Amazon Simple Storage Service (S3)

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

Simple Storage Service (S3)

- Cloud Based Storage Service
- Store and Retrieve from anywhere on the web
- Unlimited Storage
- Very large size limit for individual objects – up to 5TB
- Secure – Strong access control policies, encryption
- Reliable and Inexpensive. Pay only for what you use
- Lifecycle management

Usage Examples

- Content Distribution - Documents, Video, Images
- Log Storage - CloudWatch Metrics, Application Logs, CloudTrail logs
- Backup and Long-term storage – Database, Volume backup, content backup
- Source or Destination for several AWS Services – Machine learning, Transcoding, Hadoop, Lambda
- Buffer to allow services process at different rates

Buckets

- Container for objects
- Created in a specific region
- Globally unique
- Enforce Access control

Object

- Fundamental entity stored inside a bucket
- Conceptually equivalent to files
- Consists of a Key, Data and Metadata
- Key – uniquely identifies the object in a bucket
- Data – customer data
- Metadata – contextual information including content-type, date last modified and others

Key

- User defined unique identifier for an object (example: file name)
- Combination of bucket name and key is globally unique

Region

- AWS Region where bucket is created
- Object stored in a region never leaves the region
- You can ask S3 to replicate to a different region

Consistency Model

- S3 is a distributed system
- Objects are stored redundantly in multiple devices across multiple facilities in a region
- When you create a new object or update an object, it takes time to store or update all redundant copies
- Consistency Model defines the behavior that you can expect when reading data back during propagation

New Objects

- read-after-write consistency

"With this enhancement, Amazon S3 now supports read-after-write consistency in all regions for new objects added to Amazon S3. Read-after-write consistency allows you to retrieve objects immediately after creation in Amazon S3"

- Until change is fully propagated, object key may not appear in available list of objects in a bucket
- However, if you check the object Key Name using HEAD or GET before adding, S3 offers eventual consistency

Overwriting an Existing Object

- Eventual Consistency
- Until change is propagated, users may see:
  - Prior data
  - Newer data
- S3 will never return a mix of prior and newer data

Deleting an Existing Object

- Eventual Consistency
- Until change is propagated, users may see:
    - Prior data
    - No data
    - Object key may appear in available list of objects in a bucket

Consistency Model

Example: Walk through

Versioning

- Keep multiple versions of an object in a bucket
- Protection against accidental overwrites, deletions
- Retrieve previous versions of objects
- Enabled or Suspended at bucket level
    - Applies to all objects in the bucket
    - Once enabled can only be suspended
- Object Key and Version ID uniquely identifies an object in version enabled bucket
- Version ID is null for existing objects – all changes get a new unique Version ID

Versioning

Scenarios Walk Through

More Scenarios

Restoring Previous Version

Demo

1. Non-versioned bucket upload/delete
2. Enable Versioning
3. Upload Object
4. Delete Object
5. Delete a delete marker

Storage Class

Standard Storage (STANDARD)

- Ideal for performance sensitive use cases
- Frequently accessed data (Hot Data)
- Real time access
- Default storage class for objects

Standard Infrequent Access (STANDARD_IA)

- Performance similar to Standard
- Less frequently accessed (Warm Data)
- Real time access when needed
- Additional Retrieval fee
- Lower cost compared to Standard
- Long-lived data (backup or older data)
- Minimum size: 128KB and Duration: 30 days

Glacier (GLACIER)

- Archived data (Cold data) - No real time access
- Need to restore objects
  - Restored to RRS storage
  - Kept for specified number of days and removed
- Restore takes few minutes to several hours
  - Expedited – 1 to 5 minutes (< 250MB object size)
  - Standard (default) – 3 to 5 hours
  - Bulk – 5 to 12 hours. Very large restores (petabytes)
- Each Glacier object uses 8 KB Standard storage and 32KB for Glacier metadata

One Zone Infrequent Access (S3 One Zone-IA)

- Launched in 2018
- 20% lower cost than Standard IA
- Intended for re-creatable data and for uses that does not require multi-AZ resiliency
- Can be used in Life Cycle Transitions
- Example Usage: Secondary backup of on-premises data, Target storage class for Cross Region Replication

Introducing One Zone IA

One Zone IA Comparison

Reduced Redundancy Storage (RRS)

