

23.2 Simple Storage Service (S3)

AWS S3

Introduction

AWS Simple Storage Service (S3) is a storage over cloud. You can store and retrieve any amount of data and can be accessed from the Internet. It is reliable online storage which supports all types of contents.

For beginners to understand easily, it is similar to Google Drive or One Drive but with some advanced features. About the advanced features of S3 you will study in this chapter later on. AWS S3 provides developers access to the same highly scalable, fast, cheap and reliable data storage infrastructure.

AWS created S3 which is easy to use and manage. Using S3 you can create folders which is known as bucket and upload files which is known as objects. While you upload or download objects, you can apply permissions which can grant or deny access to others.

However, you know that main purpose of S3 is to store objects within buckets with permissions over public or private networks. In addition some basic purposes of S3 storage are listed below:

Static Web Site Hosting:

S3 provides facility to host a static website. The static website can have some client side scripts. AWS S3 is a cheap and highly available space to host static website. AWS S3 does not support server side scripting, its mean for dynamic website S3 is not recommended. AWS Web services provide other services to host dynamic websites.

Backup and archive:

You take backup of data where you need to use it for business continuity. And archive term used where you want to retain data for long term, compliance or research. AWS is preferred to store data because it based on **pay as you go** model.

Storage Sharing:

Using AWS S3, object owner can control permissions to share objects with others. You can share URL of the object with others to provide access. By default all objects are private.

Disaster Recovery:

AWS S3 supports disaster recovery (DR) architecture which is suitable for small customer too. AWS S3 is highly durable and secure, supports cross region replication.

Comparing Object storage and Traditional storage

Traditional storage:

In general practice you find two different type of traditional storages.

- a. Block Storage

b. File Storage

Block Storage operates at a raw storage device level and it comes with fixed size blocks. However file storage works at the operating system level and it manages data in a hierarchical order of files and folders.

Examples of block storage: Storage Area Network using Internet Small Computer System Interface (iSCSI) protocol.

Example of File Storage: Common Internet File

System (CIFS) Or Network File System (NFS)

Object Storage:

AWS S3 is not dependant on any server, it is accessible over the Internet. It is different from the traditional storage. It contains both data and metadata, and objects reside in S3 buckets.

