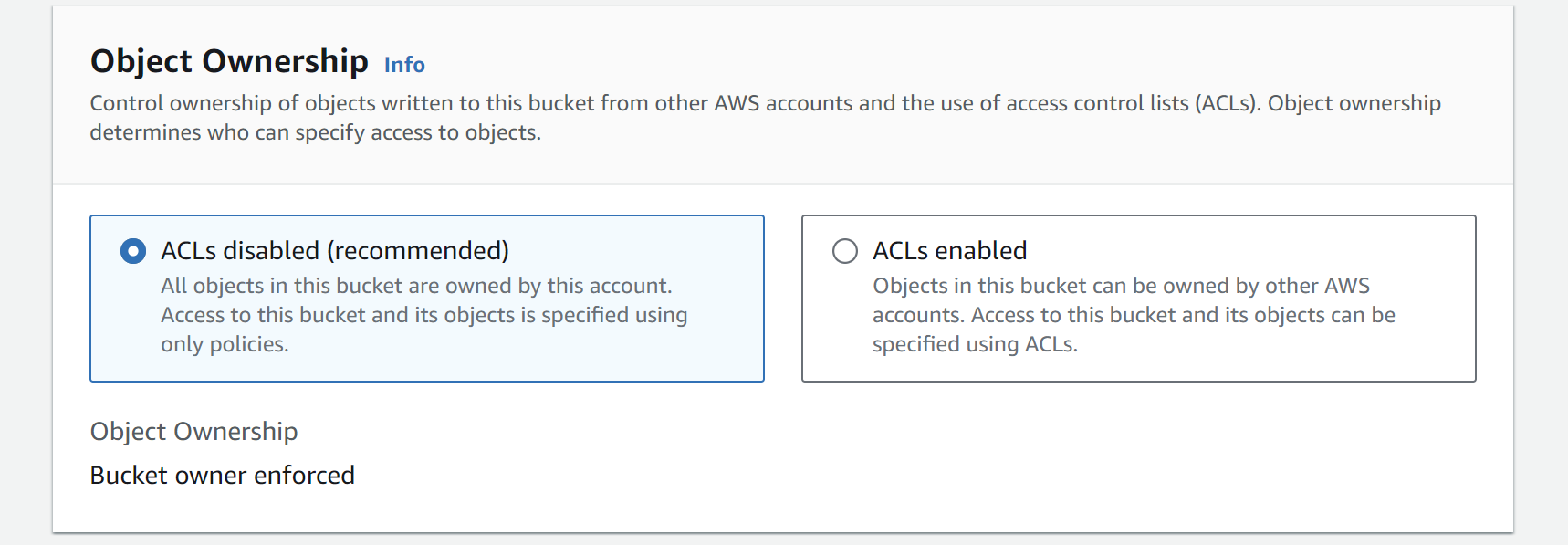
* It is object storage.
* You pay for what you use (you don’t have to provision storage in advance).
* Amazon S3 replicates your objects across multiple Availability Zones in a Region.
* Amazon S3 is not storage attached to compute.
* **Creating a bucket**

**1.General Configurations**

Create a unique global bucket name for your storage.



