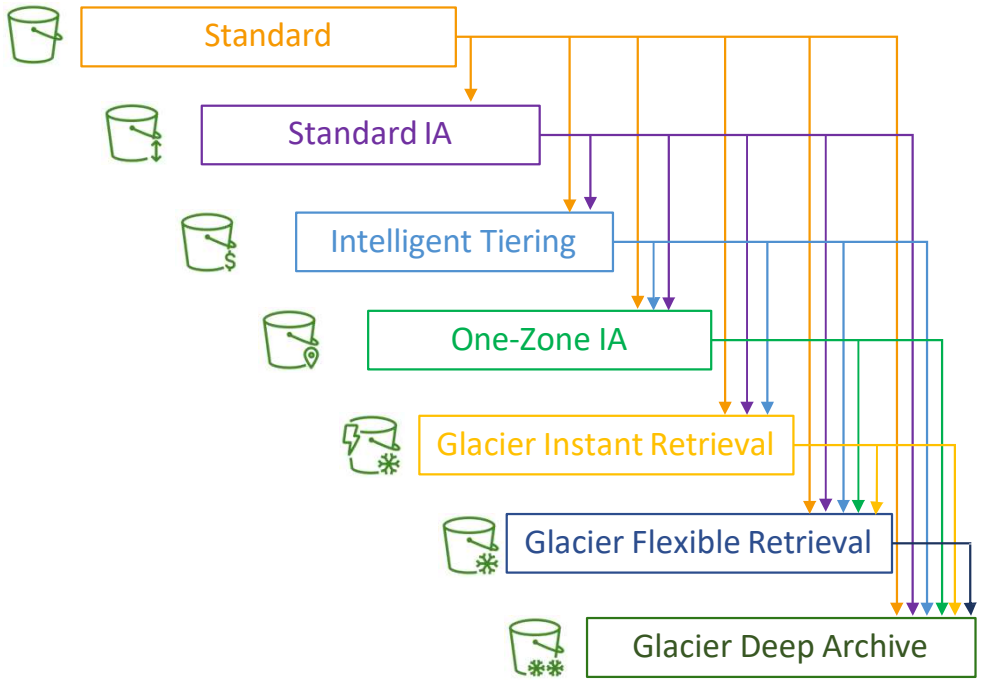
# Amazon S3 - Advanced

# 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)

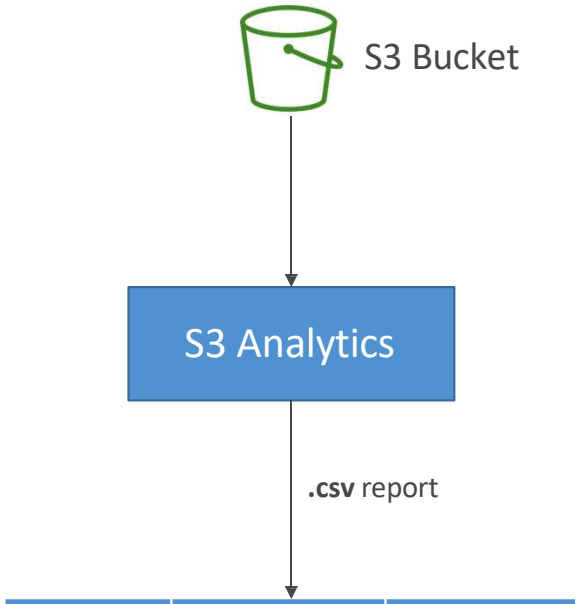
- A rule in your company states that you should be able to recover your deleted S3 objects immediately for 30 days, although this may happen rarely. After this time, and for up to 365 days, deleted objects should be recoverable within 48 hours.

- 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 dass
- 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

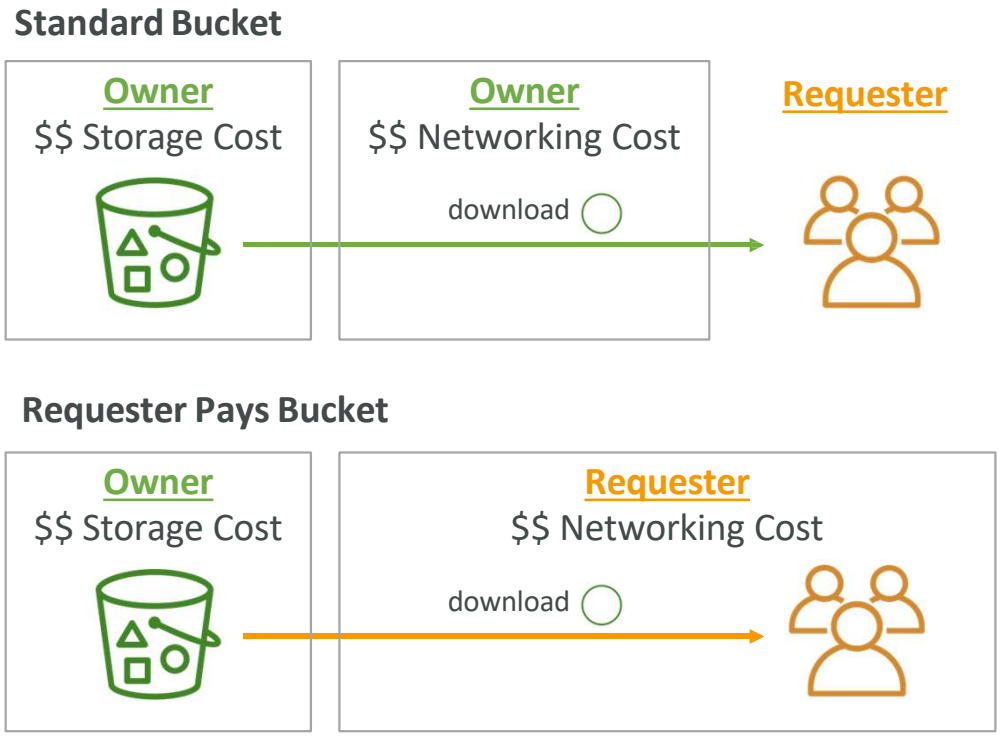
- Good first step to put together Lifecycle Rules (or improve them)!