- Storage for Non critical, easily reproducible data
- Hot or Warm data
- Real time access when needed
- Fewer redundant copies stored when compared to Standard storage – can configure event notification when object is lost
- 400 times the durability of typical hard disk drive
- Cheaper when compared to Standard storage

Storage Class	Standard	Standard–IA	One Zone-IA	RRS	Glacier
Usage	Frequently Accessed Data	Less Frequently Accessed Data	Less Frequently Accessed Non Critical Data	Frequently Accessed Non critical data	Rarely Accessed. Data Archiving
Durability	99.999999999%	99.999999999%	99.999999999%	99.99%	99.999999999%
Availability	99.99%	99.9%	99.5%	99.99%	N/A
Availability SLA (Service Credit)	99.9%	99%	99%	99.9%	N/A
Concurrent Facility Failure	2	2	Stored only in 1 AZ	1	2
Redundancy (Region)	Multiple devices in multiple AZ	Same as standard	Multiple devices in single AZ	Fewer copies	Same as standard
First Byte Latency	Milliseconds	Milliseconds	Milliseconds	Milliseconds	4 hours
Minimum Storage Duration	N/A	30 days	30 days	N/A	90 days
Minimum Size		128 KB	128 KB		
x-amz-storage-class	STANDARD	STANDARD_IA	ONEZONE_IA	REDUCED_REDUNDANCY	GLACIER

Billing

Example	Quantity	Standard	Standard–IA	One Zone-IA	RRS	Glacier
Storage	100 GB	$2.3 per month	$1.25 per month	$1 per month	$2.4 per month	$0.4 per month
PUT, COPY, POST, or LIST	1,000 requests	$0.005	$0.01	$0.01	$0.005	$0.05 for 1,000 archives or restores
GET and all other	1,000 requests	$0.0004	$0.001	$0.001	$0.0004	
Data Retrievals, Restores	NA	NA	$0.01/GB	$0.01/GB	NA	Expedited - $0.03 per GB Standard - $0.01 per GB Bulk - $0.0025 per GB
Data Transfer IN to S3	Free					
Data Transfer OUT from S3 To						
Same Region	Free					
Different Region	$0.02/GB					
CloudFront	Free					
Internet	Free Up to 1GB/month. $0.09 per GB					

Changing Storage Class

- Storage class is specified when Object is created
  - Default is S3 Standard Storage Class
- For existing objects:
  - Copy to the same bucket with same key and Specify new storage class
  - Use Lifecycle policies to change storage class

Demo

1. Specify storage class when storing object
2. Change storage class for existing object

Performance

Performance

- Max size for a single object is 5 TB
- Max upload in a single PUT is 5GB

Question: How to handle transfer failures when transferring large objects?

- For PUTs larger than 100 MB, multipart upload capability should be used
- For GETs option to retrieve object in parts

Multipart

- Upload object in parts
- Upload parts in parallel
- Restart failed load of a part due to errors
- Pause and Resume
- Begin upload even before you know the final size – upload as data is available

Multipart Steps

1. Initiate multipart upload (unique upload ID)
2. Upload object parts (each part has a number between 1 to 10,000)
3. Complete multipart upload
4. S3 after receiving completed call combines all the parts to form the object

AWS SDK has easy to use classes to automatically perform multipart upload, download

Performance

For very large number of GETs - Consider using CloudFront Content Distribution Network

- Distributes content to location closest to your Users
- Reduces call to S3, which will lower cost - S3 charges depend on number of request

Performance – Key Naming

For workloads exceeding

- Over 100 PUT/LIST/DELETE per Second
- Over 300 GET requests per second

Avoid using common prefix, sequential numbers or date time patterns in key names

- Increases likelihood of storing in the same partition for large number of keys
- Can overwhelm I/O capacity of the partition

Add a random prefix to the key name

- Adding randomness to prefix more evenly distributes key names across multiple index partitions

Lifecycle Management

Lifecycle Management

- Manage lifecycle of objects
- Transition objects to different storage class - Age based
- Expire objects – Age based
- Cleanup incomplete multipart uploads
- Support for versioned objects