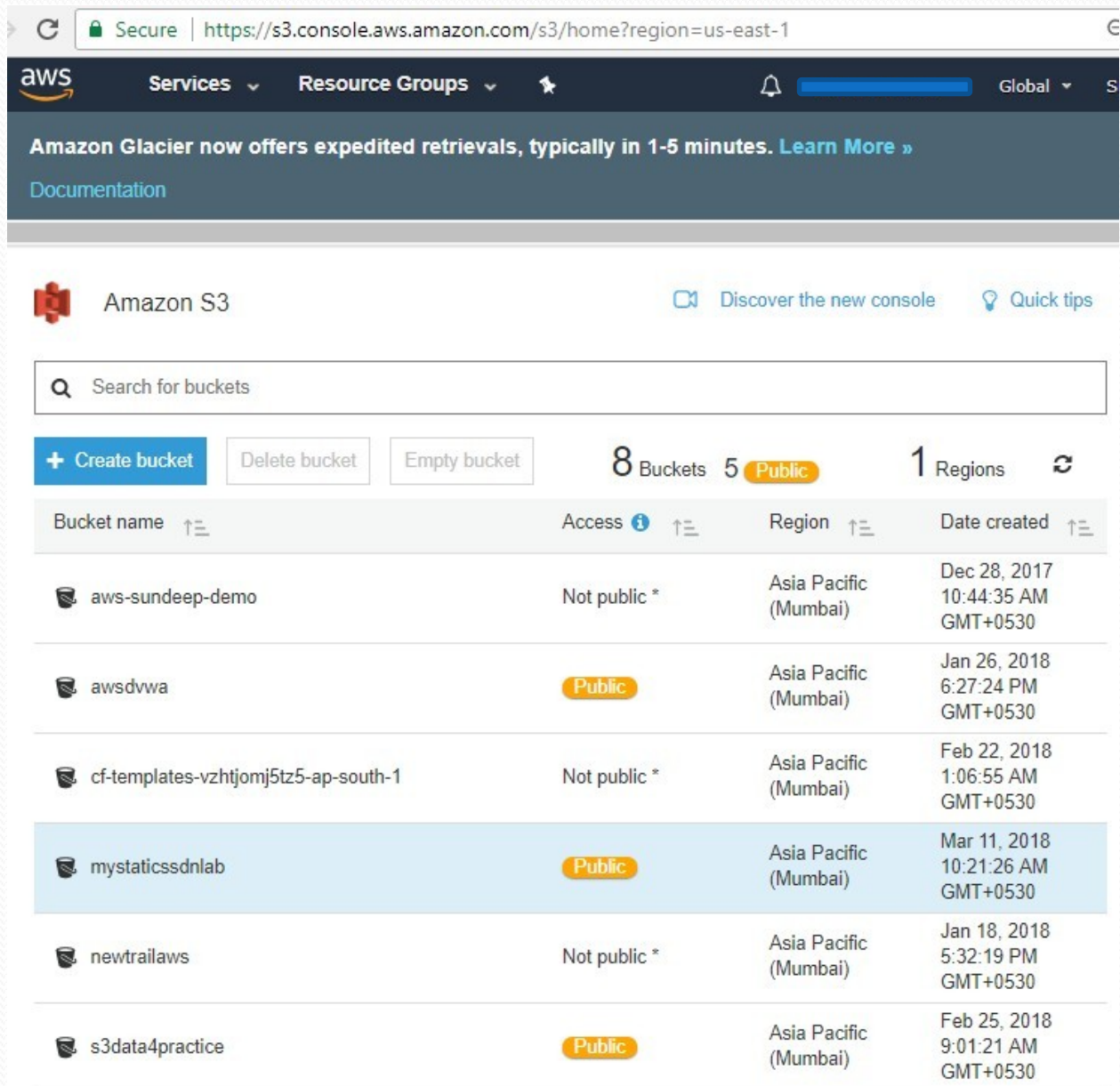
Basic components of S3

Buckets

A bucket in S3 acts like a web folder or container, in which you can store objects (files). Every bucket in AWS S3 is global, it means, the bucket name must be unique across all AWS accounts. By default you can have up to 100 buckets in your AWS account. But you can't create nested buckets.

For example: If the object named **photos/mypicture.jpg** is stored in the **ironsmith** bucket, then it is addressable using the URL

