

Guided Lab: Protecting Data on Amazon S3 Against Accidental Delete and Overwrite Using S3 Versioning

Description

Accidental deletions in Amazon S3 can happen more easily than you might think. A misplaced click, an erroneous script, or a misunderstanding in a team can lead to the loss of critical data. This reality underscores the importance of having protective measures in place.

S3 Versioning is a feature in Amazon S3 that keeps track of all versions of an object. It's like having a history of every change made to each file in your S3 bucket. Whenever you update or delete a file, S3 Versioning retains the old version, allowing you to access and restore it if needed. This is particularly useful for data recovery and backup, as it guards against accidental deletions and overwrites.

This lab will guide you through the steps to effectively implement these safeguards, providing a safety net for your data in AWS.

Prerequisites

To guarantee a successful completion of this lab, you must possess prior experience creating Amazon S3 buckets and have a solid understanding of their core components. If you believe that your knowledge in this regard is lacking, we strongly advise you to consider taking the following to acquire the required proficiency:

- Creating an Amazon S3 bucket

Objectives

In this lab, you will:

- Learn how to enable S3 Versioning to prevent overwriting of objects
- Learn how to restore previous versions

Subscribe to access AWS

PlayCloud Labs

Lab Steps

Creating an Amazon S3 bucket

1. Create an S3 bucket using the following configurations:

- Add a **Bucket Name**:

Note: Provide a unique and descriptive name for your bucket. Remember that bucket names must be globally unique across all AWS accounts. Try a different combination if you receive an error message regarding your selected name. After that, set the AWS Region to N. Virginia to place the bucket to be created there.

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

AWS Region

US East (N. Virginia) us-east-1

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 - New

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](#)

myawsbucket

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



- Next, in the **Bucket Versioning** section, make sure you **Enable Bucket Versioning**.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☐ Disable

☒ Enable



- Click **Create a bucket** to proceed in creating the S3 bucket.



Enabling Object Lock will permanently allow objects in this bucket to be locked

After you enable Object Lock for a bucket, you can't disable Object Lock or suspend Versioning for that bucket. [Learn more about Using Object Lock](#)

☐ I acknowledge that enabling Object Lock will permanently allow objects in this bucket to be locked.

