# S3 bucket

S3 bucket

Amazon s3 allows people to store objects (files) in buckets directories

Bucket must be have a globally unique name

bucket are defined at the region level

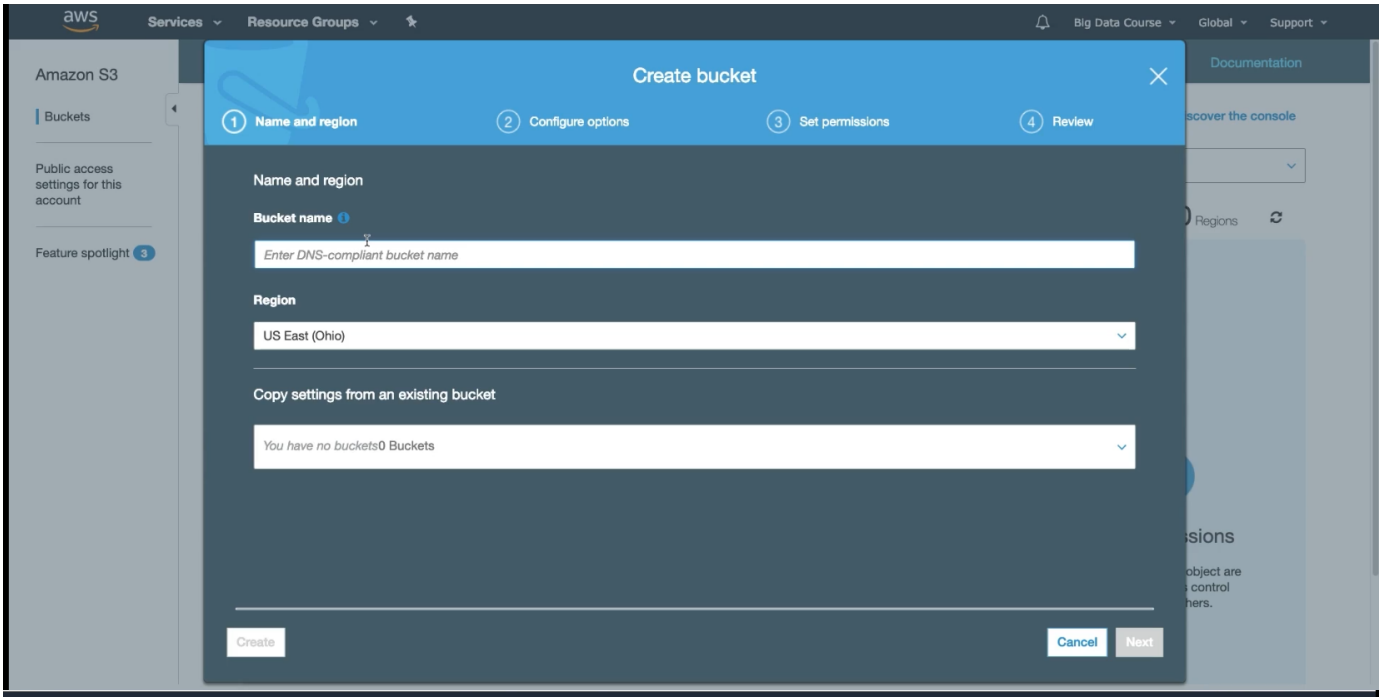
naming convention (No uppercase , No underscore , 3-63 characters long , not an ip , must start with lowercase are number )

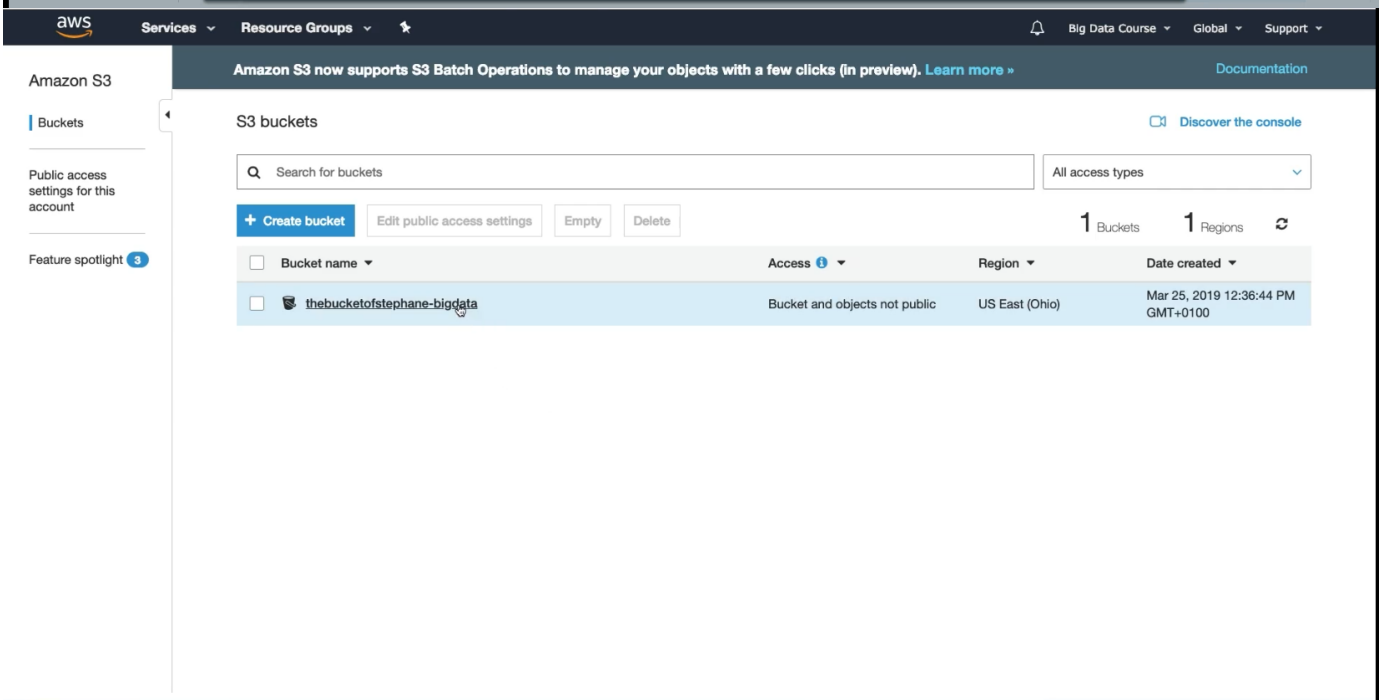
# AWS S3 Overview - Objects

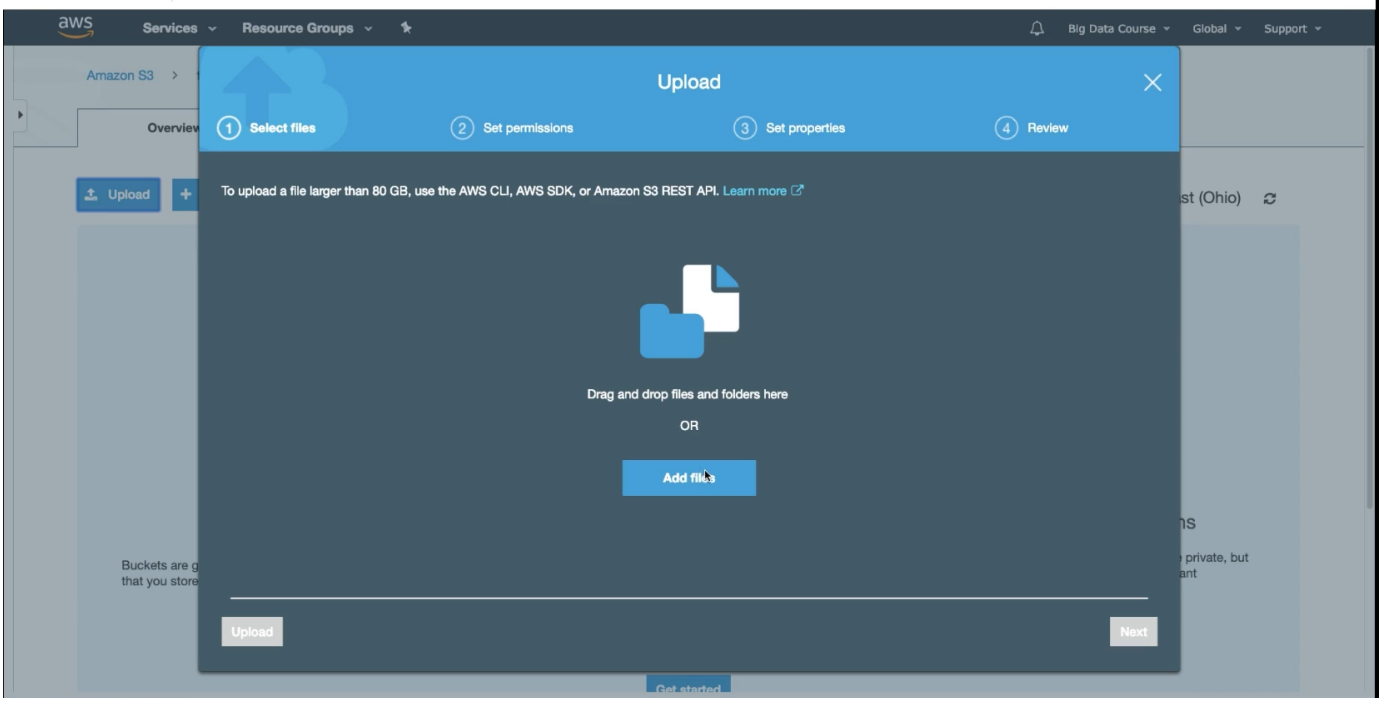
- Objects (files) have a Key. The key is the FULL path:
  - <my_bucket>/my_file.txt
  - <my_bucket>/my_folder1/another_folder/my_file.txt
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")
- Object Values are the content of the body:
  - Max Size is 5TB
  - If uploading more than 5GB, must use "multi-part upload"
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle