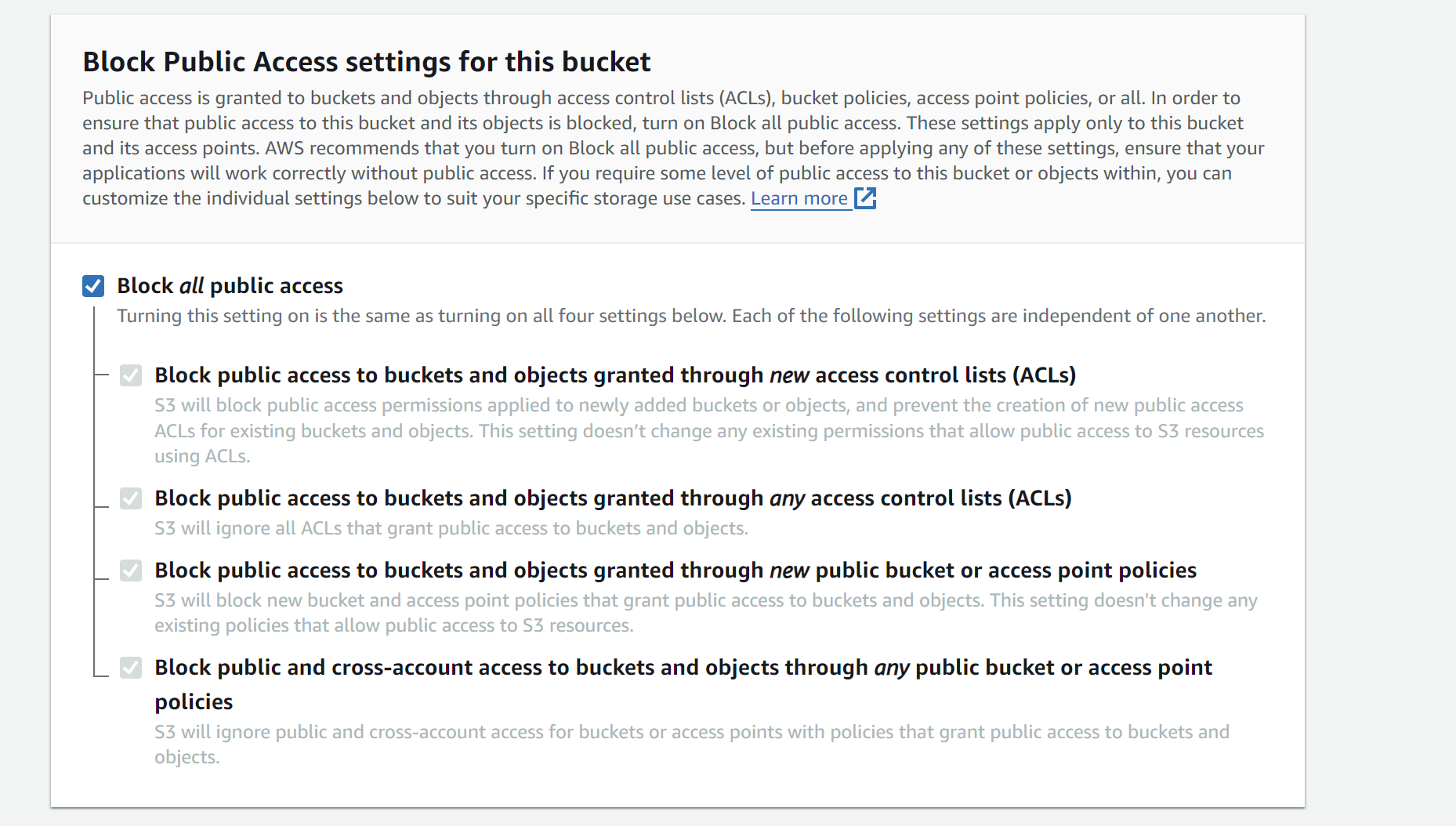
**2. Object Ownership**



Object ownership determines who can specify access to objects with ACLs (Access Control Lists).

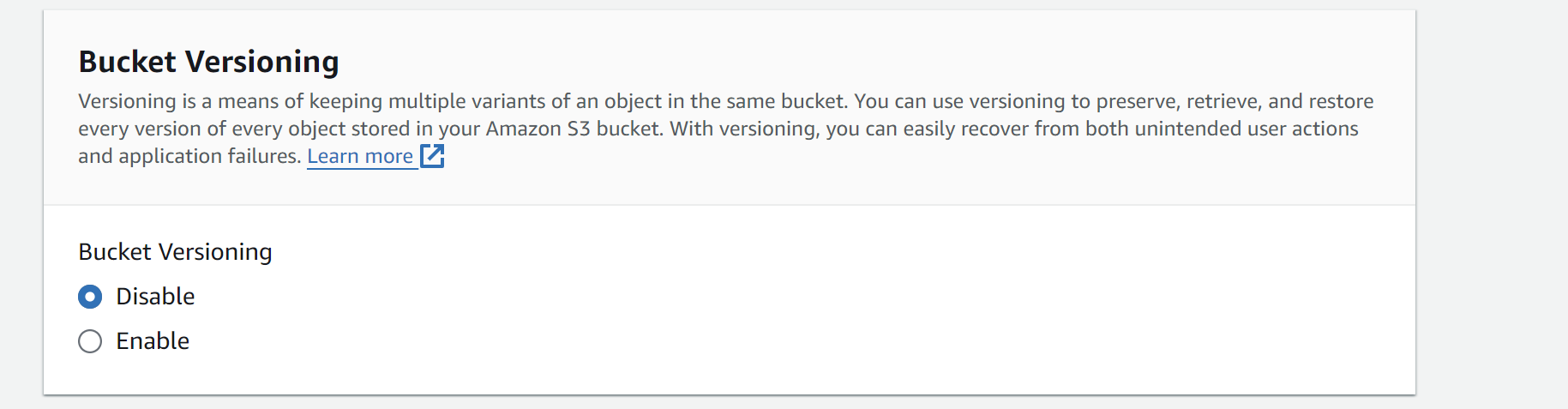
* 1. ACL Disabled: The root user own the full access to the bucket.
  2. ACL Enabled: The bucket could be shared with other aws accounts.