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

# S3 - Requester Pays

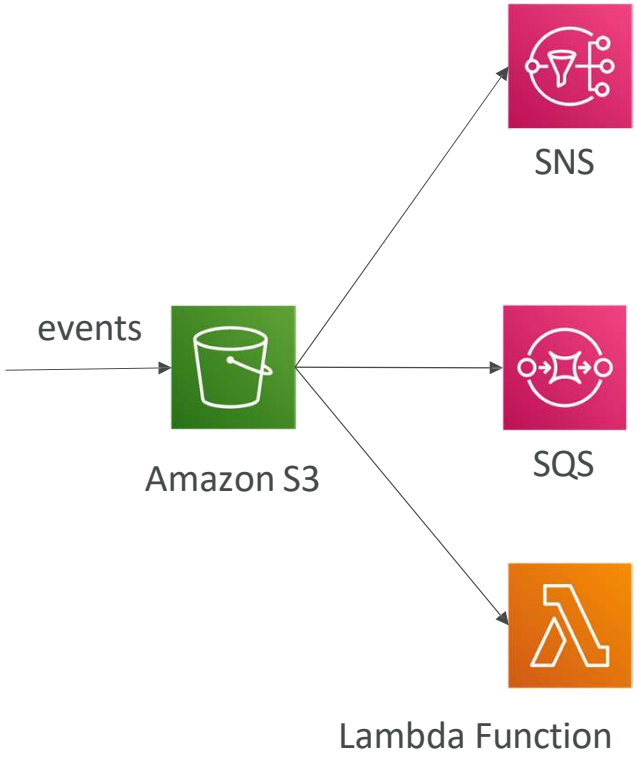
- In general, bucket owners pay for all Amazon S3 storage and data transfer costs associated with their bucket
- With Requester Pays buckets, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket
- Helpful when you want to share large datasets with other accounts
- The requester must be authenticated in AWS (cannot be anonymous)