<http://ironsmith.s3.amazonaws.com/photos/mypicture.jpg>



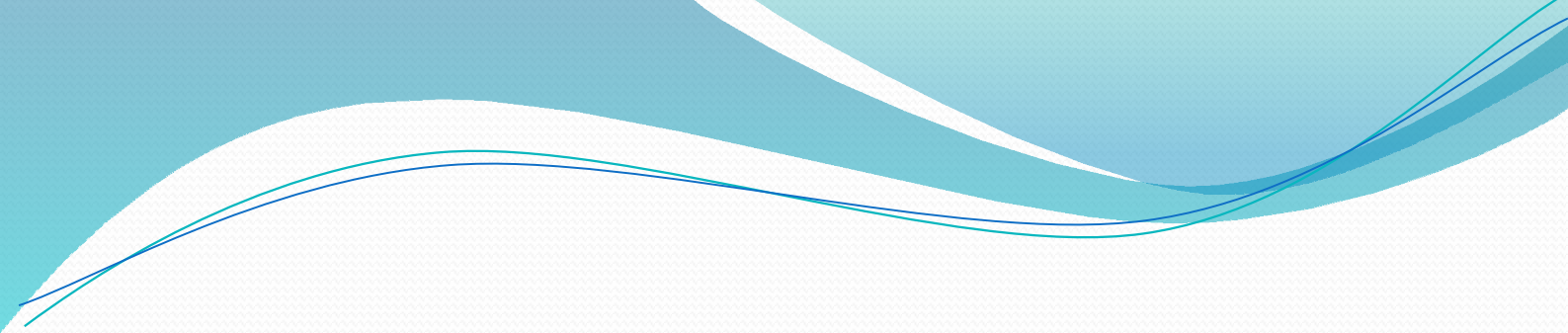
Bucket name	Access	Region	Date created
aws-sundeep-demo	Not public *	Asia Pacific (Mumbai)	Dec 28, 2017 10:44:35 AM GMT+0530
awsdvwa	Public	Asia Pacific (Mumbai)	Jan 26, 2018 6:27:24 PM GMT+0530
cf-templates-vzhtjomj5tz5-ap-south-1	Not public *	Asia Pacific (Mumbai)	Feb 22, 2018 1:06:55 AM GMT+0530
mystaticssdnlab	Public	Asia Pacific (Mumbai)	Mar 11, 2018 10:21:26 AM GMT+0530
newtrailaws	Not public *	Asia Pacific (Mumbai)	Jan 18, 2018 5:32:19 PM GMT+0530
s3data4practice	Public	Asia Pacific (Mumbai)	Feb 25, 2018 9:01:21 AM GMT+0530

Objects

Objects are fundamental entities or files stored in AWS S3 buckets. Objects contains data and metadata about the files. AWS S3 does not differentiate about what kind of data is stored in objects, data can be in text or binary form.

Object metadata in AWS S3 are categorized in two kinds:

System metadata: It is created by AWS S3 for the object. For example object creation date, size, HTTP content type, etc.



User defined metadata: for example S3 has *storage class* of an object such as STANDARD, STANDARD_IA, GLACIER, etc. You can use custom metadata to tag data.

Keys

This is the most significant part of the object. A key of an object represents the unique identifier within a bucket. Every object has a key. Because the combination of a bucket, key, and version ID is unique for each object, Amazon S3 can be thought of as a basic "bucket + key + version" and the object itself.

Every object in Amazon S3 can be uniquely addressed through the web service endpoint, bucket name, key, and optionally, a version.

For example, in the object URL

[01/AmazonS3.wsdl](#), "doc" is the name of the bucket

01/AmazonS3.wsdl is the key.

The screenshot displays the AWS S3 console interface. At the top, the navigation bar shows 'aws', 'Services', 'Resource Groups', and a search icon. Below this, the breadcrumb path is 'Amazon S3 > s3data4practice'. The object name 'index.html' is shown with a 'Latest version' dropdown. A tabbed interface at the bottom has three tabs: 'Overview', 'Properties', and 'Permissions'. Below the tabs are five buttons: 'Open', 'Download', 'Download as', 'Make public', and 'Copy path'. The 'Properties' tab is active, showing the following details:

- Owner:** da1d6c0462a7ac645fe4ae6b08527859fa12d02e5e42cd65fe806c9655e90a94
- Last modified:** Feb 25, 2018 10:16:55 AM GMT+0530
- Etag:** c259bd4a613e7bf13289b91051e0059c
- Storage class:** Standard
- Server side encryption:** None
- Size:** 65
- Link:** <https://s3.ap-south-1.amazonaws.com/s3data4practice/index.html>

An object key can be maximum up to 1024 bytes which and dots, slashes dashes.

Functions of AWS S3

AWS S3 is simple and handful with common

operations as follows: Create/delete a bucket

Read/Write an object

Delete an object

Display keys in a

bucket

AWS S3 Data Consistency Model:

Amazon S3 provides read-after-write consistency for PUTS of new objects in your S3 bucket in all regions with one caveat. The caveat is that if you make a HEAD or GET request to the key name (to find if the object exists) before creating the write, Amazon S3 provides eventual consistency for reads.

