S3 Buckets

S3

Advertised as infinitely scaling storage. Many AWS services also use S3 as part of its service.

You can use S3 for:

1. Backup
2. Storage
3. Disaster recovery
4. Archive
5. Static website
6. Software delivery
7. Data lakes and big data analytics

The objects are stored into buckets (think of it as directories). Each bucket that you create must be globally unique (across all regions and accounts). However, the buckets are per region. No uppercase no underscore restrictions for naming buckets.

Each file stored into a bucket have a key, the key for S3 is the full path. However, S3 does not have directories concept! If you have folders then it will be named as prefixed. The object key contains prefixed (which can have folder path) + the actual name file itself.

The values that the key mapped to contain the content of the file itself, max at 5TB. If you're uploading a file more than 5GB must use multi-part upload.

S3 Bucket security

1. User-based rules: Attach IAM policies to specify which API call to S3 bucket the use can make to a user.
2. Resource-based: Have three types
   1. Bucket policies: Bucket wide rules from the S3 console, this allow cross account to let other account from AWS to access it
   2. Object access control list: finer grain
   3. Bucket access control list: less common now

So basically you can attach policy to the user to give it permission to the S3 bucket. Or you attach the policy to the S3 to allow who is able to access it. You can also add IAM role to resources to give it permission so that say an EC2 instance can access it. You can also add cross account to grant it access to the bucket.

IAM principal (the who) can access and S3 object if the user IAM allows it or resource policy allows it and there is no explicit deny

You can also encrypt the object using encryption keys to add more security.

S3 static website hosting

S3 can be used for hosting static websites. The static website URL will be the bucket's public URL. In order to make this work you must configure the bucket to be publicly accessible.

S3 versioning

You can enable versioning for your files in S3. Any changes over the same key will create a version for that particular key. For example if you upload the same file twice it will just overwrite the file with version 2.

You can rollback to the previous version to prevent from accidental corruption or deletion. Files that aren't versioned have version null if you didn't enable it first.

S3 replication

Must enable version in the source and destination bucket that you are creation replication to. The buckets could be in other AWS account.

The replication occurs asynchronously.

There are two flavors:

1. CRR (Cross region replication): Compliance, lower latency access, replication across accounts
2. SRR (Same region replication): Log aggregation, live replication between production and test accounts

Only new objects are replicated, old objects won't be replicated you must use S3 batch replication to do so.
Deletions with version ID are not replicated. But deleting the entire object will be replicated since it only place a delete marker.

S3 storage classes

Durability: What are the chances of your object being lost. On average single object is loss once every 10,000 years if you store 10,000,000 objects. This durability is the same for all storage.

Availability: How readability available the S3 is. On average down 53 minutes per year.

General purpose

This is used for frequently accessed data. Use it for big data analytics, mobile and gaming application. No cost for retrieval, only for storage.

Infrequent access

For data that is less frequent accessed, but that needs rapid access sometimes.

Has lower cost than S3 standard but has cost when retrieving.

Standard-infrequent access: Use for disaster recovery and backups

One Zone-Infrequent Access: Very high durability in a single availability zone, but the data can be lost if the AZ is destroyed. Use it for storing secondary backup

Glacier

Low cost when storing for archiving and backup.

You will be paying for low storage price but high object retrieval cost

Glacier instant retrieval: Give you millisecond get time. Good for data accessed once a quarter.

Glacier flexible retrieval: Expedited (1-5 minutes), standard (3-5 hours), bulk (5-12 hours) to get data back

Glacier deep archive: Meant for long archive. Standard (12 hour), bulk (48 hours)

Intelligent-tiering

Let you move objects automatically based on access pattern. No retrieval charges. But you need to pay monitoring and auto-tiering fee.

1. Frequent access tier: default tier
2. Infrequent access tier: if obj hasn't been access for 30 days
3. Archive instant access tier: 90 days
4. Archive access tier: 90 days to 700+ days
5. Deep archive access tier: 180 to 700+ days

Lifecycle rules

You can transition objects between storages. Moving object is done by life cycle rules. You specify the storage class you want to move the object to, and the amount of days that it has to pass for the transition to occur.

You can also setup expiration actions, say delete the object after some times. You can also delete incomplete multi-part uploads.

You can also move non-current version of the object to other storage classes, doesn't have to be current versions.

S3 analytic will give you some recommendation to tell you when to transition objects to the right storage class: Help you put together life cycle rules that make sense!

Requester pays

Owner of the S3 bucket pay for storage + downloading cost, but there is the option to make the requester who is asking for the file to pay for the network cost.

S3 event notification

When you do something with S3 bucket, whether is depositing a file, or a file gets deleted, it will sent out events. Now these events can go to couple of places:

1. SNS topic
2. SQS queue
3. Lambda function
4. Event bridge: This is the latest one, you set this one to sent all events to Event Bridge then you can forward it to other AWS services. This expand on the only 3 that S3 can sent the event to. It enhances!

Services can then react to these events to say do something with the file that was just deposited.

S3 performance

It scales automatically to high request rate with 100-200ms latency. You get lots of requests PER prefix. Remember prefix is like "directories" even though S3 doesn't have a directory concept.

So if you have like 2 folders, then they each get a set of request rates per prefix.

Use multi-part upload for files > 5GB, recommended for 100MB. This speed up file upload by taking advantage of parallel uploads. You do this by dividing your big file into parts then upload each parts in parallel, S3 can reconstruct the parts into the big file after.

S3 Transfer acceleration is used if you need to transfer files across region. This is done by uploading your file to edge location which then can forward the file to the S3 bucket via a private network which is much faster than public internet to the bucket in another region. (Remember public internet requires packet hopping across different routers).

S3 Byte-range fetches is like multi-part upload but for download, to speed up download. You can request specific byte range of the file in parts, say byte 0-50, byte 51-100, and so on. This is so that you can take advantage of parallel downloads. Partial byte range can be retrieved as well.

S3 batch operations

Make lots of operation on existing S3 objects with one request.

- Encrypt, un-encrypt objects
- Restore objects from S3 glacier
- Modify ACLs
- Invoke lambda to do whatever function you want on the object

So the batch operation contain jobs which contain the cation to take and object to take it on

S3 batch operation have built-in retries, progress tracking, and generate reports.

Can use S3 select to do filtering on the object that you want to perform the operation on.

Revision #5
Created 16 February 2023 16:03:40 by Tamarine
Updated 18 February 2023 23:25:13 by Tamarine