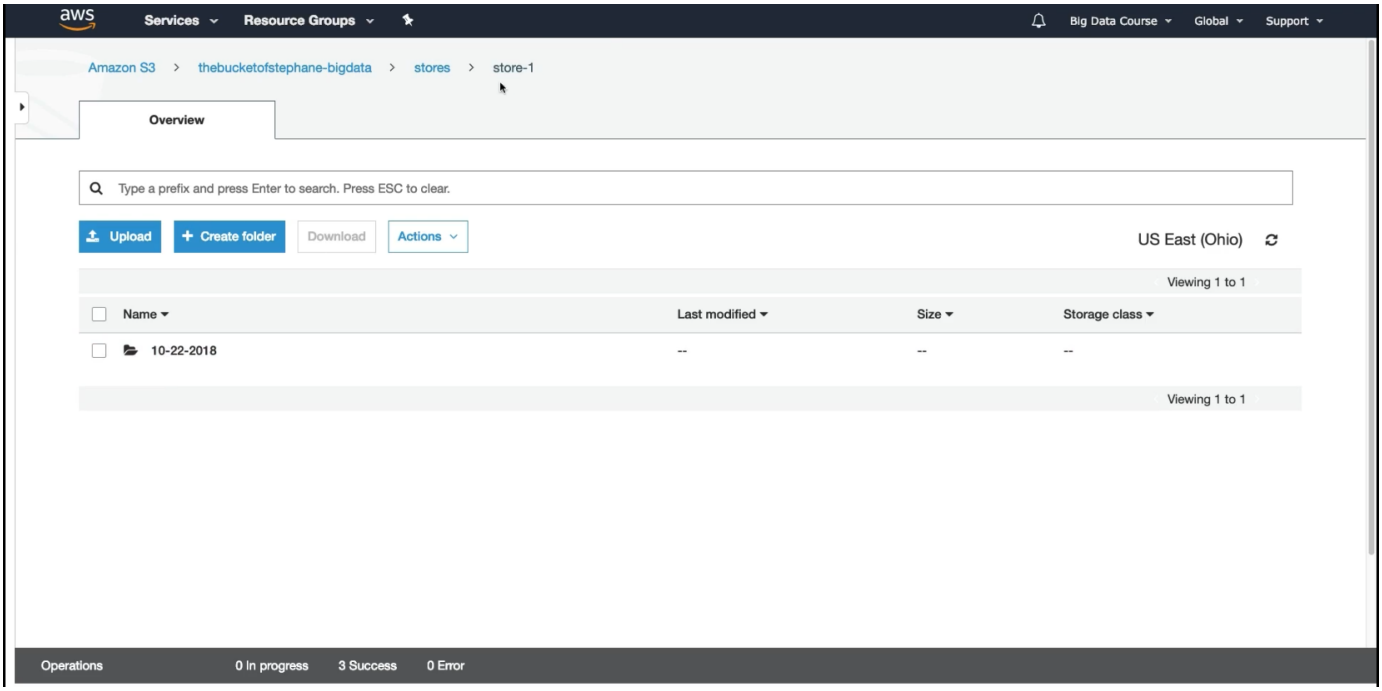
# Amazon S3 - Consistency Model

- Strong consistency as of December 2020:
- After a:
  - successful write of a new object (new PUT)
  - or an overwrite or delete of an existing object (overwrite PUT or DELETE)
- …any:
  - subsequent read request immediately receives the latest version of the object (read after write consistency)
  - subsequent list request immediately reflects changes (list consistency)
- Available at no additional cost, without any performance impact


Services
Resource Groups
Big Data Course
Global
Support
Amazon S3
Buckets
Public access settings for this account
Feature spotlight 3
Documentation
Create bucket
1 Name and region
2 Configure options
3 Set permissions
4 Review
Name and region
Bucket name
Enter DNS-compliant bucket name
Region
US East (Ohio)
Copy settings from an existing bucket
You have no buckets 0 Buckets
Create
Cancel
Next

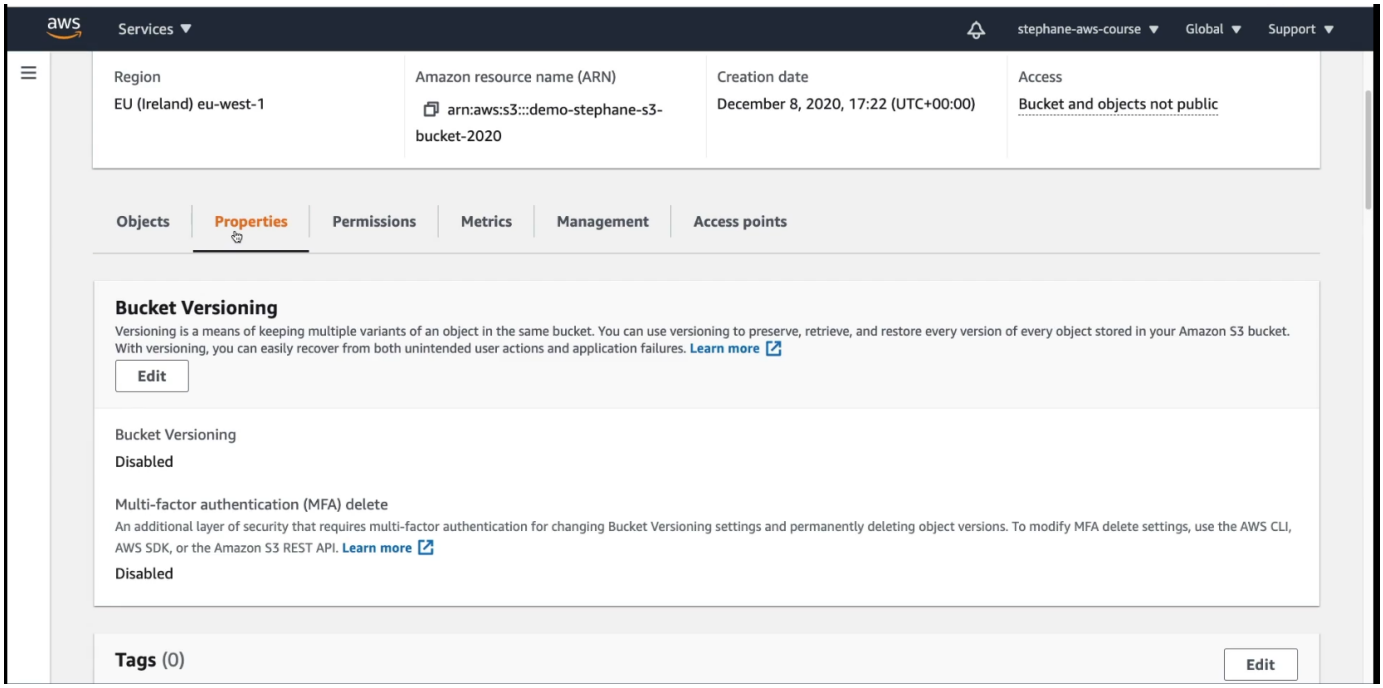

Services
Resource Groups
Big Data Course
Global
Support
Amazon S3
Buckets
Public access settings for this account
Feature spotlight 3
Amazon S3 now supports S3 Batch Operations to manage your objects with a few clicks (in preview). Learn more »
Documentation
S3 buckets
Discover the console
Search for buckets
All access types
Create bucket
Edit public access settings
Empty
Delete
1 Buckets
1 Regions
Bucket name
Access
Region
Date created
thebucketofstephane-bigdata
Bucket and objects not public
US East (Ohio)
Mar 25, 2019 12:36:44 PM GMT+0100