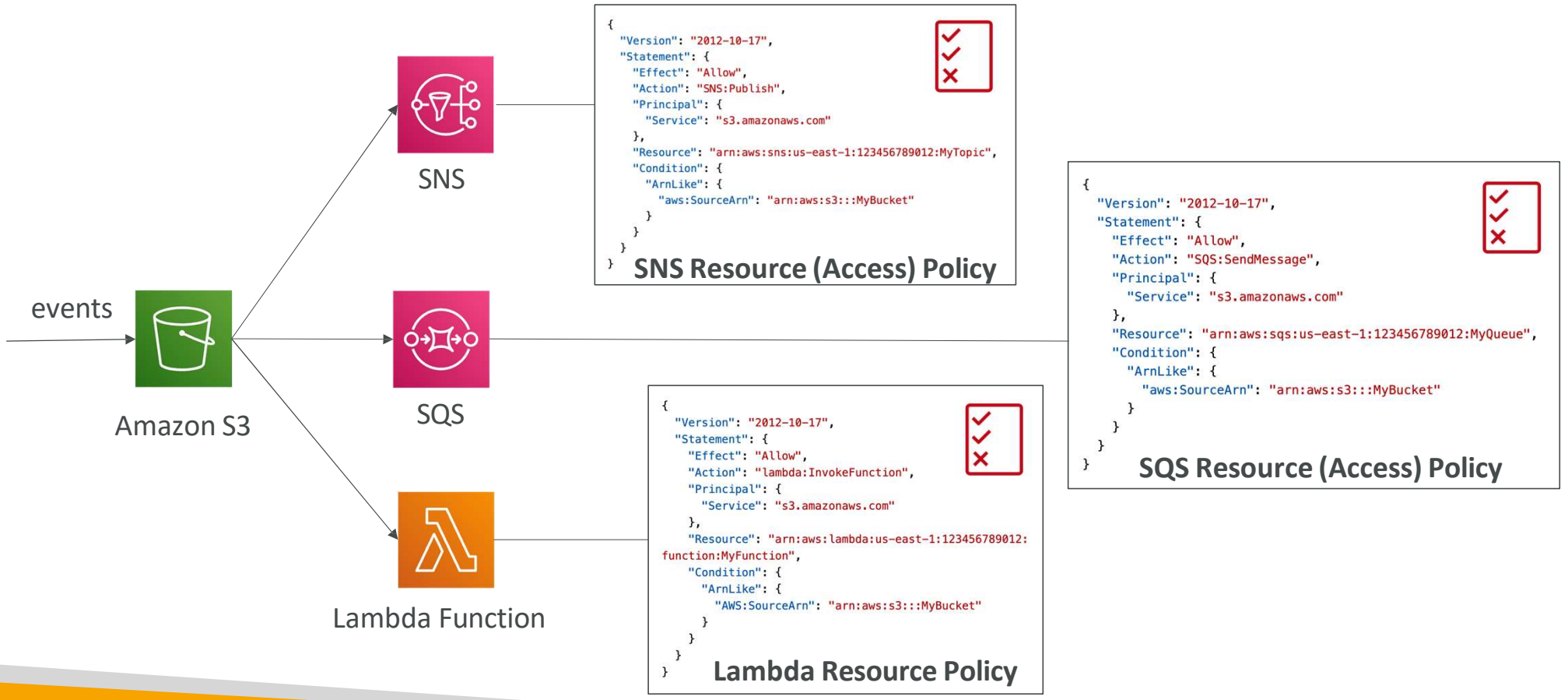
# S3 Event Notifications

- 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

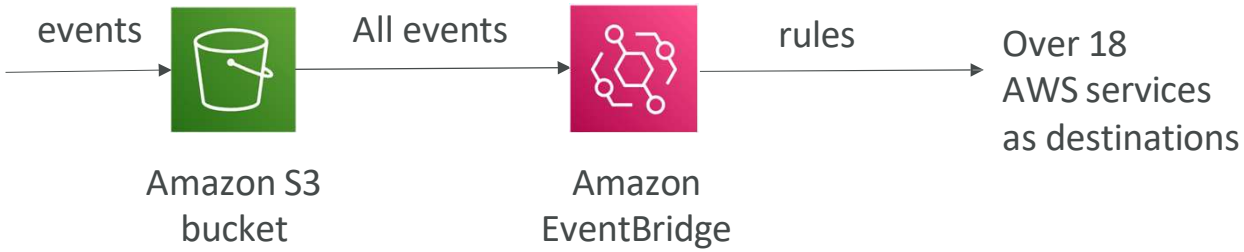
- S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer



# S3 Event Notifications - IAM Permissions



# S3 Event Notifications with Amazon EventBridge



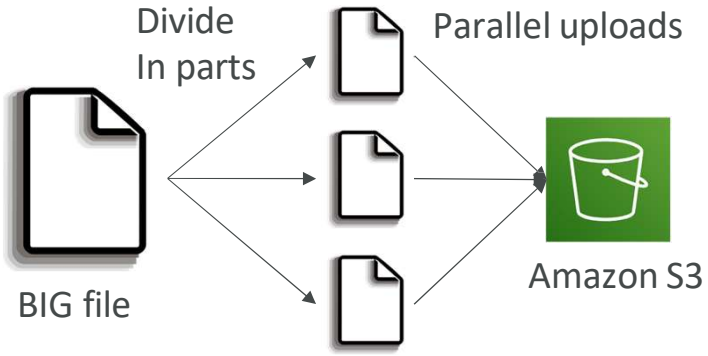

- Advanced filtering options with JSON rules (metadata, object size, name...)
- Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose...
- EventBridge Capabilities – Archive, Replay Events, Reliable delivery

# S3 - Baseline Performance

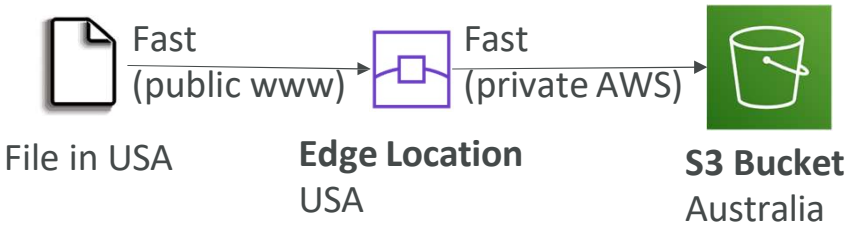
- Amazon S3 automatically scales to high request rates, latency 100-200 ms
- Your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket.
- There are no limits to the number of prefixes in a bucket.
- Example (object path => prefix):
  - 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 Performance

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



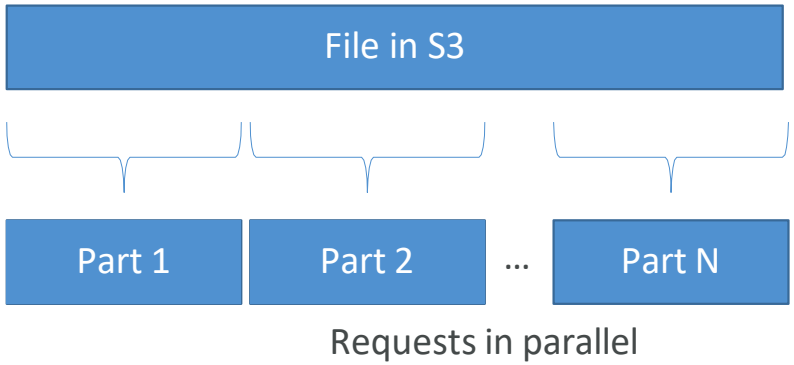

- S3 Transfer Acceleration
  - Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
  - Compatible with multi-part upload



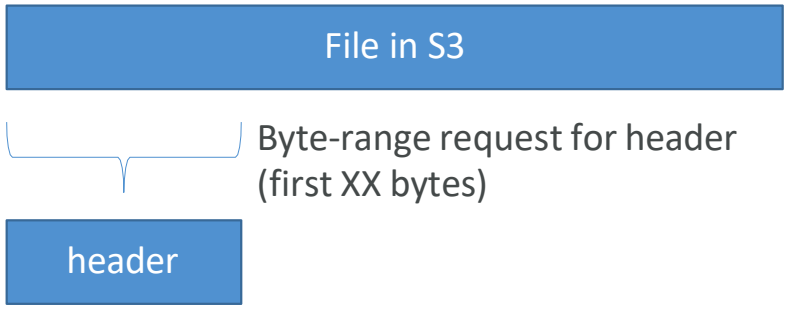
# S3 Performance – S3 Byte-Range Fetches