Lifecycle Management Scenarios

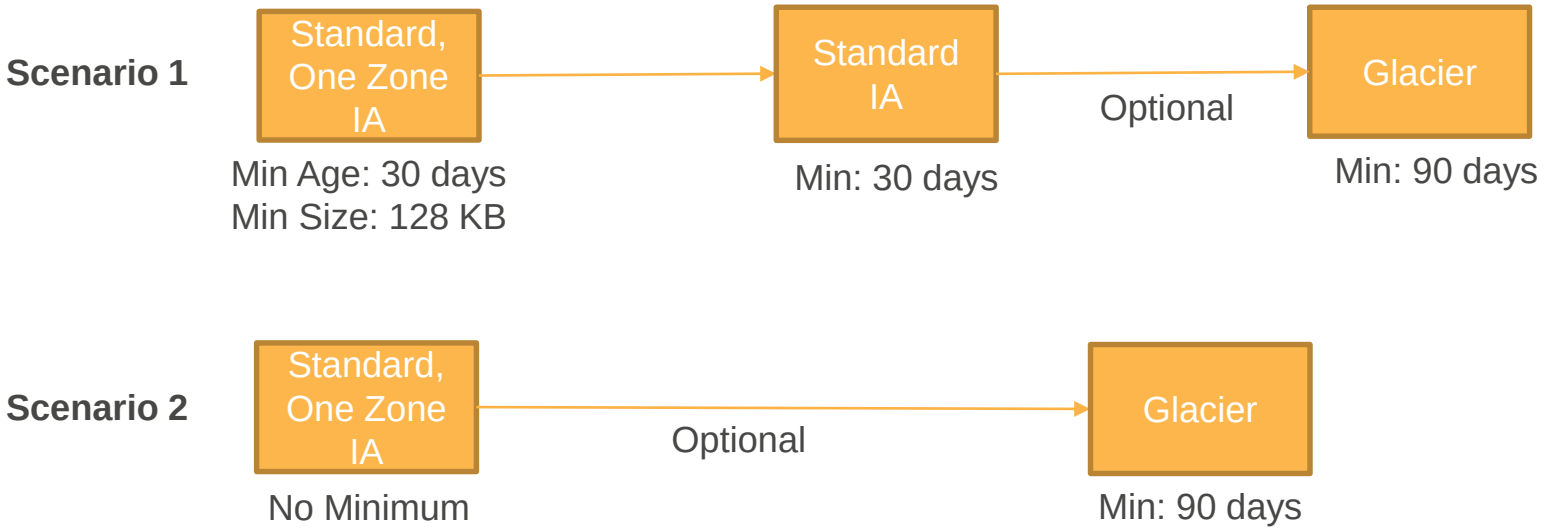
Use Case: Lifecycle Scenarios

Glacier Storage – For each archived object

1. S3 uses 8KB of storage for name and other metadata
2. Glacier uses 32KB for index and other metadata
3. Overhead can add up for small objects
4. Consider combining small objects into fewer large objects

Lifecycle Management Rules

1. Expiration Policy applied based on age at any step
2. Following Transitions are supported



One Zone IA is now replacing RRS for lifecycle transitions

Lifecycle with Versioning

- Objects in versioned bucket
  - One current version
  - Zero or more non current versions
- You can define separate rules for current and non-current object versions

Table: Non Versioned and Versioned Transitions

Demo 1

Lifecycle Rule to Manage Object Retention By Age

Demo 2

Lifecycle Rule to Manage Object Retention By Folder Structure and Age

Demo 3

Lifecycle Rule to Enforce a Tiered Storage Policy

- Optimize Cost of Storage

Requirement:

- Hot Data in S3 Standard (default)
- Transition Warm Data to S3 Standard-IA after 30 days
- Transition Cold Data to Glacier after 60 days
- Expire Objects after 1 year

Demo 4

Lifecycle Rule to Enforce Tiered Storage Policy for Previous Versions of Object

- Non-Current version default storage
- Transition non-current version to Glacier - 30 days after becoming non-current
- Expire non-current version - 180 days after becoming non-current

Cross Region Replication

Cross Region Replication (CRR)

- Replicate S3 Buckets to a different region
- Automatic, Asynchronous
- Metadata, Object ACLs are also replicated as well as any changes to metadata or ACL
- Versioning required on source and destination buckets

Cross Region Replication (CRR)

Use Case and Scenarios

Cross Region Replication (CRR)

- Source and Destination buckets in different regions
- One replication setup per source bucket
- Source Bucket Owner must have permission to objects
- Destination Bucket if owned by different account must grant permission to source account to replicate
- S3 service needs permission through role – Assume Role privileges and Role with permissions

CRR – What is replicated?