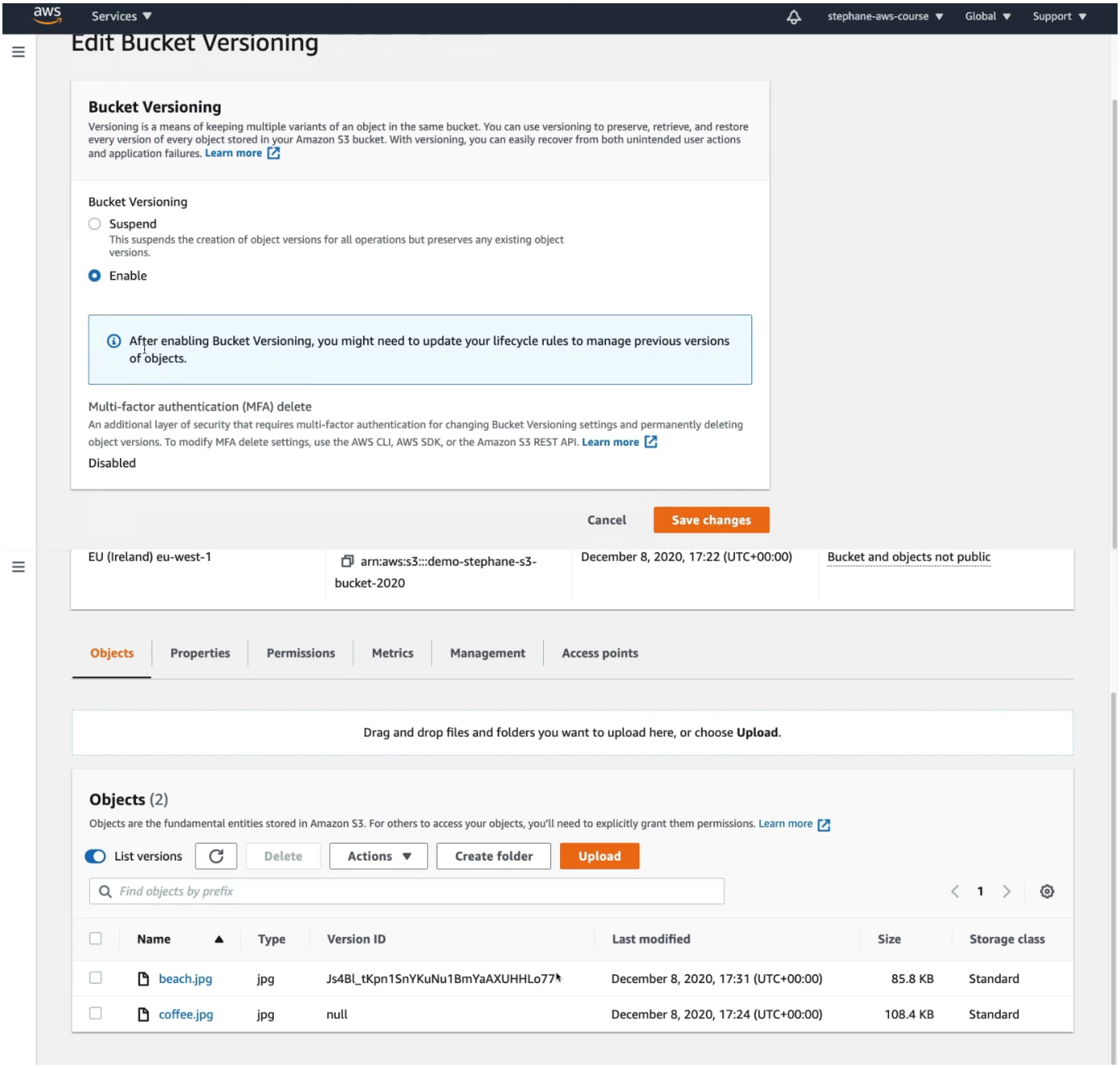

Services
Resource Groups
Big Data Course
Global
Support
Amazon S3
Upload
Upload
1 Select files
2 Set permissions
3 Set properties
4 Review
To upload a file larger than 80 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more
Drag and drop files and folders here
OR
Add files
Upload
Next


aws
Services
Resource Groups
Big Data Course
Global
Support
Amazon S3 > thebucketofstephane-bigdata > stores > store-1
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
+ Create folder
Download
Actions
US East (Ohio)
Viewing 1 to 1
Name
Last modified
Size
Storage class
10-22-2018
--
--
--
Viewing 1 to 1
Operations
0 In progress
3 Success
0 Error

# Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at the **bucket level**
- Same key overwrite will increment the "version": 1, 2, 3….
- It is best practice to version your buckets
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Notes:
  - Any file that is not versioned prior to enabling versioning will have version "null"
  - Suspending versioning does not delete the previous versions

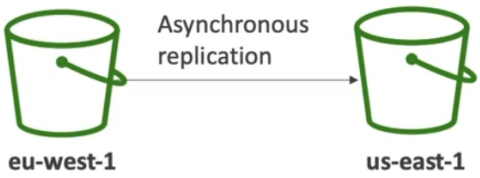





in bucket versioning you can delete any file it will create delete marker you easily restore the file if you delete the market it will delete permanently

# S3 Replication (CRR & SRR)

- **Must enable versioning** in source and destination
- Cross Region Replication (CRR)
- Same Region Replication (SRR)
- Buckets can be in different accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3