Amazon S3 offers eventual consistency for overwrite PUTS in all regions.

Updates to a single key are atomic. For example, if you PUT to a key and then make a subsequent read, you might return the old data or the updated data. Amazon S3 never writes corrupted or partial data.

From <

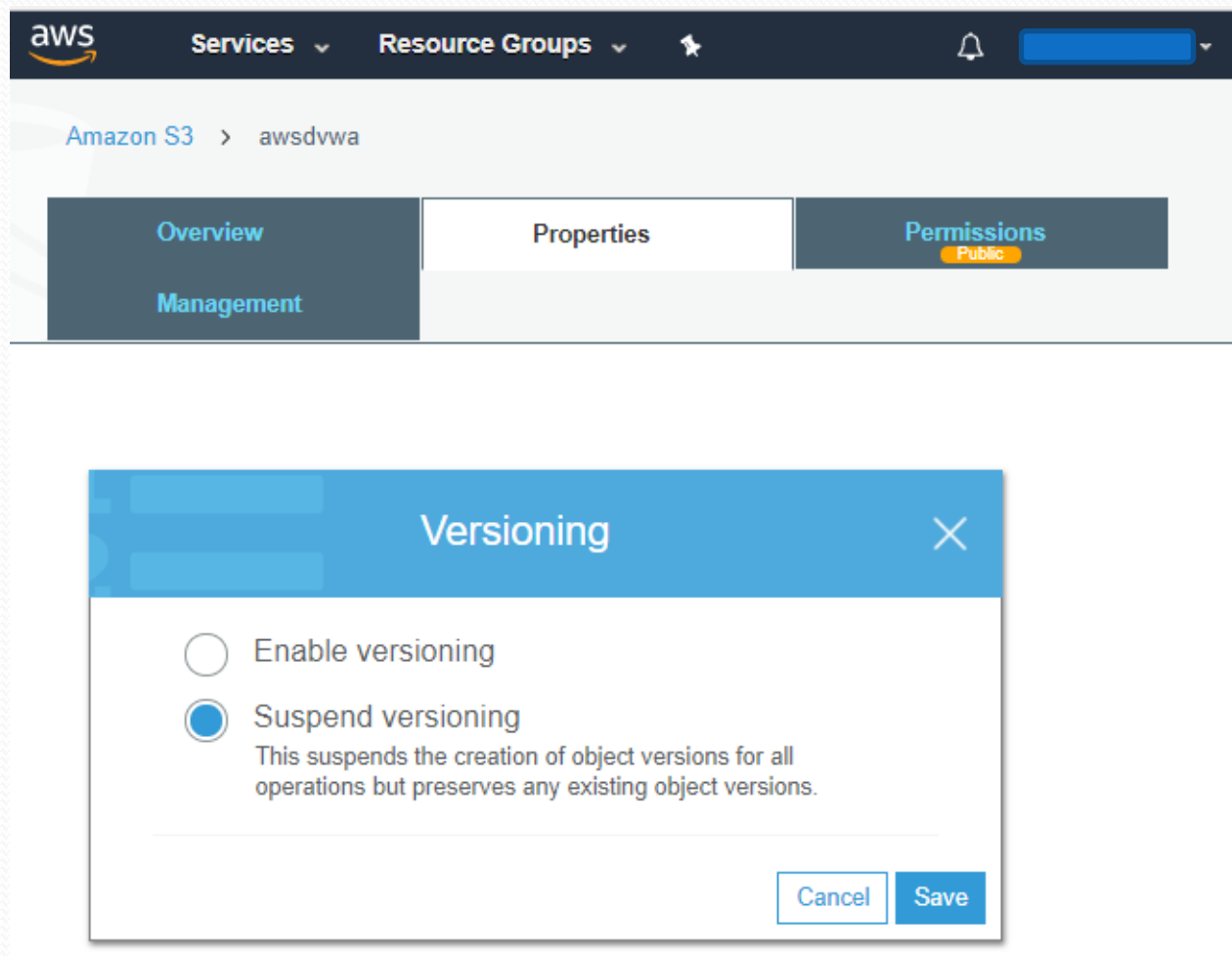
<http://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html>