**3. Block all public access to bucket**



By default the bucket can only be accessed by its root user and accessible IAM accounts. Turning off this would allow access to your bucket to any AWS user.

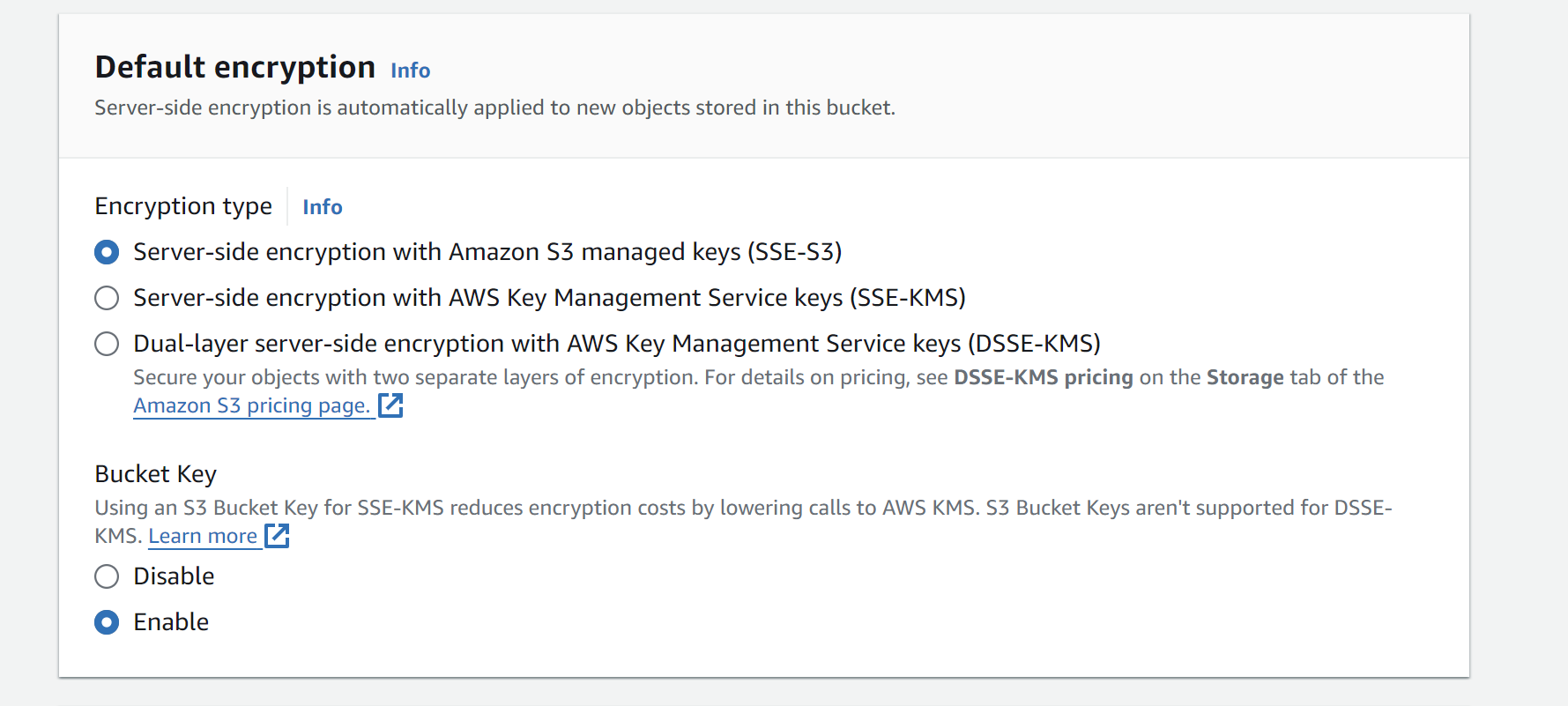
**4. Bucket Versioning:**



Using this we can store multiple variants of an object in the same bucket.