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

Can be used to speed up downloads

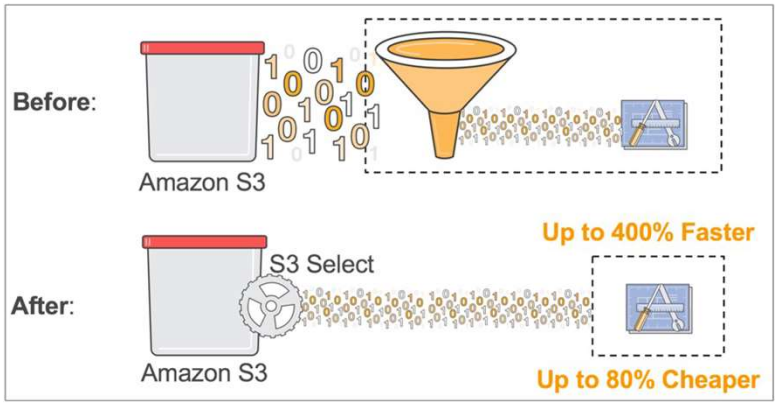



Can be used to retrieve only partial data (for example the head of a file)



# S3 Select & Glacier Select

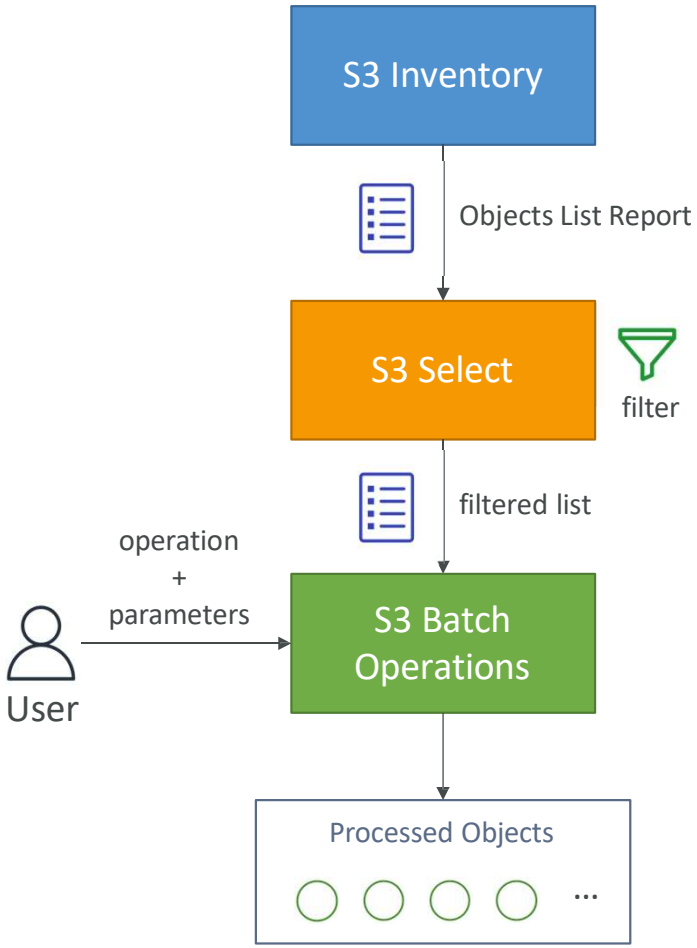
- Retrieve less data using SQL by performing server-side filtering
- Can filter by rows & columns (simple SQL statements)
- Less network transfer, less CPU cost client-side




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

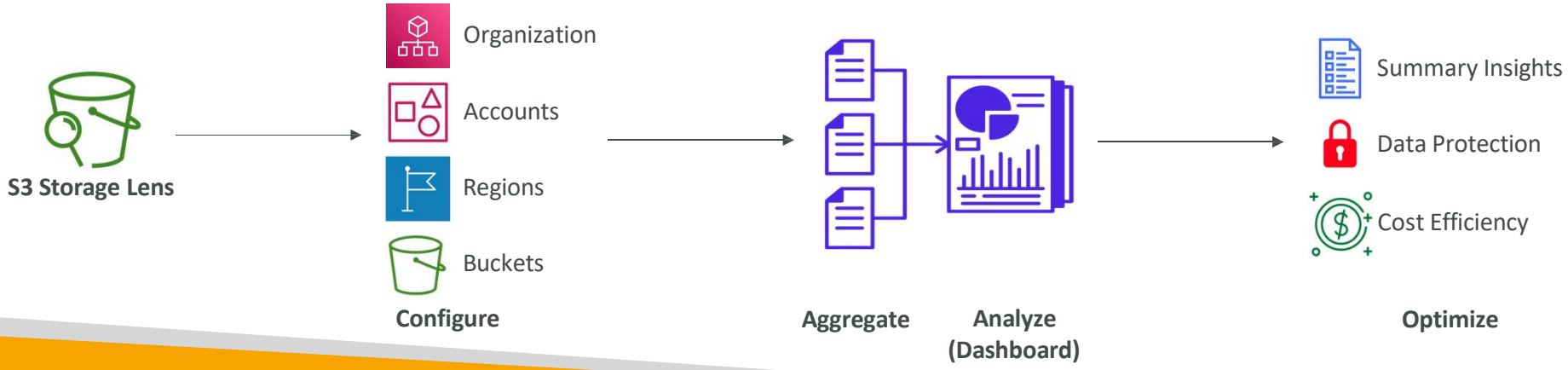
# S3 Batch Operations

- Perform bulk operations on existing S3 objects with a single request, example:
  - Modify object metadata & properties
  - Copy objects between S3 buckets
  - Encrypt un-encrypted objects
  - Modify ACLs, tags
  - Restore objects from S3 Glacier
  - Invoke Lambda function to perform custom action on each object
- A job consists of a list of objects, the action to perform, and optional parameters
- S3 Batch Operations manages retries, tracks progress, sends completion notifications, generate reports ...
- You can use S3 Inventory to get object list and use S3 Select to filter your objects