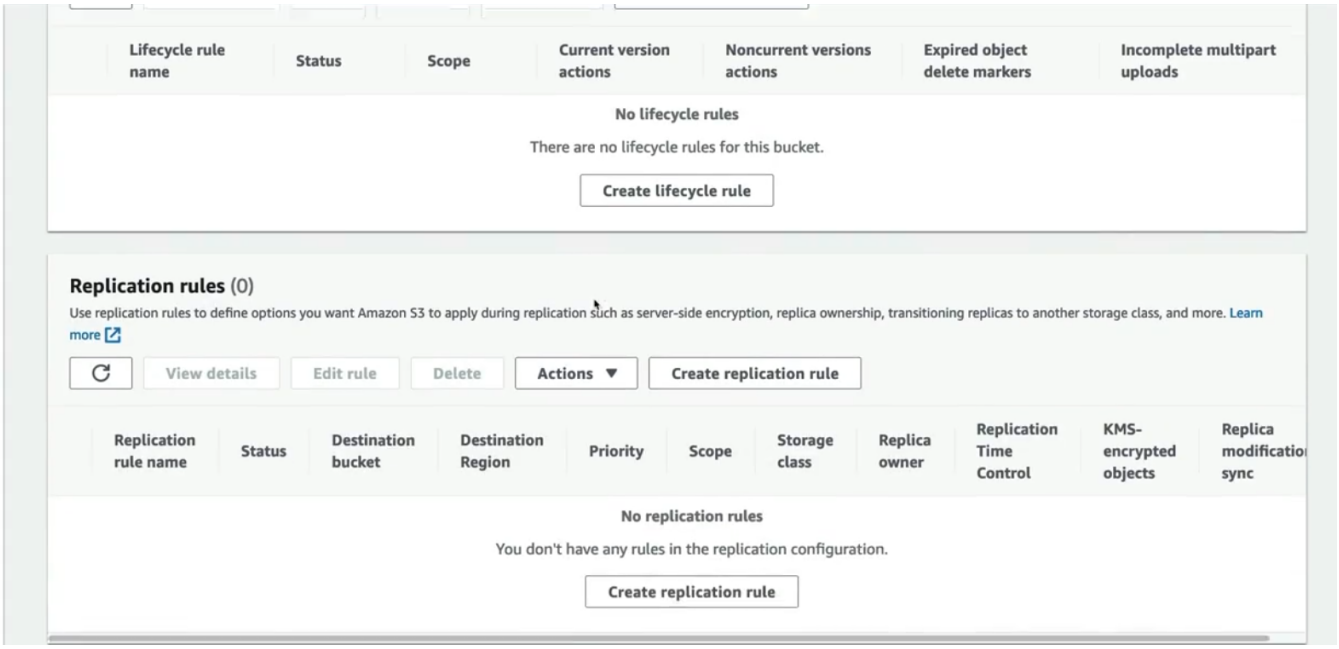


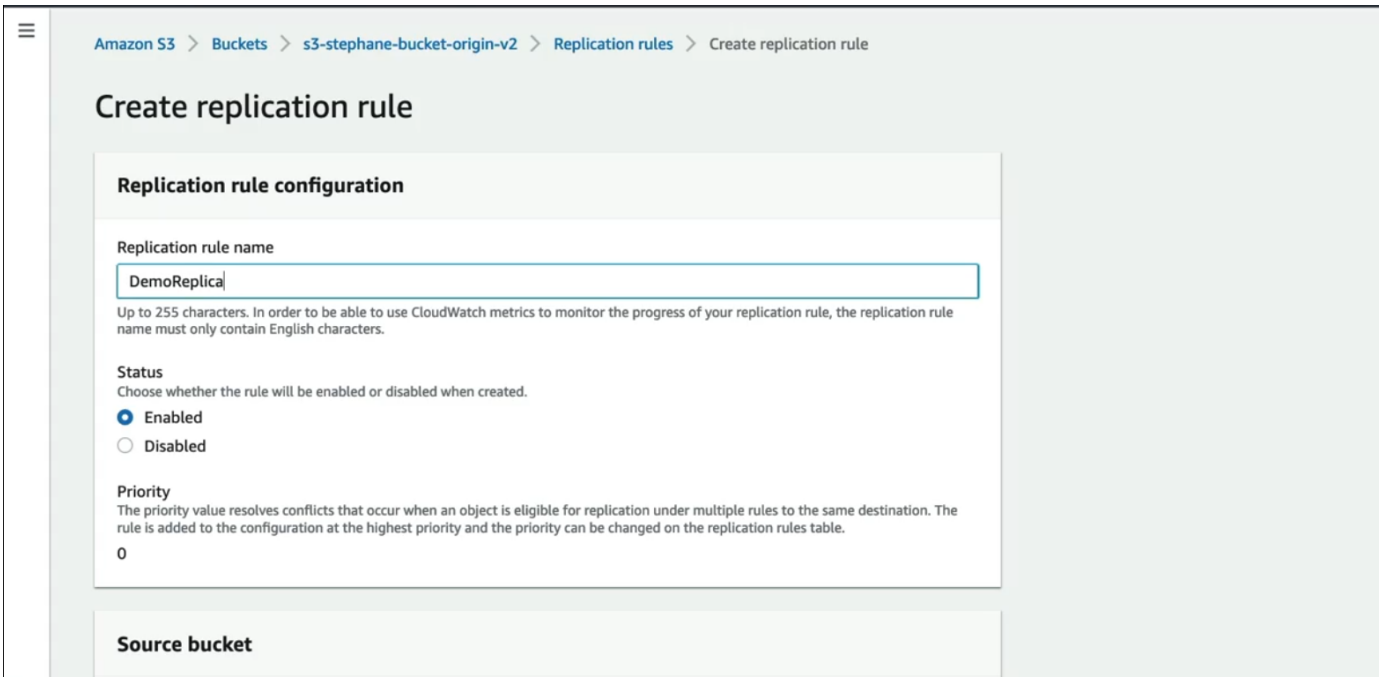

- CRR - Use cases: compliance, lower latency access, replication across accounts
- SRR – Use cases: log aggregation, live replication between production and test accounts

# S3 Replication – Notes

- After activating, only new objects are replicated
- Optionally, you can replicate existing objects using **S3 Batch Replication**
  - Replicates existing objects and objects that failed replication
- For DELETE operations:
  - Can replicate delete markers from source to target (optional setting)
  - Deletions with a version ID are not replicated (to avoid malicious deletes)
- There is no "chaining" of replication
  - If bucket 1 has replication into bucket 2, which has replication into bucket 3
  - Then objects created in bucket 1 are not replicated to bucket 3

under management


Lifecycle rule name
Status
Scope
Current version actions
Noncurrent versions actions
Expired object delete markers
Incomplete multipart uploads
No lifecycle rules
There are no lifecycle rules for this bucket.
Create lifecycle rule
Replication rules (0)
Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. Learn more
View details
Edit rule
Delete
Actions
Create replication rule
Replication rule name
Status
Destination bucket
Destination Region
Priority
Scope
Storage class
Replica owner
Replication Time Control
KMS-encrypted objects
Replica modificatio sync
No replication rules
You don't have any rules in the replication configuration.
Create replication rule


Amazon S3 > Buckets > s3-stephane-bucket-origin-v2 > Replication rules > Create replication rule
Create replication rule
Replication rule configuration
Replication rule name
DemoReplica
Up to 255 characters. In order to be able to use CloudWatch metrics to monitor the progress of your replication rule, the replication rule name must only contain English characters.
Status
Choose whether the rule will be enabled or disabled when created.
Enabled
Disabled
Priority
The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.
0
Source bucket

Choose a rule scope

- ○ Limit the scope of this rule using one or more filters
- ◉ Apply to all objects in the bucket

## Destination

Destination

You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. Learn more or see Amazon S3 pricing

- ◉ Choose a bucket in this account
- ○ Specify a bucket in another account

Bucket name

Choose the bucket that will receive replicated objects.

Enter bucket name | Browse S3

Destination Region

-

## IAM role

Choose the bucket that will receive replicated objects.

s3-stephane-bucket-replica-v2 | Browse S3

Destination Region

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

## IAM role

- ◉ Choose from existing IAM roles
- ○ Enter IAM role ARN

IAM role

Choose IAM role | View

## Encryption

☐ Replicate objects encrypted with AWS KMS

You can replicate objects that are encrypted with AWS Key Management Service (AWS KMS) keys.

## Destination storage class

# S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering

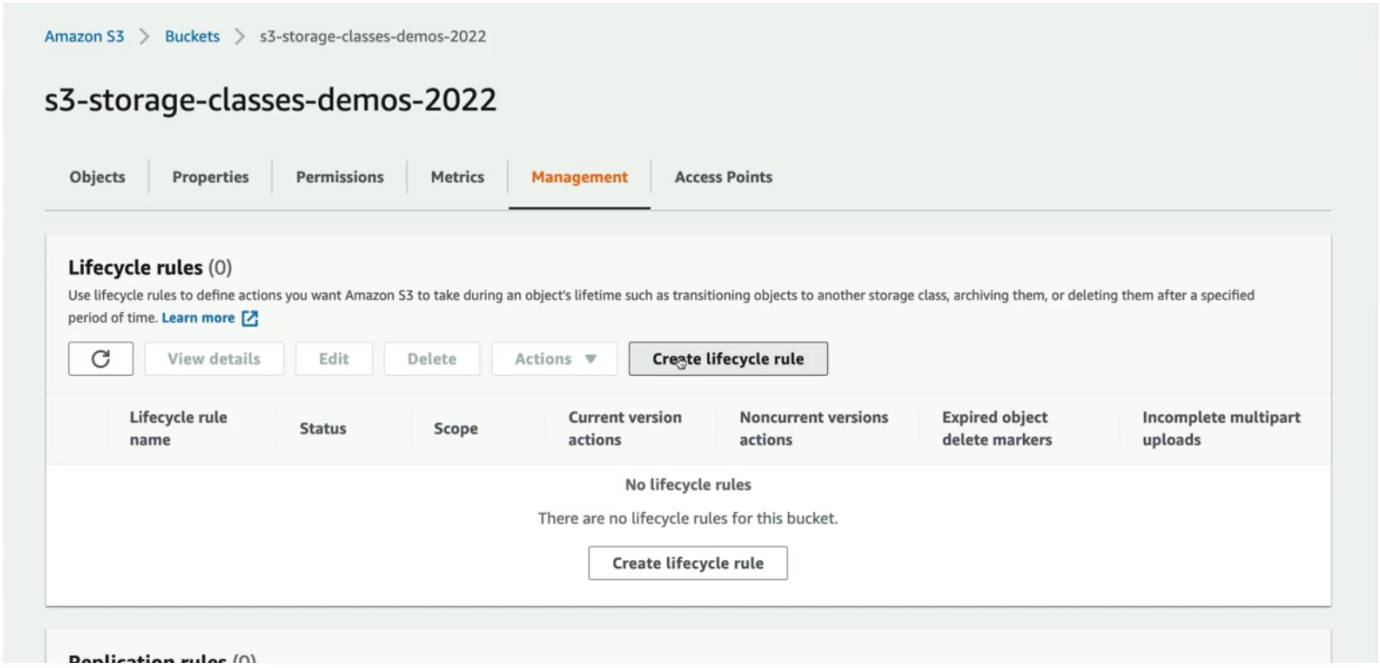
- Can move between classes manually or using S3 Lifecycle configurations

## S3 Storage Classes Comparison

| | Standard | Intelligent-Tiering | Standard-IA | One Zone-IA | Glacier Instant Retrieval | Glacier Flexible Retrieval | Glacier Deep Archive |
|---|---|---|---|---|---|---|---|
| Durability | 99.999999999% == (11 9's) | | | | | | |
| Availability | 99.99% | 99.9% | 99.9% | 99.5% | 99.9% | 99.99% | 99.99% |
| Availability SLA | 99.9% | 99% | 99% | 99% | 99% | 99.9% | 99.9% |
| Availability Zones | >= 3 | >= 3 | >= 3 | 1 | >= 3 | >= 3 | >= 3 |
| Min. Storage Duration Charge | None | None | 30 Days | 30 Days | 90 Days | 90 Days | 180 Days |
| Min. Billable Object Size | None | None | 128 KB | 128 KB | 128 KB | 40 KB | 40 KB |
| Retrieval Fee | None | None | Per GB retrieved | Per GB retrieved | Per GB retrieved | Per GB retrieved | Per GB retrieved |

# S3 Storage Classes – Price Comparison Example: us-east-1

| | Standard | Intelligent-Tiering | Standard-IA | One Zone-IA | Glacier Instant Retrieval | Glacier Flexible Retrieval | Glacier Deep Archive |
|---|---|---|---|---|---|---|---|
| Storage Cost (per GB per month) | $0.023 | $0.0025 - $0.023 | %0.0125 | $0.01 | $0.004 | $0.0036 | $0.00099 |
| Retrieval Cost (per 1000 request) | **GET:** $0.0004<br>**POST:** $0.005 | **GET:** $0.0004<br>**POST:** $0.005 | **GET:** $0.001<br>**POST:** $0.01 | **GET:** $0.001<br>**POST:** $0.01 | **GET:** $0.01<br>**POST:** $0.02 | **GET:** $0.0004<br>**POST:** $0.03<br><br>**Expedited:** $10<br>**Standard:** $0.05<br>**Bulk:** free | **GET:** $0.0004<br>**POST:** $0.05<br><br>**Standard:** $0.10<br>**Bulk:** $0.025 |
| Retrieval Time | Instantaneous | | | | | **Expedited** (1 – 5 mins)<br>**Standard** (3 – 5 hours)<br>Bulk (5 – 12 hours) | **Standard** (12 hours)<br>**Bulk** (48 hours) |
| Monitoring Cost (pet 1000 objects) | | $0.0025 | | | | | |




create a name

click apply all objects in the bucket

and on the move current version of object to storage classes

Choose the actions you want this rule to perform. Per-request fees apply. Learn more or see Amazon S3 pricing

- [x] Move current versions of objects between storage classes
- [ ] Move noncurrent versions of objects between storage classes
- [ ] Expire current versions of objects
- [ ] Permanently delete noncurrent versions of objects
- [ ] Delete expired object delete markers or incomplete multipart uploads

These actions are not supported when filtering by object tags or object size.

## Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. Learn more

| Choose storage class transitions | Days after object creation | |
|---|---|---|
| Standard-IA | 30 | Remove |
| Intelligent-Tiering | 60 | Remove |

Add transition

## Review transition and expiration actions

### Current version actions

Day 0

- Objects uploaded

↓

Day 30

- Objects move to Standard-IA

↓

Day 60

- Objects move to Intelligent-Tiering

↓

Day 180

- Objects move to Glacier Flexible Retrieval (formerly Glacier)

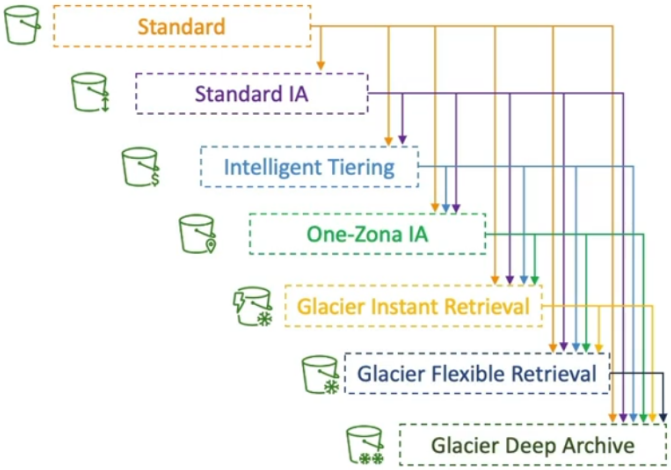
### Noncurrent versions actions

Day 0

No actions defined.

# Amazon S3 – Moving between Storage Classes

- You can transition objects between storage classes
- For infrequently accessed object, 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 a **Lifecycle Rules**




# 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 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 a 365 days
  - Can be used to delete old versions of files (if versioning is enabled)
  - Can be used to delete incomplete Multi-Part uploads
- Rules can be created for a certain prefix (example: *s3://mybucket/mp3/**)
- Rules can be created for certain objects Tags (example: *Department: Finance*)

# Amazon S3 – Lifecycle Rules (Scenario 1)

- Your application on EC2 creates images thumbnails after profile photos are uploaded to Amazon S3. These thumbnails can be easily recreated, and only need to be kept for 60 days. The source images should be able to be immediately retrieved for these 60 days, and afterwards, the user can wait up to 6 hours. How would you design this?

- S3 source images can be on **Standard**, with a lifecycle configuration to transition them to **Glacier** after 60 days
- S3 thumbnails can be on **One-Zone IA**, with a lifecycle configuration to expire them (delete them) after 60 days

# Amazon S3 – Lifecycle Rules (Scenario 2)

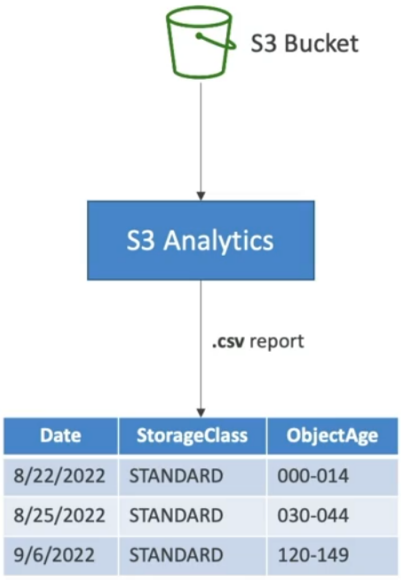
- A rule in your company states that you should be able to recover your deleted S3 objects immediately for 30 days, although this may happen rarely. After this time, and for up to 365 days, deleted objects should be recoverable within 48 hours.

- **Enable S3 Versioning** in order to have object versions, so that "deleted objects" are in fact hidden by a "delete marker" and can be recovered
- Transition the "noncurrent versions" of the object to **Standard IA**
- Transition afterwards the "noncurrent versions" to **Glacier Deep Archive**

# Amazon S3 Analytics – Storage Class Analysis

- Help you decide when to transition objects to the right storage class
- Recommendations for **Standard** and **Standard IA**
  - Does NOT work for One-Zone IA or Glacier
- Report is updated daily
- 24 to 48 hours to start seeing data analysis

- Good first step to put together Lifecycle Rules (or improve them)!



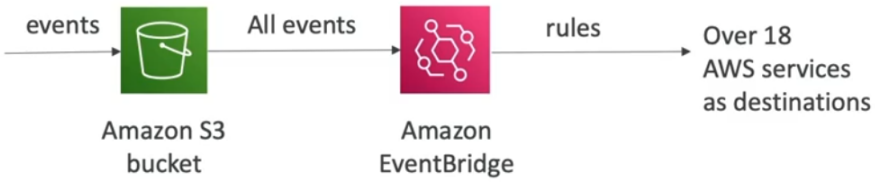

| Date | StorageClass | ObjectAge |
|---|---|---|
| 8/22/2022 | STANDARD | 000-014 |
| 8/25/2022 | STANDARD | 030-044 |
| 9/6/2022 | STANDARD | 120-149 |

# S3 Event Notifications

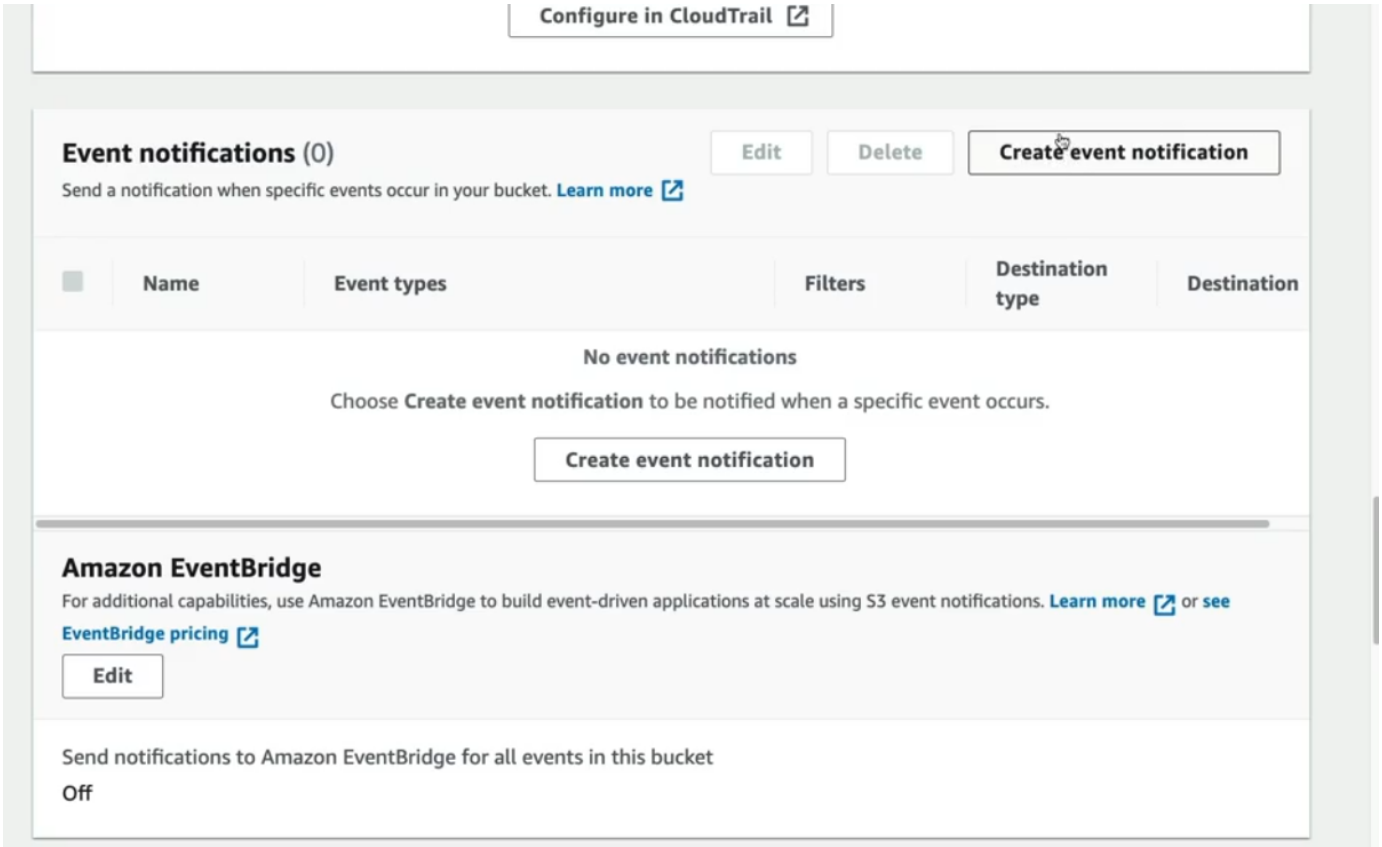
- S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication…
- Object name filtering possible (*.jpg)
- Use case: generate thumbnails of images uploaded to S3
- Can create as many "S3 events" as desired

