Amazon S3 - Advanced

Amazon S3 – Moving between Storage Classes

- You can transition objects between storage classes.
- For infrequently accessed objects, move them to **Standard IA**.
- For archive objects that you don't need fast access to, move them to **Glacier** or **Glacier Deep Archive**.
- Moving objects can be automated using **Lifecycle Rules**.

Transition Paths (Explained)

Objects stored in **Standard** storage class can be transitioned to:

- **Standard IA**
- **Intelligent Tiering**
- **One-Zone IA**
- **Glacier Instant Retrieval**
- **Glacier Flexible Retrieval**
- **Glacier Deep Archive**

From **Standard IA**, objects can be moved to:

- **Intelligent Tiering**
- **One-Zone IA**
- **Glacier Instant Retrieval**
- **Glacier Flexible Retrieval**
- **Glacier Deep Archive**

Intelligent Tiering allows transitions to:

- **One-Zone IA**
- **Glacier Instant Retrieval**
- **Glacier Flexible Retrieval**
- **Glacier Deep Archive**

One-Zone IA can be transitioned to:

- **Glacier Instant Retrieval**
- **Glacier Flexible Retrieval**
- **Glacier Deep Archive**

Glacier Instant Retrieval can be transitioned to:

- **Glacier Flexible Retrieval**
- **Glacier Deep Archive**

Glacier Flexible Retrieval can be transitioned to:

- **Glacier Deep Archive**

Amazon S3 – Lifecycle Rules

Transition Actions

Configure objects to transition to another storage class:

- Move objects to **Standard IA** class 60 days after creation
- Move objects to **Glacier** for archiving after 6 months

Expiration Actions

Configure objects to expire (delete) after some time:

- Access log files can be set to delete after 365 days
- **Can be used to delete old versions of files** (if versioning is enabled)
- Can be used to delete **incomplete Multi-Part uploads**

Rule Scope

- Rules can be created for a certain **prefix**
  Example: `s3://mybucket/mp3/*`
- Rules can be created for certain object **tags**
  Example: `Department: Finance`

Amazon S3 – Lifecycle Rules (Scenario 1)

Scenario

Your application on EC2 creates image thumbnails after profile photos are uploaded to Amazon S3.

- Thumbnails can be easily recreated and only need to be stored for 60 days.
- Source images should be immediately retrievable for 60 days.
- After 60 days, users can tolerate a retrieval delay of up to 6 hours.

Design

- **S3 Source Images**
  - Storage Class: **Standard**
  - Lifecycle Rule: Transition to **Glacier** after 60 days
- **S3 Thumbnails**
  - Storage Class: **One-Zone IA**
  - Lifecycle Rule: **Expire** (delete) after 60 days

Amazon S3 – Lifecycle Rules (Scenario 2)

Scenario

A rule in your company requires that deleted S3 objects:

- Must be recoverable **immediately** for the first **30 days**
- After 30 days, but up to **365 days**, should remain **recoverable within 48 hours**

Design

- **Enable S3 Versioning**
  This ensures deleted objects are not actually removed, but are hidden by a “delete marker” and can be recovered.

- **Lifecycle Rule for Noncurrent Versions**:
    - After an object is deleted (i.e., becomes noncurrent), transition it to **Standard IA** (after 30 days)
    - Then transition it to **Glacier Deep Archive** (after 365 days total)

This configuration keeps objects quickly accessible initially, while reducing storage costs over time.

Amazon S3 Analytics – Storage Class Analysis

Overview

- Helps you decide **when to transition objects** to the right storage class.
- Provides **recommendations** for:
    - **Standard**
    - **Standard-IA**
- Does **not** support:
    - **One-Zone IA**
    - **Glacier**

Reporting

- The report is **updated daily**.
- It takes **24 to 48 hours** to start seeing data analysis.

Output Format

- Report is exported as a **.csv file**.
- Includes fields like:
    - **Date**
    - **StorageClass**
    - **ObjectAge** (e.g., 000–014, 030–044)

Use Case

- Great first step for creating or improving **Lifecycle Rules**.

S3 – Requester Pays

Default Behavior

- By default, **bucket owners** pay for:
    - **S3 storage**
    - **Data transfer costs** (including downloads)

Requester Pays Buckets

- With **Requester Pays**, the **requester** pays for:
    - The cost of the **request**
    - The **data download** from the bucket
- The **bucket owner** continues to pay for **storage costs**.

Use Cases

- Useful when sharing **large datasets** with other AWS accounts.
- Prevents the bucket owner from bearing the cost of data retrieval by others.

Requirements

- The **requester must be authenticated** in AWS.
  - Anonymous users are **not allowed**.

Visual Summary

Bucket Type	Storage Cost	Networking (Download) Cost
Standard Bucket	Owner	Owner
Requester Pays Bucket	Owner	Requester

S3 Event Notifications

Supported Events

- `s3:ObjectCreated`
- `s3:ObjectRemoved`
- `s3:ObjectRestore`
- `s3:Replication`
- ...