By enabling it, we can store our data in the form of versions.

**5. Default Encryption:**



Server side encryption with Amazon S3 managed keys(SSE-S3): This default option encrypts data with S3 managed keys. Each object is encrypted with a unique key. As an additional safeguard, SSE-S3 encrypts the key itself with a root key that it regularly rotates.

Server-side encryption with Amazon KMS Keys (SSE-KMS): Server-side encryption with AWS KMS keys (SSE-KMS) is provided through an integration of the AWS KMS service with Amazon S3. With AWS KMS, you have more control over your keys..

Dual-layer server-side encryption with AWS KMS( DSSE-KMS):This provides double layer protection with Amazon KMS Keys.

**6. Bucket Key**

The S3 Bucket Key is like a master key for your S3 buckets. Instead of having separate keys for every bucket, we can have one masterkey that opens any bucket.

**Advanced Settings:**

**1. Object Lock**A screenshot of a computer

Description automatically generated

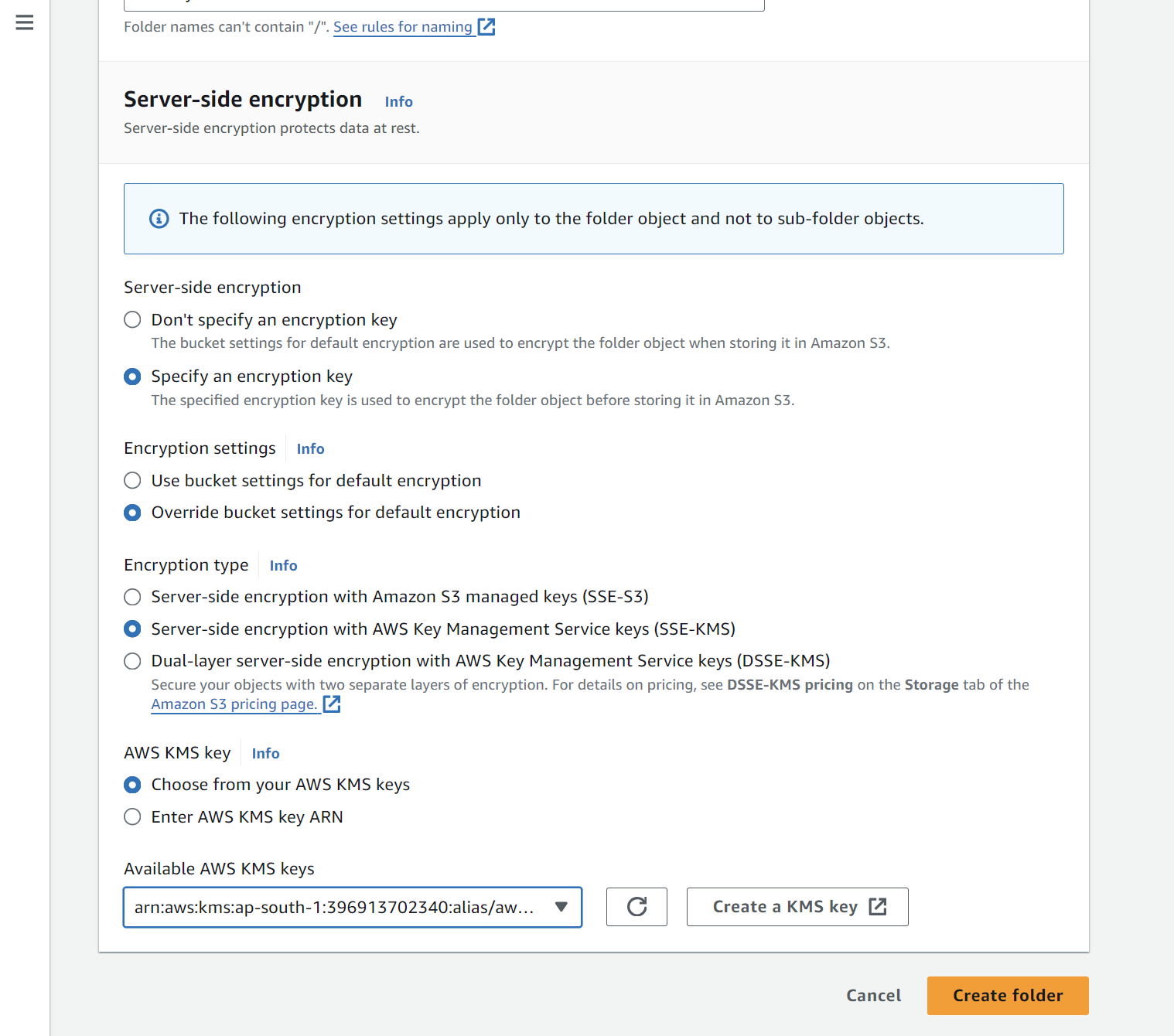
This feature allows to prevent objects in your bucket from being deleted or overwritten.