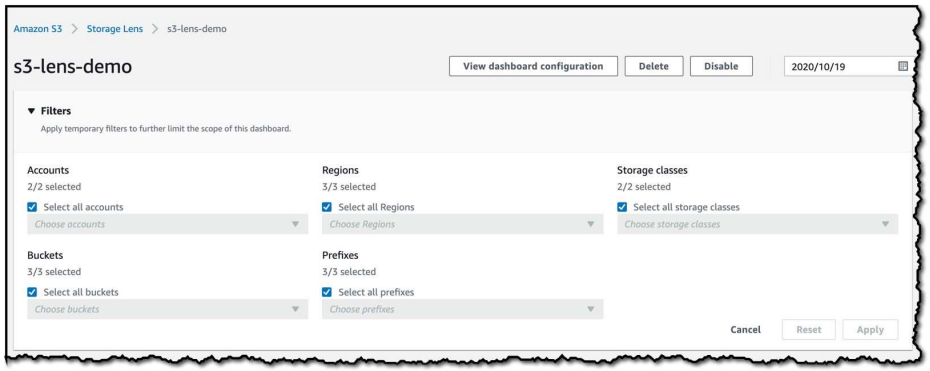
# S3 - Storage Lens

- Understand, analyze, and optimize storage across entire AWS Organization
- Discover anomalies, identify cost efficiencies, and apply data protection best practices across entire AWS Organization (30 days usage & activity metrics)
- Aggregate data for Organization, specific accounts, regions, buckets, or prefixes
- Default dashboard or create your own dashboards
- Can be configured to export metrics daily to an S3 bucket (CSV, Parquet)

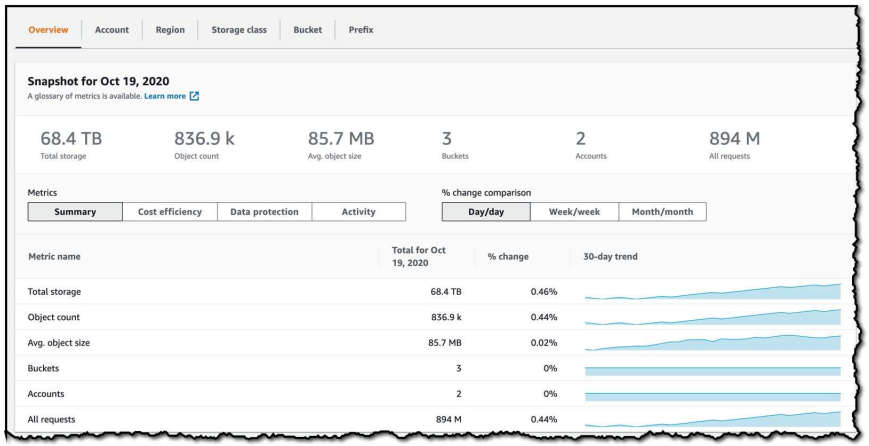


# Storage Lens - Default Dashboard

- Visualize summarized insights and trends for both free and advanced metrics
- Default dashboard shows Multi-Region and Multi-Account data
- Preconfigured by Amazon S3
- Can't be deleted, but can be disabled




https://aws.amazon.com/blogs/aws/s3-storage-lens/




https://aws.amazon.com/blogs/aws/s3-storage-lens/

# Storage Lens – Metrics

- Summary Metrics
  - General insights about your S3 storage
  - StorageBytes, ObjectCount...
  - Use cases: identify the fastest-growing (or not used) buckets and prefixes

- Cost-Optimization Metrics
  - Provide insights to manage and optimize your storage costs
  - NonCurrentVersionStorageBytes, IncompleteMultipartUploadStorageBytes...
  - Use cases: identify buckets with incomplete multipart uploaded older than 7 days, Identify which objects could be transitioned to lower-cost storage dass

# Storage Lens – Metrics

- Data-Protection Metrics
  - Provide insights for data protection features
  - VersioningEnabledBucketCount, MFADeleteEnabledBucketCount, SSEKMSEnabledBucketCount, CrossRegionReplicationRuleCount...
  - Use cases: identify buckets that aren't following data-protection best practices
- Access-management Metrics
  - Provide insights for S3 Object Ownership
  - ObjectOwnershipBucketOwnerEnforcedBucketCount...
  - Use cases: identify which Object Ownership settings your buckets use
- Event Metrics
  - Provide insights for S3 Event Notifications
  - EventNotificationEnabledBucketCount (identify which buckets have S3 Event Notifications configured)

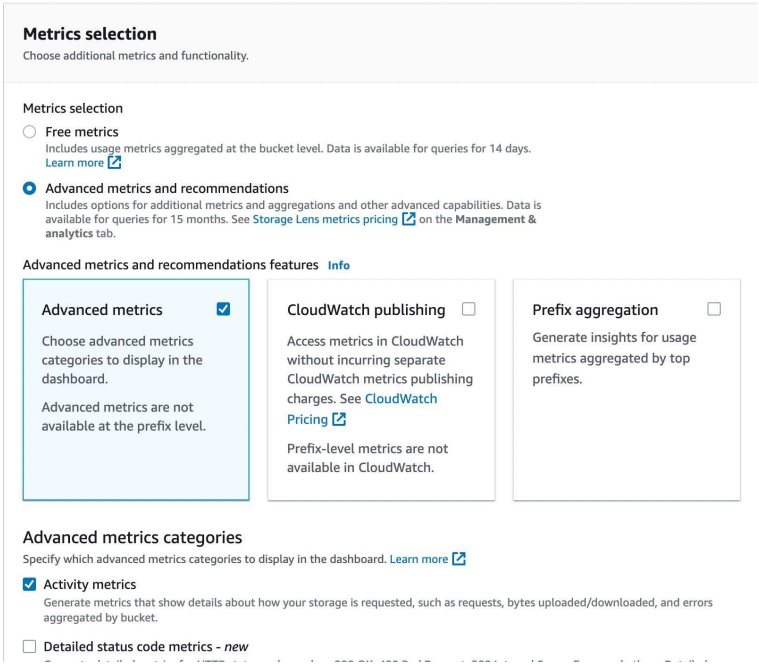
# Storage Lens – Metrics

- Performance Metrics
  - Provide insights for S3 Transfer Acceleration
  - TransferAccelerationEnabledBucketCount (identify which buckets have S3 Transfer Acceleration enabled)
- Activity Metrics
  - Provide insights about how your storage is requested
  - AllRequests, GetRequests, PutRequests, ListRequests, BytesDownloaded...
- Detailed Status Code Metrics
  - Provide insights for HTTP status codes
  - 200OKStatusCount, 403ForbiddenErrorCount, 404NotFoundErrorCount...