Features

- Supports **object name filtering** (e.g., `*.jpg`)
- **Use case**: Automatically generate thumbnails when images are uploaded to S3
- You can define **multiple S3 event notifications** per bucket

Destinations

S3 events can trigger:

- **SNS topics**
- **SQS queues**
- **Lambda functions**

Behavior

- Events are typically delivered in **seconds**
- Delivery may take **a minute or more** in some cases

S3 Event Notifications – IAM Permissions

To allow S3 to send events to SNS, SQS, or Lambda, **resource-based policies** must be set on the target services.

SNS Resource Policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "SNS:Publish",
    "Principal": {
      "Service": "s3.amazonaws.com"
```

```
    },
    "Resource": "arn:aws:sns:us-east-1:123456789012:MyTopic",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:s3:::MyBucket"
      }
    }
  }
}
```

SQS Resource Policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "SQS:SendMessage",
    "Principal": {
      "Service": "s3.amazonaws.com"
    },
    "Resource": "arn:aws:sqs:us-east-1:123456789012:MyQueue",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:s3:::MyBucket"
      }
    }
  }
}
```

Lambda Resource Policy

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "lambda:InvokeFunction",
    "Principal": {
      "Service": "s3.amazonaws.com"
    },
    "Resource": "arn:aws:lambda:us-east-1:123456789012:function:MyFunction",
    "Condition": {
      "ArnLike": {
        "AWS:SourceArn": "arn:aws:s3:::MyBucket"
      }
    }
  }
}
```

Summary

- These policies allow Amazon S3 to **publish**, **send messages**, or **invoke functions** on the target resources.
- The `Condition` with `ArnLike` ensures that the permission is **only granted for a specific S3 bucket**.

S3 Event Notifications with Amazon EventBridge

Overview

- Amazon S3 can send **all events** to **Amazon EventBridge**
- EventBridge can route events to **over 18 AWS services**

Features

- **Advanced Filtering** using JSON-based rules
  (e.g., based on metadata, object size, object name…)
- **Multiple Destinations**
  Examples:
  - AWS Step Functions
  - Amazon Kinesis Streams / Firehose
  - Lambda, SNS, SQS, etc.
- **EventBridge Capabilities**
  - **Archive** events
  - **Replay** events
  - **Reliable delivery**

S3 – Baseline Performance

- Amazon S3 **automatically scales** to handle high request rates.
- Typical **latency** is between **100–200 ms**.
- Each prefix in a bucket supports:
  - **3,500 PUT/COPY/POST/DELETE requests per second**
  - **5,500 GET/HEAD requests per second**

Prefix Strategy

- There are **no limits** on the number of prefixes in a bucket.
- Distributing requests across multiple prefixes increases total throughput.

Example (Object Path => Prefix)

- `bucket/folder1/sub1/file` => `/folder1/sub1/`
- `bucket/folder1/sub2/file` => `/folder1/sub2/`
- `bucket/1/file` => `/1/`
- `bucket/2/file` => `/2/`

By **spreading reads** across all four prefixes, you can achieve:

- **22,000 GET/HEAD requests per second**

S3 Performance

Multi-Part Upload

- **Recommended for files > 100MB**

- **Required for files > 5GB**
- Allows upload of parts in **parallel**, which can **speed up transfer performance**

Process

1. File is divided into parts
2. Each part is uploaded in parallel to Amazon S3
3. Parts are reassembled into a single object by S3

S3 Transfer Acceleration

- Speeds up uploads by routing files to an **AWS edge location** near the client
- The edge location **forwards data over AWS's internal network** to the destination S3 bucket
- Useful when uploading from **geographically distant locations**
- Fully **compatible with multi-part upload**

Example

- A file in the USA is uploaded to a bucket in Australia:
  - Data travels quickly from the client to a **local edge location**
  - Then continues rapidly over AWS infrastructure to the **S3 bucket in Australia**

S3 Performance – S3 Byte-Range Fetches

Overview

- Allows parallelization of **GET** requests by specifying **byte ranges**
- Improves **resilience** in case of partial failures
- Enhances **download speed** by retrieving file segments in parallel

Use Cases

Speed up downloads

- Split a large file into byte ranges
- Send **parallel GET requests** for each part
- Reassemble the parts locally

Retrieve only part of a file

- Useful to get the **header** or **first few bytes**
- Example: fetch metadata or validate format before downloading entire file

Example

Parallel download

File in S3 +---+ | Part 1 | Part 2 | ... | Part N | +---+

Requests in parallel:

- GET byte-range 0-99999
- GET byte-range 100000-199999 ...

Partial read

File in S3 +---+ | Header | Body | +---+

GET byte-range 0-1023 → retrieve only the header

S3 Batch Operations

Overview

Perform bulk operations on existing S3 objects using a **single request**.

Examples of Supported Actions

- Modify object metadata & properties
- Copy objects between S3 buckets