- S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer

# S3 Event Notifications with Amazon EventBridge




- Advanced filtering options with JSON rules (metadata, object size, name...)
- Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose…
- EventBridge Capabilities – Archive, Replay Events, Reliable delivery



Amazon S3 > Buckets > demo-stephane-v3-event-notifications > Create event notification

# Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

## General configuration

Event name

DemoEventNot

Event name can contain up to 255 characters.

Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

*images/*

Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

*.jpg*

## Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

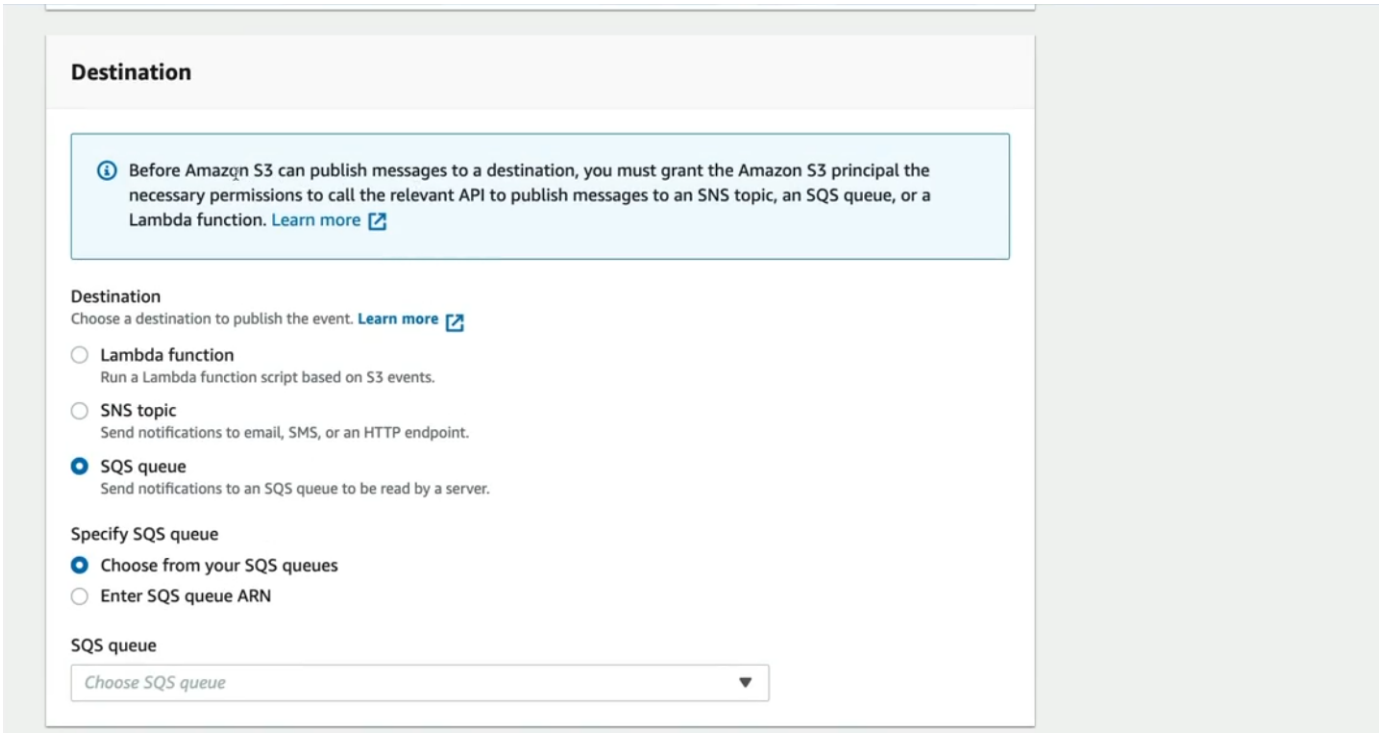
### Object creation

- [ ] All object create events
  s3:ObjectCreated:*
- [x] Put
  s3:ObjectCreated:Put
- [x] Post
  s3:ObjectCreated:Post
- [ ] Copy
  s3:ObjectCreated:Copy
- [ ] Multipart upload completed
  s3:ObjectCreated:CompleteMultipartUpload

### Object removal

- [ ] All object removal events
  s3:ObjectRemoved:*
- [ ] Permanently deleted
  s3:ObjectRemoved:Delete
- [ ] Delete marker created
  s3:ObjectRemoved:DeleteMarkerCreated

### Object restore

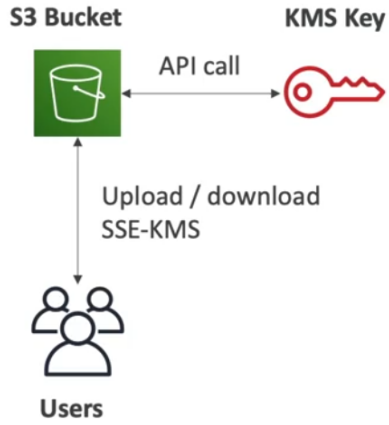



# S3 – Baseline Performance

- Amazon S3 automatically scales to high request rates, latency 100-200 ms
- Your application can achieve at least 3,500 PUT/COPY/POST/DELETE and 5,500 GET/HEAD requests per second per prefix in a bucket.
- There are no limits to the number of prefixes in a bucket.
- Example (object path => prefix):
  - bucket/folder1/sub1/file => /folder1/sub1/
  - bucket/folder1/sub2/file => /folder1/sub2/
  - bucket/1/file => /1/
  - bucket/2/file => /2/
- If you spread reads across all four prefixes evenly, you can achieve 22,000 requests per second for GET and HEAD

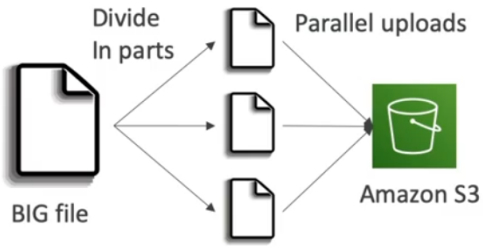
# S3 – KMS Limitation

- If you use SSE-KMS, you may be impacted by the KMS limits
- When you upload, it calls the **GenerateDataKey** KMS API
- When you download, it calls the **Decrypt** KMS API
- Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)
- You can request a quota increase using the Service Quotas Console

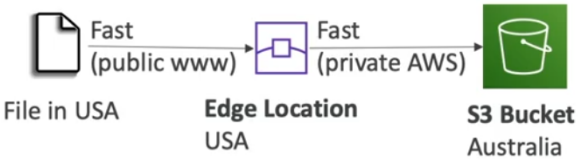



# S3 Performance

- Multi-Part upload:
  - recommended for files > 100MB, must use for files > 5GB
  - Can help parallelize uploads (speed up transfers)



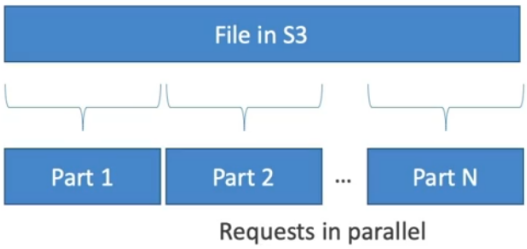

- S3 Transfer Acceleration
  - Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
  - Compatible with multi-part upload



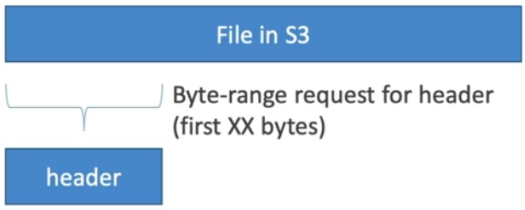
# S3 Performance – S3 Byte-Range Fetches

- Parallelize GETs by requesting specific byte ranges
- Better resilience in case of failures