>

S3 Bucket Features

Versioning

S3 versioning features enables you to keep multiple variants of an object within the same bucket. It is very useful to preserve, retrieve and restore each version of objects stored in AWS S3 buckets. It is also useful to recover objects, if objects from a bucket are deleted accidentally. Consider a case, you have an object stored in S3 bucket, and versioning is turned on for the bucket, if you upload a modified version of the object to the same bucket. You will have even the old version of the object which can be retrieved any time.



Important: Once versioning is enabled for a bucket, it can only be suspended, it can't be removed.

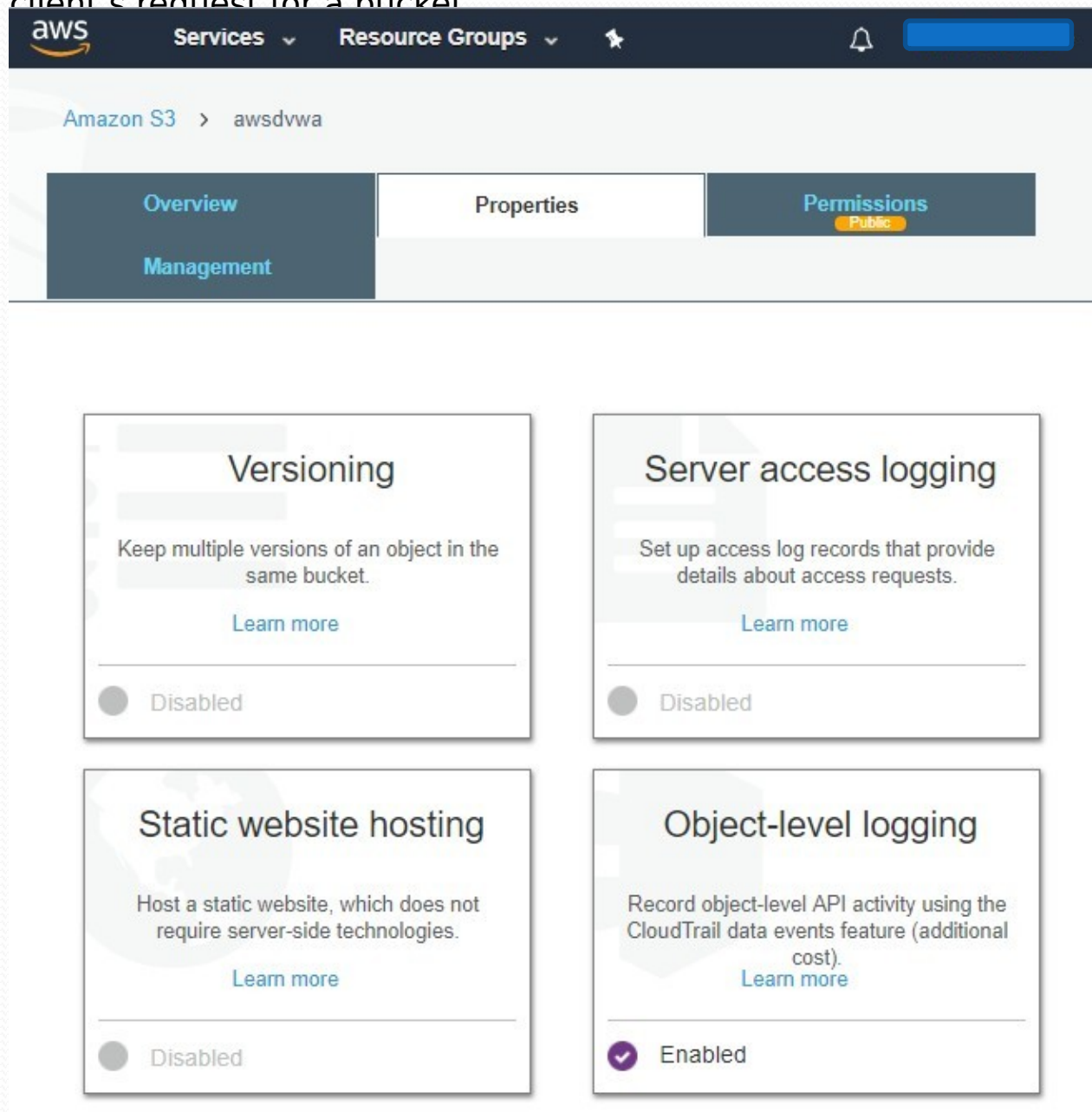
Multi Factor Authentication (MFA) Delete