# Storage Lens - Free vs. Paid

- Free Metrics
  - Automatically available for all customers
  - Contains around 28 usage metrics
  - Data is available for queries for 14 days

- 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



# Amazon S3 - Security

# Amazon S3 – Object Encryption

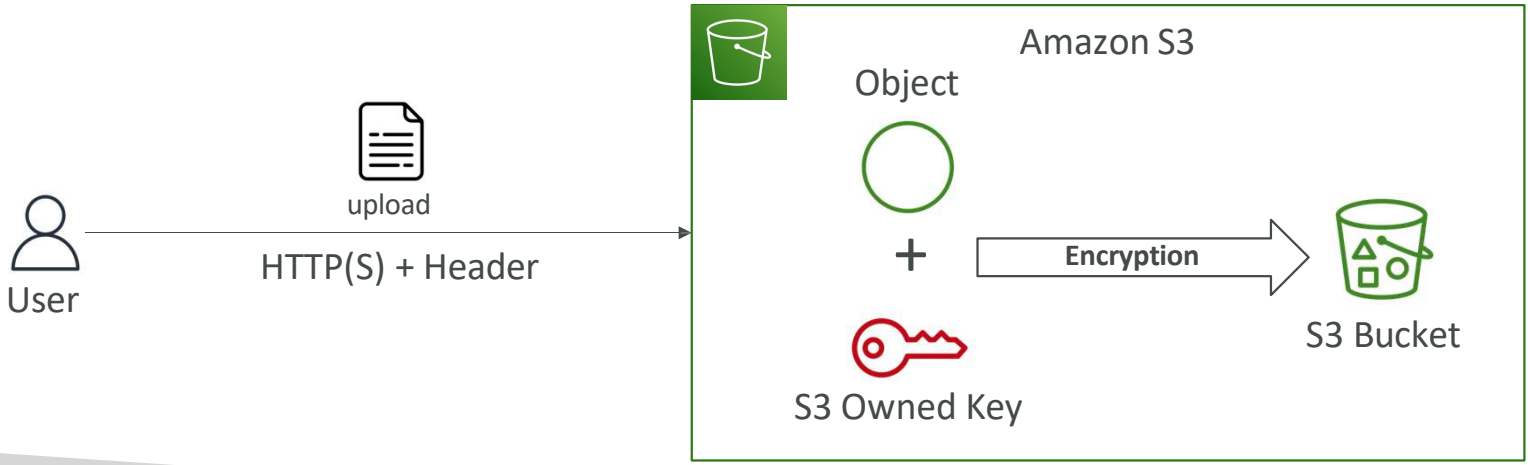
- You can encrypt objects in S3 buckets using one of 4 methods

- Server-Side Encryption (SSE)
  - Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) – Enabled by Default
    - 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

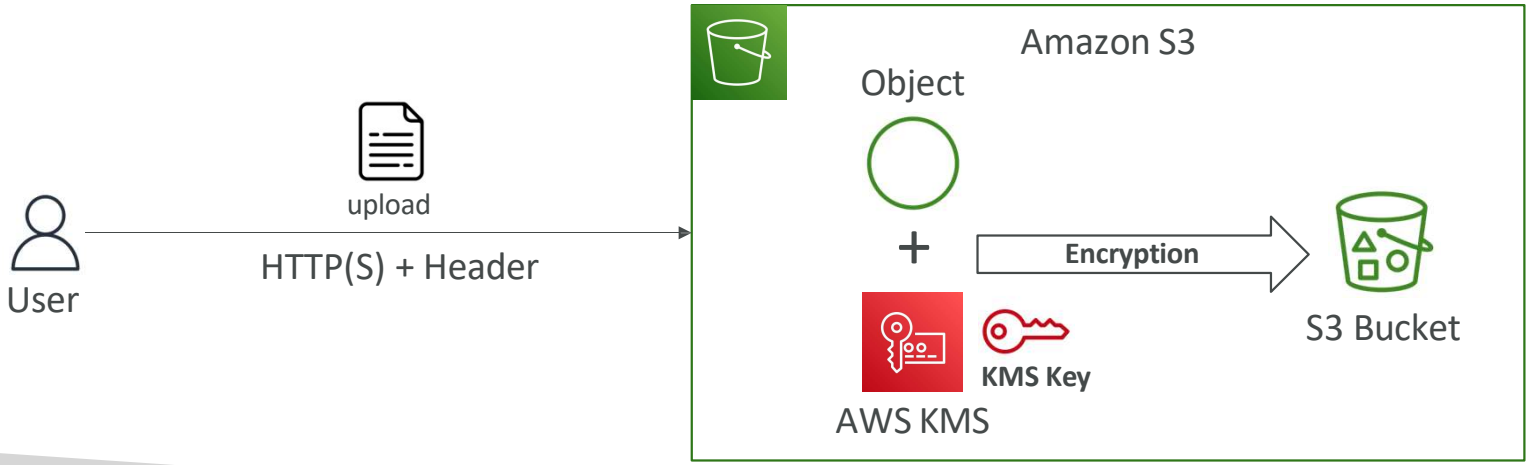
# Amazon S3 Encryption - SSE-S3

- 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"
- Enabled by default for new buckets & new objects