* **Organising data in Bucket**

**A screenshot of a computer

Description automatically generated**

**Creating folder with Server-side encryption**

****

Here we are creating folder with an SSE-KMS encryption type with our specific key available in our region.

Specifying ACLs:

A screenshot of a computer

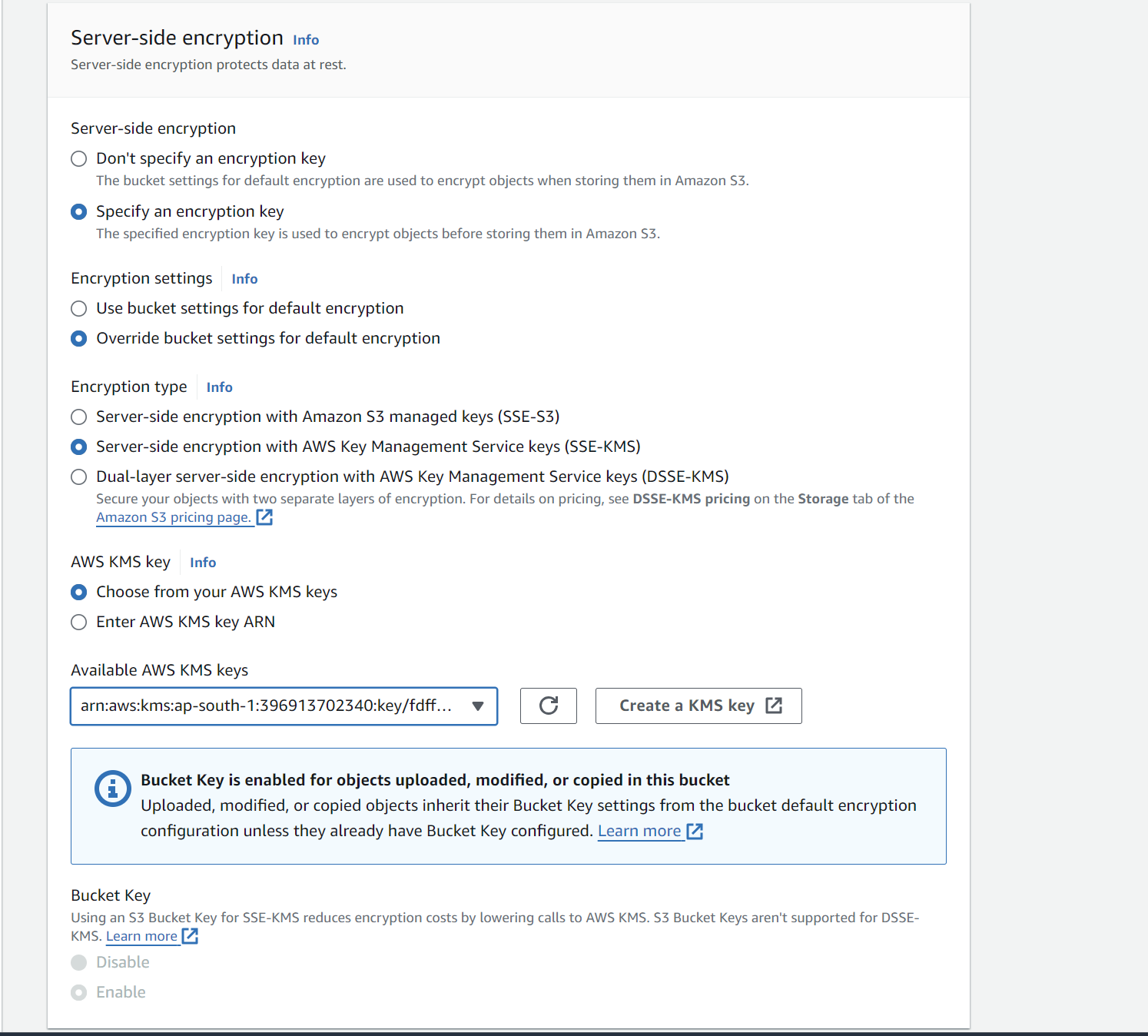
Description automatically generated

Storage Class:

A screenshot of a computer

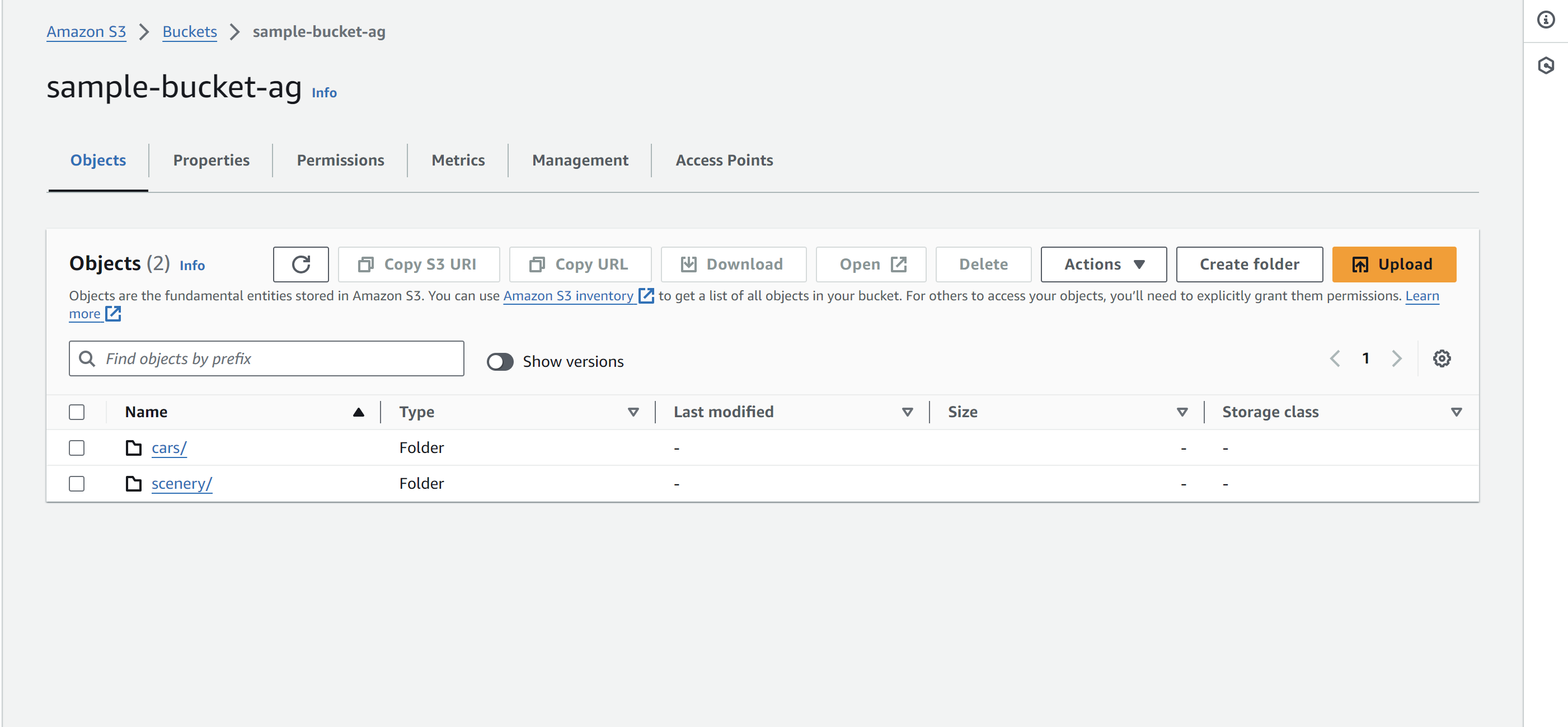
Description automatically generated

**Defining SSE-KMS for the files in the folder**

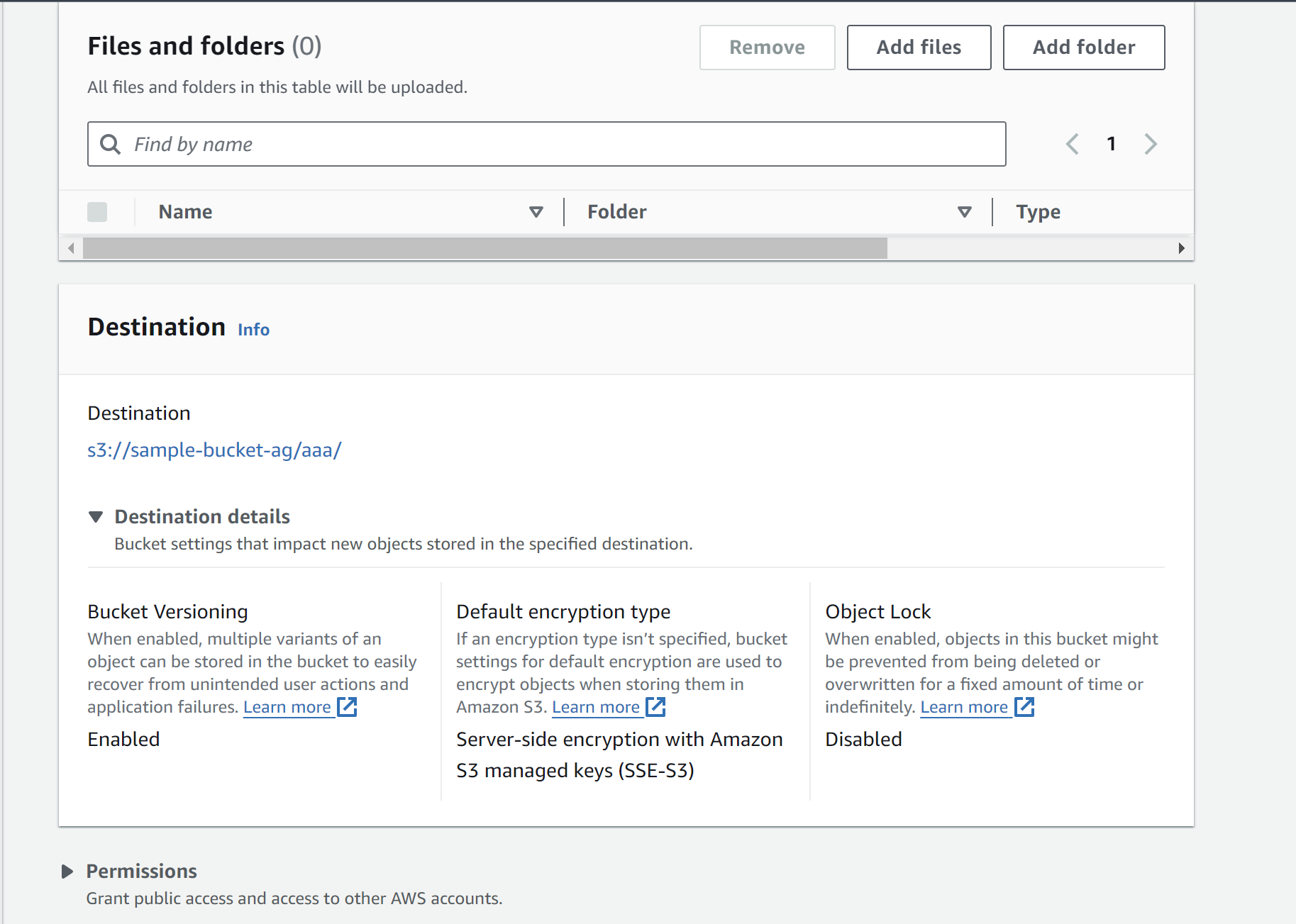


A screenshot of a computer

Description automatically generated

****

* **Data Backup:**
  1. **Bucket Versioning**

****

* 1. **Use AWS Backup**

****

* **Retrieving Backed Up Data from S3:**
* Access the Bucket: Open the Amazon S3 console and navigate to your bucket.
* Access the Object: Navigate to the folder of the deleted object. Turn on Show versions.
* 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.
* Download the File: Choose ‘Actions’, and then choose ‘Open’ or 'Download’.
* **Cleaning Up:**

Make sure to delete your S3 buckets to avoid billing up after the free-tier AWS account ends.

* **References:**

1. AWS Cloud Techinical Essentials- Coursera
2. Amazon AWS Documentation