Multi Factor Authentication Delete is an additional layer of security for your data stored in S3 bucket. It required 6 digit number via authentication device to delete an object or change the version state of a bucket.

Logging:

As you know logging consist a list of activities about service or server. But AWS S3 does not collect service access logs. For developers and administrators server access logs are very useful, because logs provides insight natures of requests made by users/clients.

By default logging for the bucket is disabled, but when you enable it, an access log record collect details about the client's request for a bucket



Object Level Logging :

The CloudTrail data events feature is enabled for this bucket. Go to the CloudTrail console to view and configure the settings for trails.



Server Access Logging:

Set up access log records that provide details about access requests.

Encryption

Data Security point of view encryption is important feature. AWS S3 allows you to encrypt your data stored in bucket both at rest and in transit. To encrypt data in transit secure socket layer (SSL) and client side encryption (CSE) is recommended.

Event Notifications

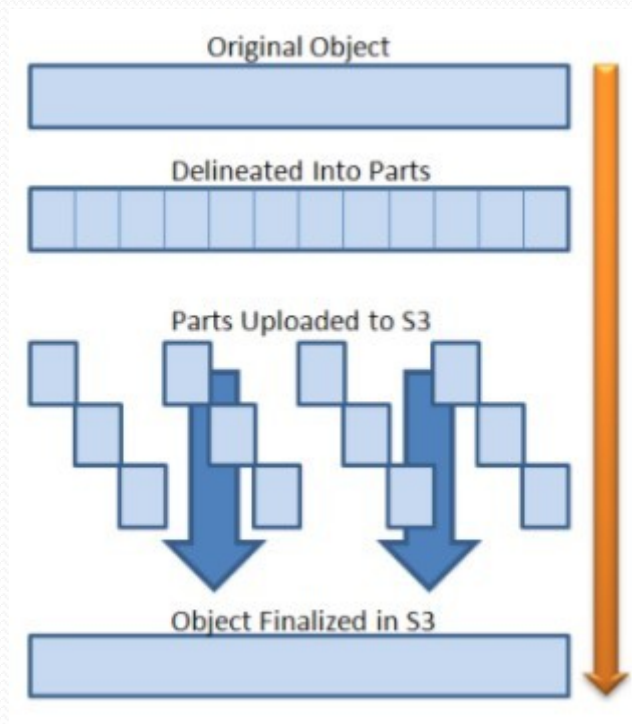
Event notification allows to send alerts, run workflows or trigger actions in response of condition match.

Multipart Upload

Any object which size is more than 100 MB is recommended to upload using multipart upload API. For better network utilization and through parallel transfer of object, multipart upload plays a significant role. This facilitates you to upload objects of large size as a set of parts which gives better performance. This also gives ability to pause and resume upload objects. The current limit of maximum size to upload on S3 is 5 GB.