- New objects created after CRR is enabled
- Object updates, ACL and Metadata changes
- Deletes
    - Delete without Version ID, Delete marker in source is replicated to destination
    - Delete with Version ID. Not replicated to protect against malicious delete
- **2018 Update**: Objects encrypted with SSE-S3 (S3 managed keys) and **SSE-KMS**
- Cross account source objects for which source bucket owner has ACL access

CRR – What is not replicated?

- Existing objects before CRR was enabled
- Objects with Server Side Encryption – Customer Provided Keys (SSE-C) are not replicated
- Source bucket owner does not have access to Object (cross account owner created object)
- Actions performed by Lifecycle configurations are not replicated
- Objects in source bucket that are replicas (created by another CRR) is not replicated
- Bucket level sub resources are not replicated

CRR – Replication Status

Source Object Metadata

- x-amz-replication-status with value: PENDING, COMPLETED, FAILED

Destination Object Metadata

- x-amz-replication-status with value: REPLICA

Access Management

Access Management

- User Policies (covered in IAM Lecture)
- Resource Policies
    - Bucket Policies (covered in IAM Lecture)
    - Bucket Access Control List (ACL)
    - Object Access Control List (ACL)
- When to use ACLs?

Object ACL

- Control permissions at object level - Permissions vary by object
- If Object owner is different from bucket owner – Object ACL is the only way object owner can grant permissions
    - Bucket owner cannot read until given permission
    - Bucket owner can deny access to object
- No user level permissions. Only at account level
- Grantee: Another Account or predefined S3 groups. Account can be referred by email address or Canonical ID

Bucket ACL

Only recommended use for Bucket ACL

- Grant access to S3 `Log Delivery Group` to write S3 access logs to your bucket
- Bucket ACL is the only way in which `Log Delivery Group` can be granted access
- No user level permissions. Only at account level
- Grantee: Another Account or predefined S3 groups. Account can be referred by email address or Canonical ID

Bucket Policy

- User Policy or Bucket Policy – to manage access within same AWS account
- Bucket Policy
  - Can be used to grant cross-account access permissions for all S3 Actions (no need to use IAM role as a proxy)
  - Bucket ACL can also grant cross-account access but only for some S3 Actions

Encryption

Data Protection

- In-transit protection
  - HTTPS endpoints for AWS Services
  - Client side encryption
- Data at rest
  - S3 Server Side Encryption – S3 encrypts object when storing and decrypts when retrieving
  - Client Side Encryption – Encrypt data on your side and upload encrypted data to S3. Encryption process, keys, and tools are managed by client

Server Side Encryption

- S3 encrypts when writing object and decrypts when reading object
- For authorized users, no difference between encrypted and unencrypted object – Transparently handled by S3
- Three key management choices
    - S3 managed keys (SSE-S3)
    - Key Management Service managed keys (SSE-KMS)
    - Customer provided keys (SSE-C)
- AES256 Encryption Algorithm
- Object Data is encrypted. Metadata is not encrypted

	SSE – S3	SSE - KMS	SSE - C
Master Key and Access Control	S3 manages	Customer Manages using AWS Key Management System (KMS). Separate Access Control	Customer maintains the key
Data Key	Each Object is encrypted with Unique Data Key	Each Object is encrypted with Unique Data Key	Customer Provides the Data Key with every Object Request
Data Key Security	Data Key is encrypted with Master Key. Stored with Object	Data Key is encrypted with Master Key. Stored with Object	Data Key is not stored by S3. Salt derived from Data key is stored to validate future requests
Encryption Instruction Header	s3:x-amz-server-side-encryption:AES256,true	s3:x-amz-server-side-encryption:aws:kms	x-amz-server-side-encryption-customer-algorithm:AES256
Additional Header		Optional: s3:x-amz-server-side-encryption-aws-kms-key-id:<ARN for KMS Key>	x-amz-server-side-encryption-customer-key:<Base 64 encoded 256 bit key> x-amz-server-side-encryption-customer-key-MD5:<Hash for the key>
Key For Download	Not Required	Not Required	Same Key as above along with MD5 Hash

Static Website Hosting

Static Website

- Content must be publicly accessible
  - Bucket Policy
  - Individual object ACL
- For Private Website – Use CloudFront Content Distribution Network

Table: Difference between REST API Endpoint and Website

REST Access:

https://s3-us-west-2.amazonaws.com/bucket_name/access_test.txt

https://bucket_name.s3-us-west-2.amazonaws.com/access_test.txt

Pre-signed URL

- Allow others temporary access to specific access to download object or upload one:
  - Grant temporary access to a third party to download a document
  - Allow a third part to upload a file to your bucket
- Using S3 Client SDK and using a appropriate credentials, you can generate pre-signed url
- URL Query String contains enough information to authenticate the request.