Can be used to speed up downloads

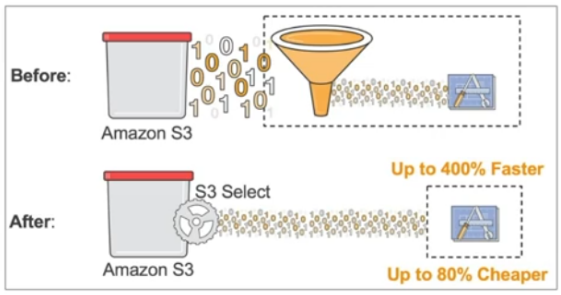



Can be used to retrieve only partial data (for example the head of a file)

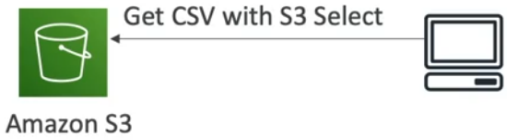



# S3 Select & Glacier Select

- Retrieve less data using SQL by performing **server-side filtering**
- Can filter by rows & columns (simple SQL statements)
- Less network transfer, less CPU cost client-side




https://aws.amazon.com/blogs/aws/s3-glacier-select/



# Amazon S3 – Object Encryption

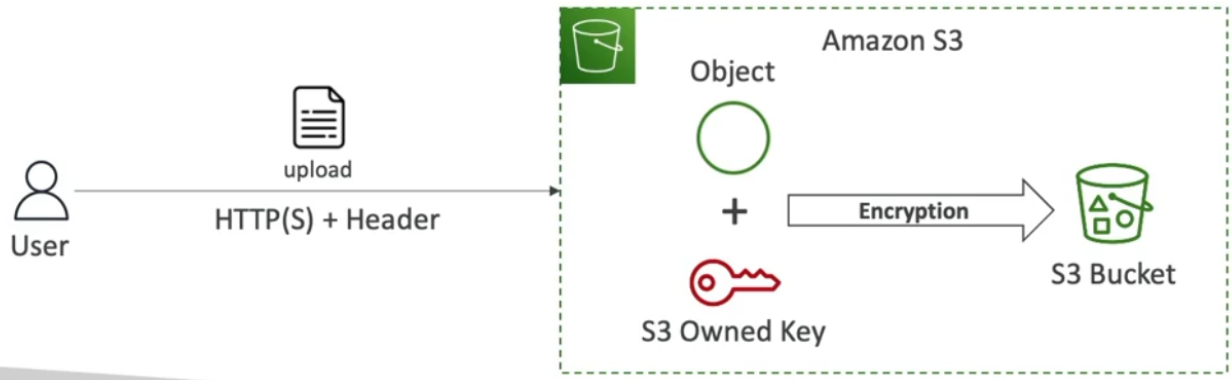
- You can encrypt objects in S3 buckets using one of 4 methods

- Server-Side Encryption (SSE)
  - Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
    - Encrypts S3 objects using keys handled, managed, and owned by AWS
  - Server-Side Encryption with KMS Keys stored in AWS KMS (SSE-KMS)
    - Leverage AWS Key Management Service (AWS KMS) to manage encryption keys
  - Server-Side Encryption with Customer-Provided Keys (SSE-C)
    - When you want to manage your own encryption keys
- Client-Side Encryption

- It's important to understand which ones are for which situation for the exam

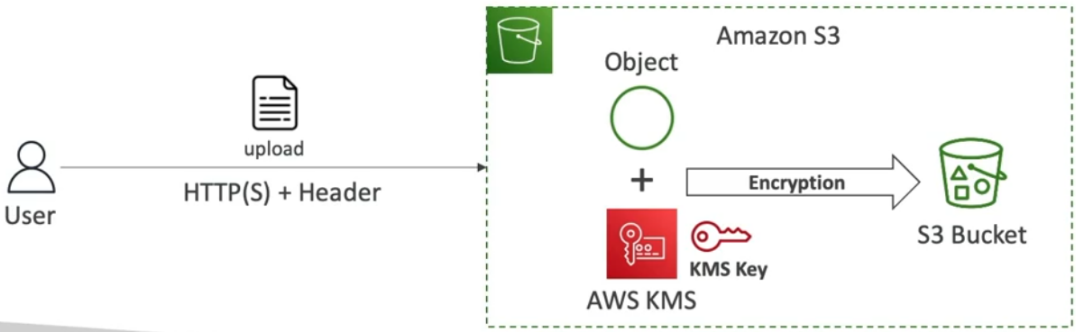
# Amazon S3 Encryption – SSE-S3

- Encryption using keys handled, managed, and owned by AWS
- Object is encrypted server-side
- Encryption type is AES-256
- Must set header "x-amz-server-side-encryption": "AES256"



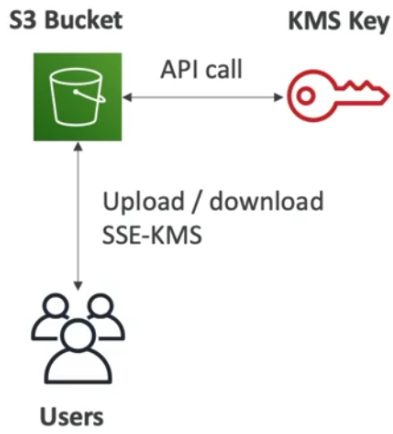
# Amazon S3 Encryption – SSE-KMS

- Encryption using keys handled and managed by AWS KMS (Key Management Service)
- KMS advantages: user control + audit key usage using CloudTrail
- Object is encrypted server side
- Must set header **"x-amz-server-side-encryption": "aws:kms"**



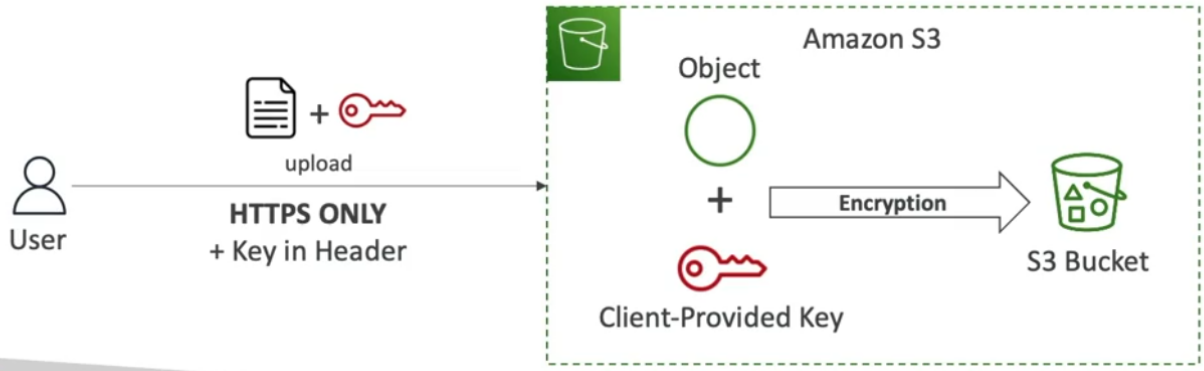

# SSE-KMS Limitation

- If you use SSE-KMS, you may be impacted by the KMS limits
- When you upload, it calls the **GenerateDataKey** KMS API
- When you download, it calls the **Decrypt** KMS API
- Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)
- You can request a quota increase using the Service Quotas Console



# Amazon S3 Encryption – SSE-C

- Server-Side Encryption using keys fully managed by the customer outside of AWS
- Amazon S3 does NOT store the encryption key you provide
- HTTPS must be used
- Encryption key must provided in HTTP headers, for every HTTP request made




# Amazon S3 Encryption – Client-Side Encryption

- Use client libraries such as Amazon S3 Client-Side Encryption Library
- Clients must encrypt data themselves before sending to Amazon S3
- Clients must decrypt data themselves when retrieving from Amazon S3
- Customer fully manages the keys and encryption cycle