- **Encrypt un-encrypted objects**
- Modify ACLs and tags
- Restore objects from **S3 Glacier**
- Invoke a **Lambda function** to perform custom logic on each object

Job Structure

- A job includes:
  - A **list of objects**
  - The **action** to perform
  - **Optional parameters**

Capabilities

- Handles:
  - **Retries**
  - **Progress tracking**
  - **Completion notifications**
  - **Job reports**

Integration Flow

1. Use **S3 Inventory** to generate an object list.
2. Use **Athena** to filter/query objects of interest.
3. Submit the filtered list to **S3 Batch Operations** with the desired operation and parameters.
4. Processed objects are updated as per the job definition.

Notes

- Useful for large-scale, automated maintenance or migration tasks across S3.

S3 – Storage Lens

Overview

- Helps you **understand, analyze, and optimize** Amazon S3 storage across your entire **AWS Organization**.
- Identifies **anomalies**, suggests **cost-saving opportunities**, and enforces **data protection best practices**.

Features

- Provides **30-day usage and activity metrics**
- Aggregates data at multiple levels:
  - AWS **Organization**
  - **Accounts**
  - **Regions**
  - **Buckets**

    - **Prefixes**

Dashboard

- Includes a **default dashboard**
- You can also create **custom dashboards** tailored to your needs

Data Export

- Can export metrics **daily** to an S3 bucket
- Export formats: **CSV** or **Parquet**

Use Cases

- Gain **summary insights**
- Enforce **data protection**
- Improve **cost efficiency**

Storage Lens – Default Dashboard

Key Features

- Visualizes **summarized insights and trends** for both **free** and **advanced metrics**
- Displays **multi-region** and **multi-account** data
- Preconfigured by **Amazon S3** (no setup required)

Characteristics

- **Cannot be deleted**, but it **can be disabled**
- Includes filters for:
    - Accounts
    - Regions
    - Buckets
    - Prefixes
    - Storage classes

Dashboard Views

- Metrics include:
    - **Total storage**
    - **Object count**
    - **Average object size**
    - **Number of buckets and accounts**
    - **All requests**
- Trend views:
    - Daily
    - Weekly
    - Monthly

Learn more: AWS Storage Lens Blog

Storage Lens – Metrics

Summary Metrics

- Provide **general insights** about your S3 storage

- Key metrics:
  - `StorageBytes`
  - `ObjectCount`
- **Use cases**:
  - Identify **fast-growing buckets or prefixes**
  - Detect **unused storage paths**

Cost-Optimization Metrics

- Help **manage and reduce** storage costs
- Key metrics:
  - `NonCurrentVersionStorageBytes`
  - `IncompleteMultipartUploadStorageBytes`
- **Use cases**:
  - Identify **incomplete multipart uploads** older than 7 days
  - Find objects that could be **moved to lower-cost storage classes**

Storage Lens – Metrics (Advanced Categories)

Data-Protection Metrics

- Provide visibility into **data protection features** configured in buckets
- Key metrics:
  - `VersioningEnabledBucketCount`
  - `MFADeleteEnabledBucketCount`
  - `SSEKMSEnabledBucketCount`
  - `CrossRegionReplicationRuleCount`
- **Use cases**:
  - Identify buckets that **do not follow best practices** for data protection

Access-Management Metrics

- Provide visibility into **Object Ownership settings**
- Key metric:
  - `ObjectOwnershipBucketOwnerEnforcedBucketCount`
- **Use cases**:
  - Determine which **Object Ownership setting** each bucket uses

Event Metrics

- Provide visibility into **S3 Event Notifications**
- Key metric:
  - `EventNotificationEnabledBucketCount`
- **Use cases**:
  - Identify buckets with **S3 Event Notifications configured**

Storage Lens – Metrics (continued)

Performance Metrics

- Provide insights for **S3 Transfer Acceleration**

- Key metric:
  - `TransferAccelerationEnabledBucketCount`
- **Use case**: Identify which buckets have Transfer Acceleration enabled

Activity Metrics

- Provide insights on **how S3 is being used**
- Key metrics:
  - `AllRequests`
  - `GetRequests`
  - `PutRequests`
  - `ListRequests`
  - `BytesDownloaded`
- **Use case**: Analyze traffic patterns and storage usage

Detailed Status Code Metrics

- Provide detailed insights on **HTTP status codes**
- Key metrics:
  - `200OKStatusCount`
  - `403ForbiddenErrorCount`
  - `404NotFoundErrorCount`
- **Use case**: Monitor access errors and troubleshoot client requests

Storage Lens – Free vs. Paid

Free Metrics

- Automatically available for all customers
- Includes around **28 usage metrics**
- Data is **available for 14 days** for queries

Advanced Metrics and Recommendations

- Additional **paid** metrics and features
- **Advanced Metrics**:
  - Activity
  - Advanced Cost Optimization
  - Advanced Data Protection
  - Status Code
- **CloudWatch Publishing**:
  - Access metrics in CloudWatch **without additional charges**
- **Prefix Aggregation**:
  - Collect metrics at the **prefix level**
- Data is **available for queries for 15 months**