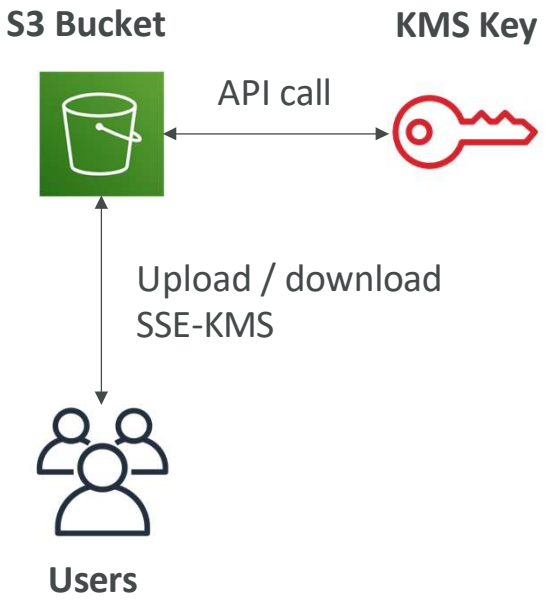
# Amazon S3 Encryption - SSE-KMS

- 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"



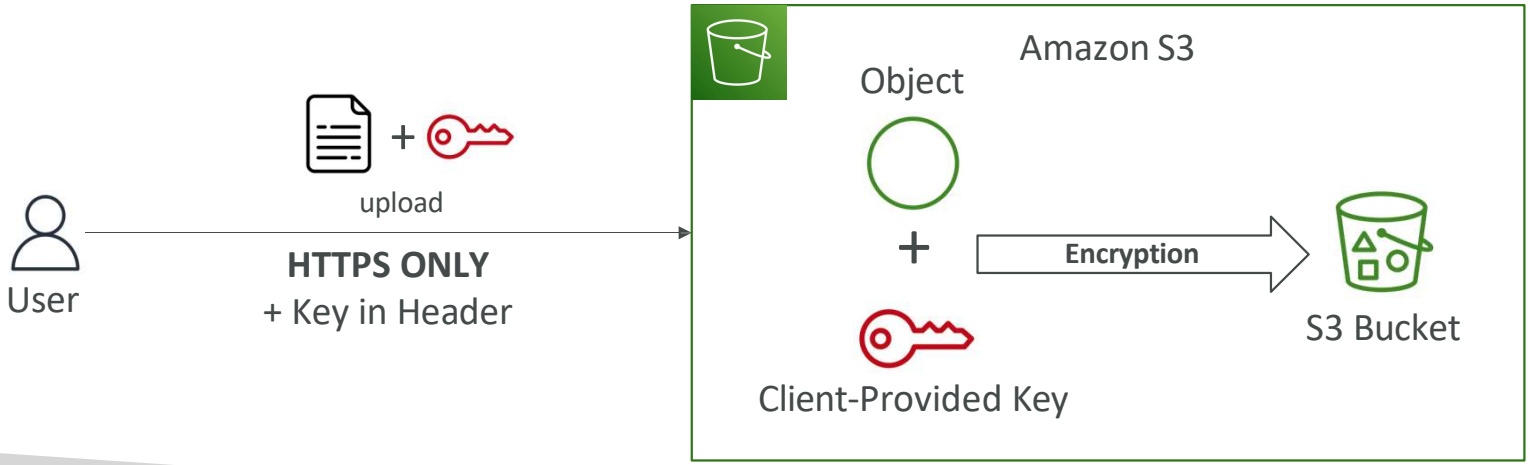
# SSE-KMS Limitation

- 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



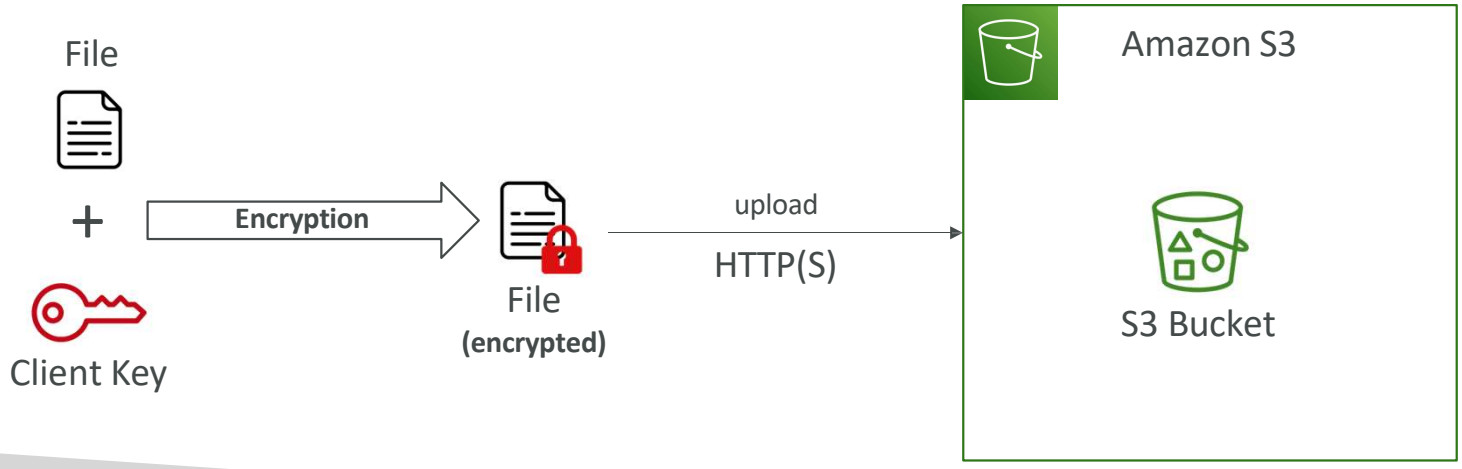
# Amazon S3 Encryption - SSE-C

- 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



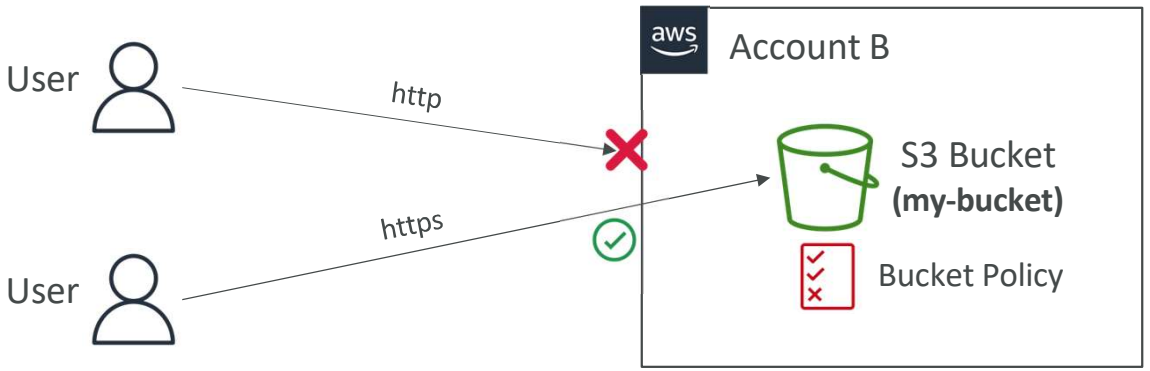
# Amazon S3 – Encryption in transit (SSL/TLS)

- 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

# Amazon S3 - Force Encryption in Transit aws:SecureTransport




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-bucket/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