[i](#) After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

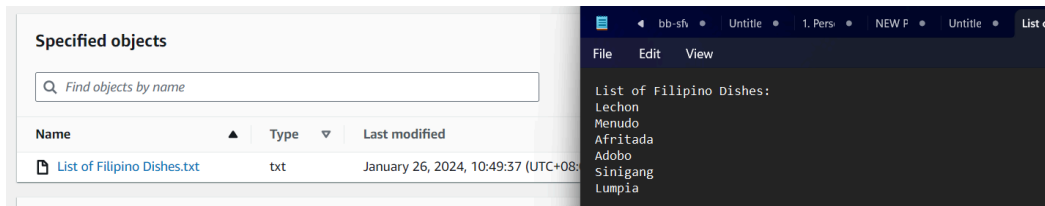


Cancel

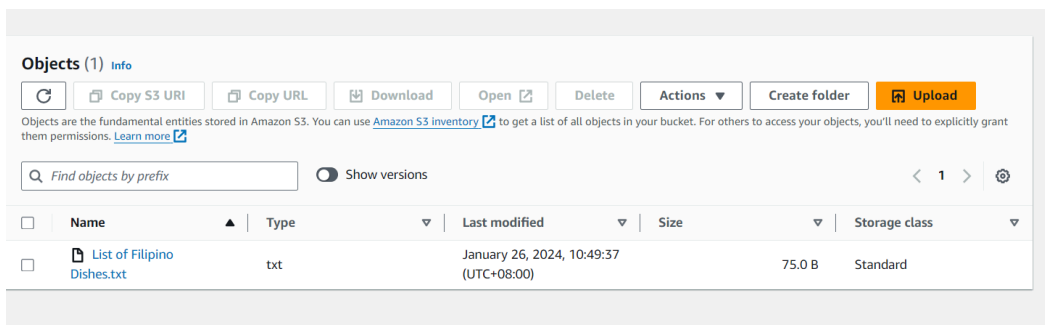
Create bucket

Restoring a file from an accidental overwrite

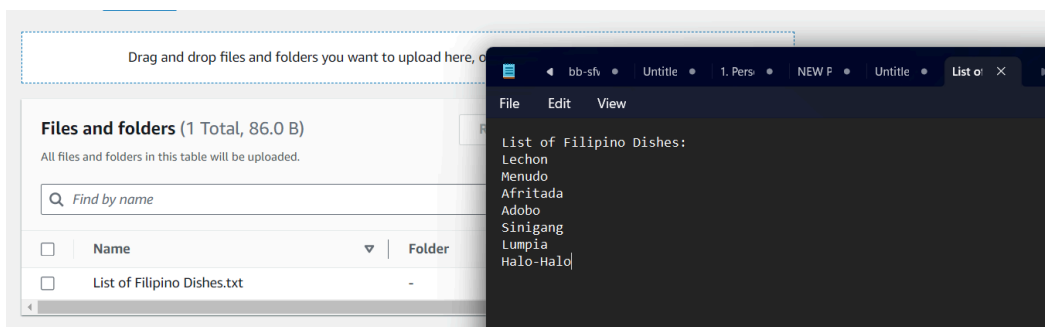
1. Create a text file on your computer containing the words “**List of Filipino Dishes**” and save it. Upload the file to the bucket. Select the **Objects** tab. Then, within the **Objects** section, choose **Upload**.



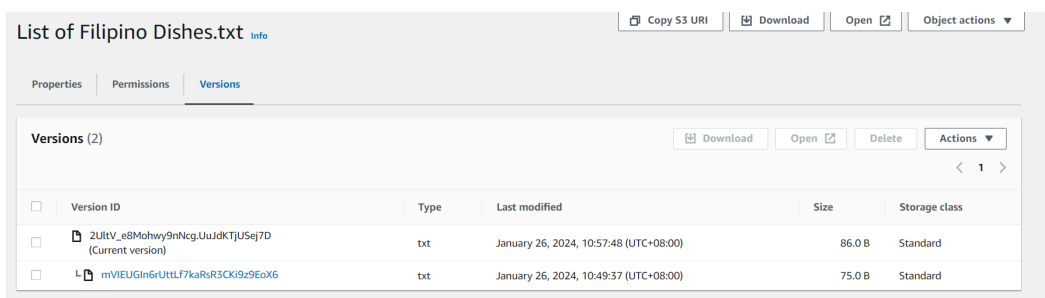
2. After finishing the file upload process, view the object in the S3 bucket.



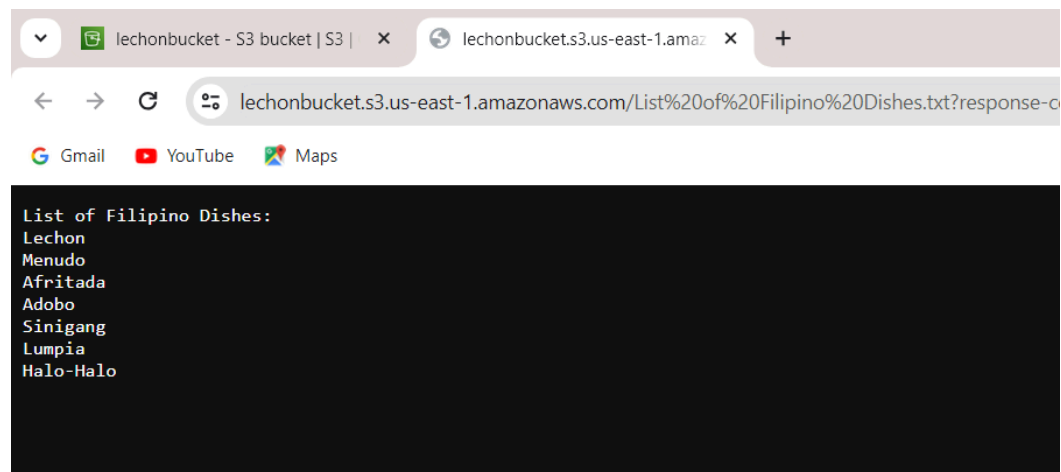
3. On your local computer, edit the same file that you created earlier. Change or add a text, then save it. Upload the updated file to S3 Bucket by repeating Step 1.



4. To view all versions of the objects stored in the bucket, select the “**Versions**” option. The console will display a list of unique version IDs for each object version, along with the date and time it was created and other relevant properties.