Multipart upload process defined in three steps:

- a. Initiation
- b. Uploading the parts of objects
- c. Completion or Finalization

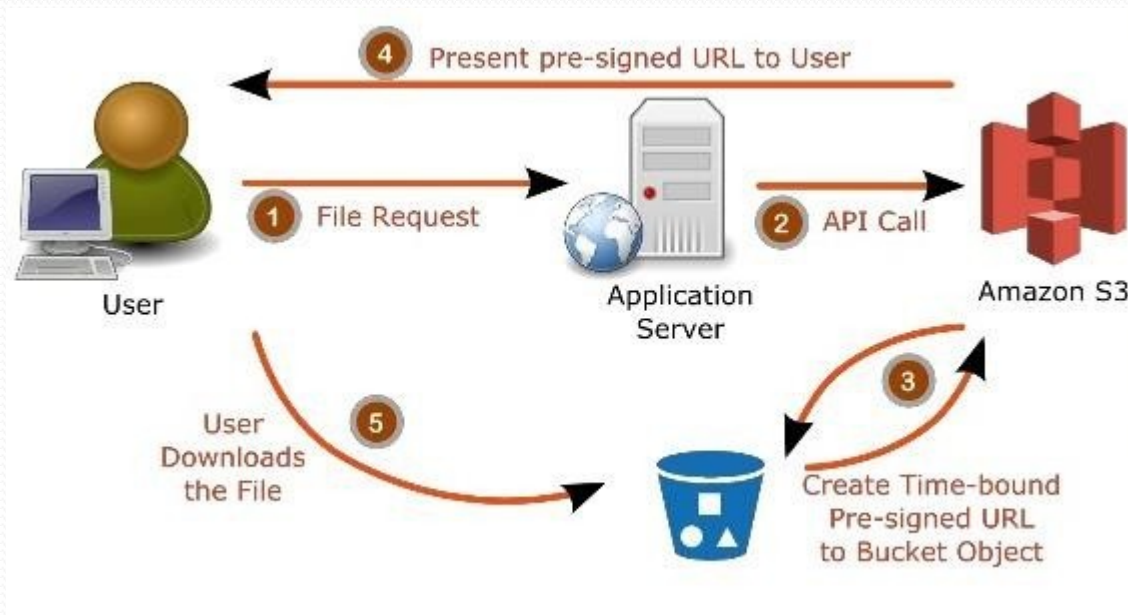


Not

S3 Browser is a freeware Windows client for [AmazonS3](#) and [Amazon CloudFront](#). Amazon S3 provides a simple web services interface that can be used to deliver your files using a global network of edge locations. You can download S3 Browser from the link <http://s3browser.com/>. It can be

Pre-Signed URLs

Every object uploaded in S3 bucket has a URL and this is private, and owner of the object has permission to access. Later, the owner can provide permissions to access the object. This means the object owner can create and share pre-signed URL, using security credentials. In addition to limit the download for a particular time, the owner can grant time-limit. This way pre-singed URLs are valid for specified time.



Lifecycle Management

Let us understand the concept of lifecycle management through simple examples:

Sometimes to see the business requirements, data of organization at initial period seems important and is being used frequently. After certain period of time you feel that you might not need frequent access to this data. But your organization wants you to archive them for later use.

Lifecycle management configuration is a group of rules which defines how S3 handle objects on your behalf?

Object lifecycle management actions are categorized as follows:

Transition actions

For example, you want to transfer data to the STANDARD-IA storage class after 15 days of creation of data or it can be transferred for archival to the GLACIER storage class after 90 days of creation.

Expiration actions

You set expire date for the objects, later AWS S3 can delete those objects on your behalf.

Replication

You use cross region replication in order to transfer objects from source bucket to destination bucket. This supports automatic and asynchronous copying of objects across buckets. To enable this feature you must add a replication configuration to your source bucket to direct S3 to replicate.

Pre requisite to start cross-region replication:

The source and destination bucket should be enabled for versioning Both buckets must be on different region

S3 must have permission to replicate objects. If not, you can use IAM to create role.

In order to understand practically, you must follow the activity. The activity link is

<https://docs.aws.amazon.com/AmazonS3/latest/dev/crr-walkthrough1.html>

Outline