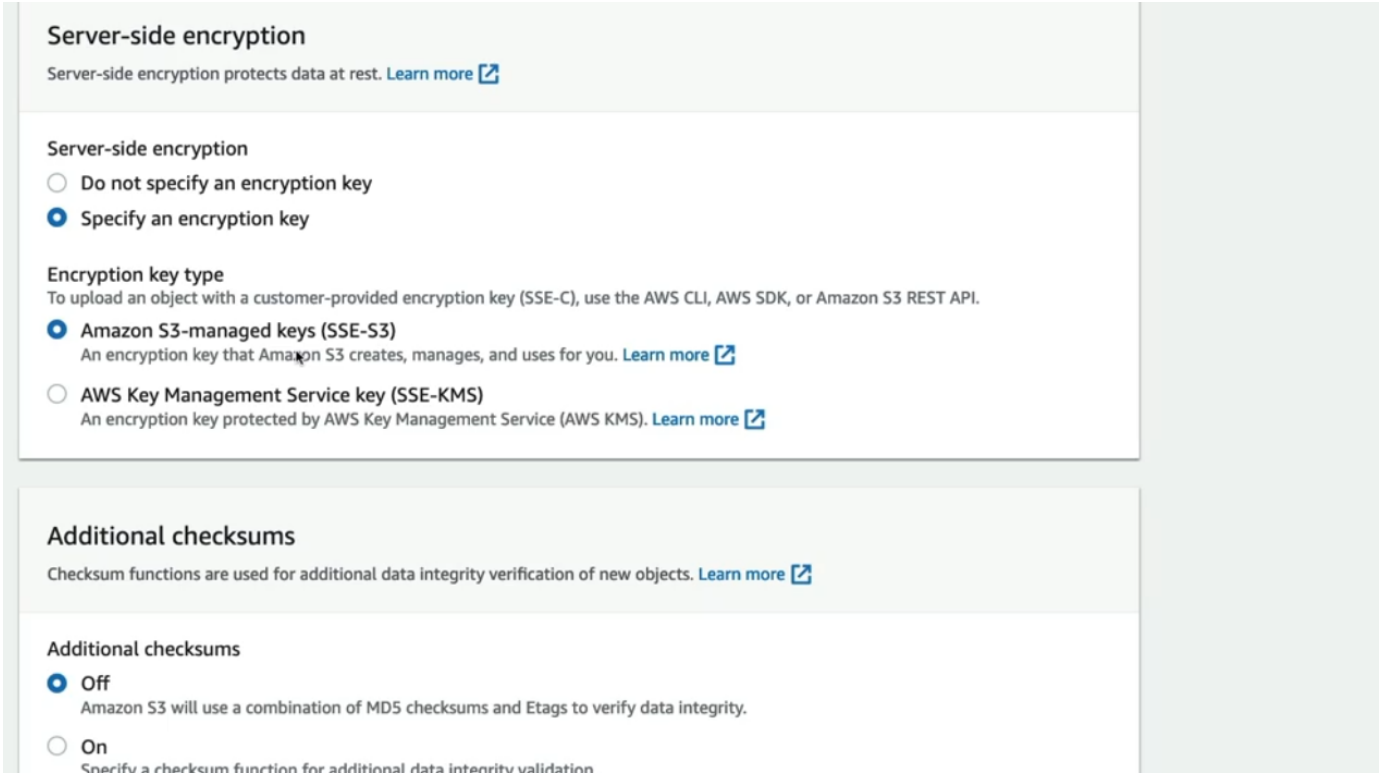
# Amazon S3 – Encryption in transit (SSL/TLS)

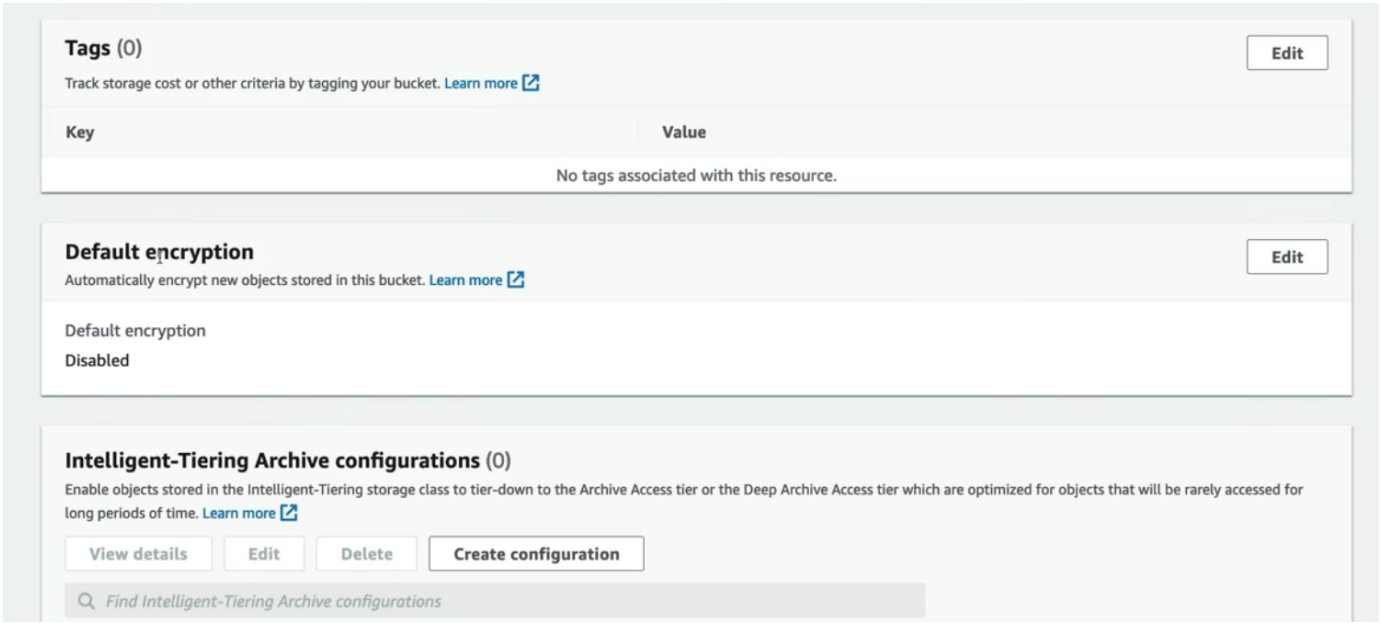
- Encryption in flight is also called SSL/TLS

- Amazon S3 exposes two endpoints:
  - HTTP Endpoint – non encrypted
  - HTTPS Endpoint – encryption in flight

- HTTPS is recommended
- HTTPS is mandatory for SSE-C
- Most clients would use the HTTPS endpoint by default

under properties -> serverside encryption






# Amazon S3 – Default Encryption vs. Bucket Policies

- One way to "force encryption" is to use a bucket policy and refuse any API call to PUT an S3 object without encryption headers

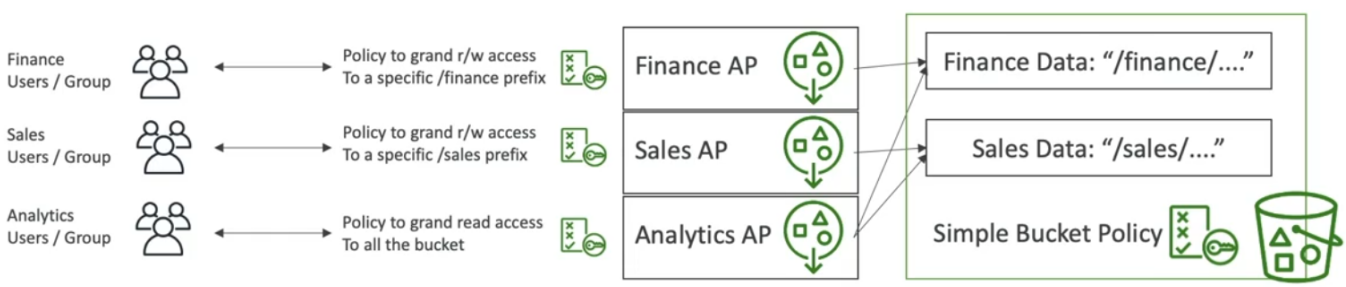
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyIncorrectDecryptionHeader",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [ "s3:PutObject" ],
            "Resource": [ "arn:aws:s3:::examplebucket/*" ],
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-server-side-encryption": "AES256"
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyUnencryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [ "s3:PutObject" ],
            "Resource": [ "arn:aws:s3:::examplebucket/*" ],
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption": true
                }
            }
        }
    ]
}
```

- Another way is to use the "default encryption" option in S3

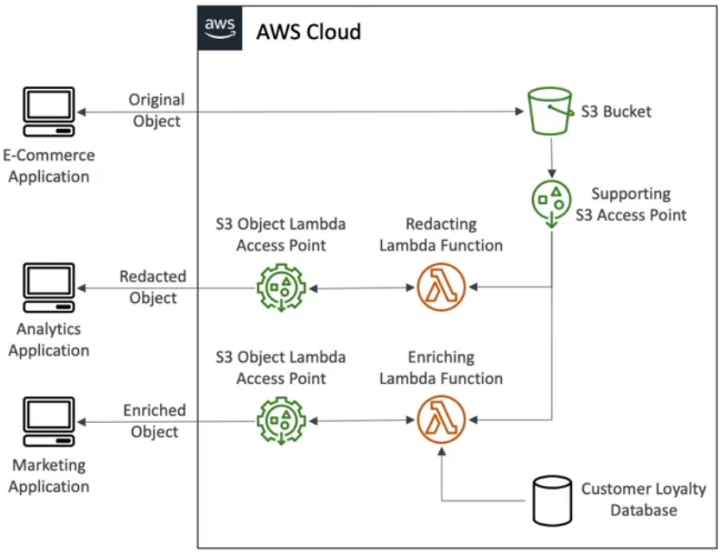
# S3 – Access Points

- Each Access Point gets its own DNS and policy to limit who can access it
  - A specific IAM user / group
  - One policy per Access Point => Easier to manage than complex bucket policies



# S3 Object Lambda

- Use AWS Lambda Functions to change the object before it is retrieved by the caller application
- Only one S3 bucket is needed, on top of which we create S3 Access Point and S3 Object Lambda Access Points.




- Use Cases:
  - Redacting personally identifiable information for analytics or non-production environments.
  - Converting across data formats, such as converting XML to JSON.
  - Resizing and watermarking images on the fly using caller-specific details, such as the user who requested the object.