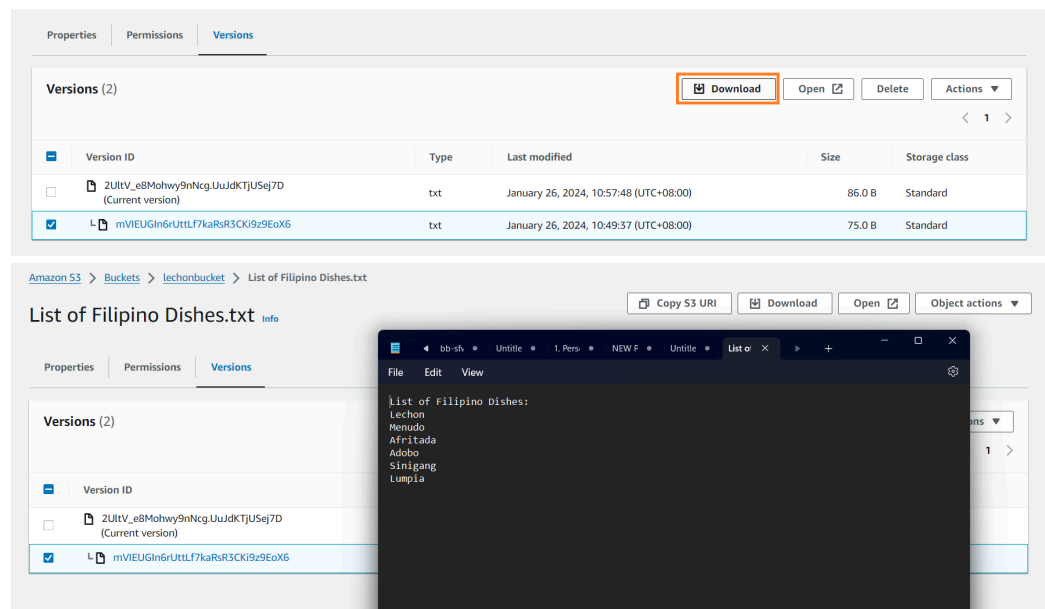


5. To confirm that the change has been saved, open the “**List of Filipino Dishes**” object, and the word “**Halo-Halo**” must be displayed.



6. In case you accidentally make changes to the file, restore the previous version of the file by:

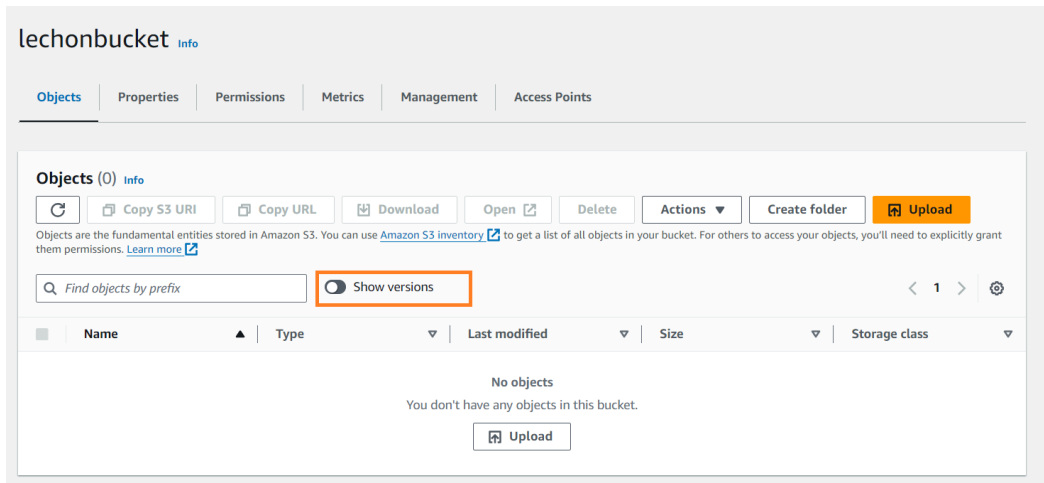
- Going to the **Versions** page of the **List of Filipino Dishes object**, select the check box next to the **Version ID** for the versions that you want to retrieve.
- Choose **Actions**, choose **Download**, and save the object.



Restoring a file from an accidental deletion

1. To simulate an accidental deletion, you need to have a deleted object first. Delete the “**List of Filipino Dishes**” object.

2. Open the bucket of the deleted object and **enable** the **Show Versions**.



Select the previous version of the object. **Note:** Don't select the **Type** with the **Delete** marker.

Then, click the Download button.

3. That's it! You've successfully restored the desired version of your file