Notifications

Demo

Monitoring - CloudWatch

- BucketSizeBytes – Reports total bytes stored in a bucket. Includes Standard, Standard-IA, Reduced Redundancy. Does not include Glacier storage
- NumberOfObject – Total number of objects stored for all storage classes except Glacier Storage
- Filters, Dimensions:
  - BucketName – Filter data for specified bucket
  - StorageType – Filter data for specified storage class. Does not include Glacier Storage.
- Daily metrics for S3 are automatically available in CloudWatch at no additional cost

Server Access Logs

- Logs every single access request to a source bucket
- Each access is a row and with space delimited fields.
- Needs to be enabled explicitly at bucket level
- Periodic collection of log data and stores captured log in the bucket you specify (target bucket)
- Log Delivery Group needs to have write access to your target bucket
- CloudTrail can track Bucket Level API (console, SDK, REST,SOAP,…) calls made to your S3 resources – but not individual object access requests

Request Routing - Redirection

- S3 uses DNS to route the requests to facilities that can process request
- Temporary routing errors can occur
- S3 maintains high availability by periodically updating IP Addresses associated with end-points in DNS as needed
- Clients that do not respect DNS Time To Live settings may encounter redirection response (caching DNS for a very long time).
- If request arrives at wrong location, S3 responds with a temporary redirect to resend to a new end point
- If request is incorrectly formed, S3 responds with a permanent redirects with details on how to request correctly
- Every S3 program must be designed to handle redirect responses

https://s3-us-west-2.amazonaws.com/bucket_name/access_test.txt

Request Routing Demo

S3 Request Routing Flow

Permanent Redirect

https://s3.amazonaws.com/bucket_name/access_test.txt

Access

https://s3-us-west-2.amazonaws.com/bucket_name/access_test.txt

https://bucket_name.s3-us-west-2.amazonaws.com/access_test.txt

S3 Transfer Acceleration

- Customers from all over the world uploading to your central bucket - Inconsistent performance due to errors, bandwidth issues, large file sizes
- Take advantage of Amazon CloudFront's Global network of edge locations
- Clients upload to edge location and data is routed to central bucket over an optimized network
- Additional charges apply - $0.04 to $0.08 per GB

bucketname.s3-accelerate.amazonaws.com (IPv4)

bucketname.s3-accelerate.dualstack.amazonaws.com (IPv6 and IPv4)

Transfer Acceleration

AWS provides a transfer acceleration comparison tool to compare performance of accelerated and non-accelerated upload speeds

- Compare transfer performance to different regions with and without transfer acceleration
- Tests with multipart uploads
- region=region&origBucketName=yourBucketName
- Not HIPAA compliant

BitTorrent

- S3 supports BitTorrent protocol for distributing publicly accessible files
- Lower cost of file distribution – S3 cost increases with the popularity of the object
  - More requests and more data transfer
- BitTorrent uses peer-to-peer to use downloading clients as distributors for others
  - For subsequent requests to BitTorrent for the same S3 object, it will be served by BitTorrent
- Supported for files less than 5GB

Publishing Content using BitTorrent

- Every publicly object (ACL object permission) is automatically available for download with BitTorrent
- BitTorrent requires a .torrent file that describes data to be downloaded and where you to find the data
- S3 automatically generates .torrent file when “?torrent” parameter is added to the GET request
- BitTorrent client can download object using .torrent file or publishing a link to ?torrent URL for your object
- .torrent file is generated on demand by S3 so it may take sometime when dealing with large objects

Cross Origin Resource Sharing (CORS)

- CORS refers to web applications from one domain interacting with resources from another domain
- S3 can be used to host scripts, stylesheets, images, videos and other resources that web applications can use
- In S3, you can configure specific domains that can make CORS request to interact with S3 content
- If web browser request's *origin* header match *AllowedOrigin* element in the Bucket, request is processed and returned

Requester Pays

- Bucket owner pays for storage and data transfer costs
- Option to ask for the requester to pay for data transfer
  - Bucket Owner always pays for the storage
- When enabled, requester must include `x-amz-request-payer` header to confirm they are paying!
- Anonymous access not allowed

Demo