# Amazon S3 - Default Encryption vs. Bucket Policies

- SSE-S3 encryption is automatically applied to new objects stored in S3 bucket
- Optionally, you can "force encryption" using a bucket policy and refuse any API call to PUT an S3 object without encryption headers (SSE-KMS or SSE-C)

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

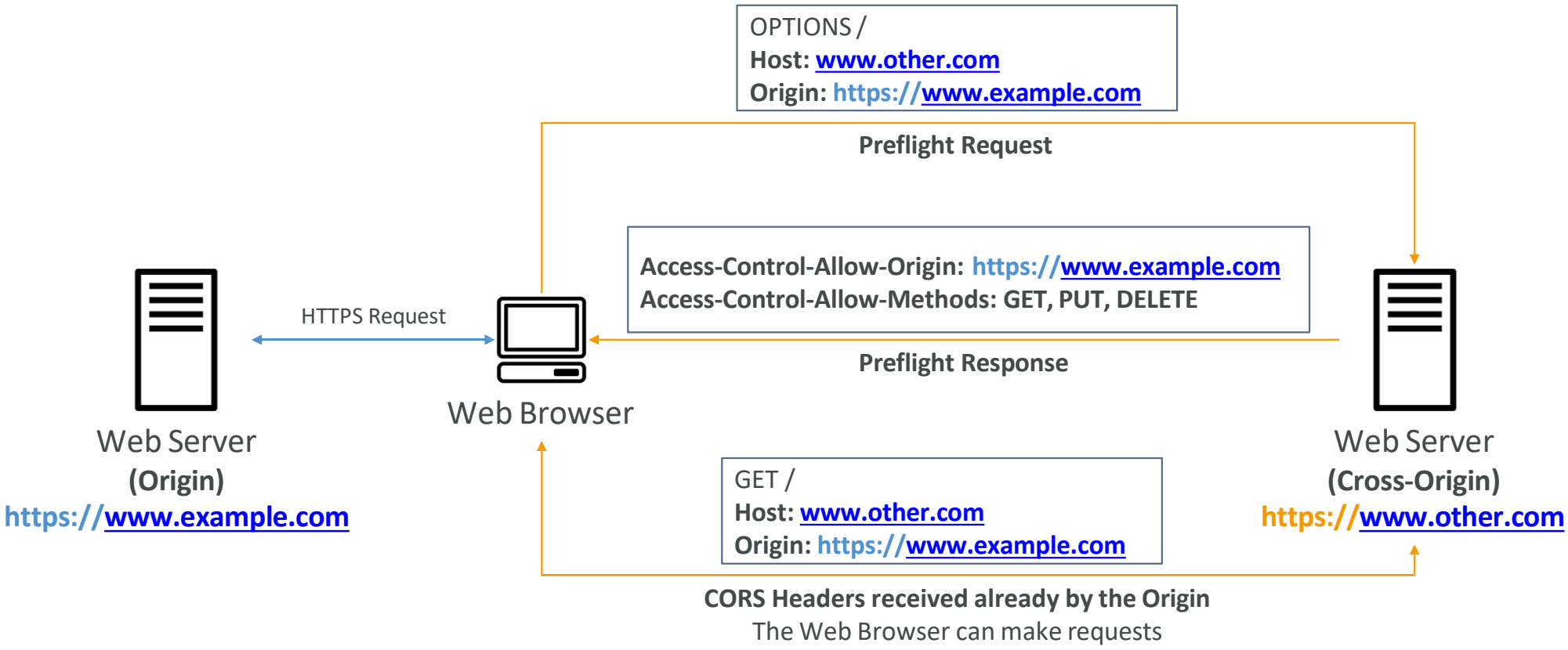
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "s3:PutObject",
            "Principal": "*",
            "Resource": "arn:aws:s3:::my-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm":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

- Note: Bucket Policies are evaluated before "Default Encryption"

# What is CORS?

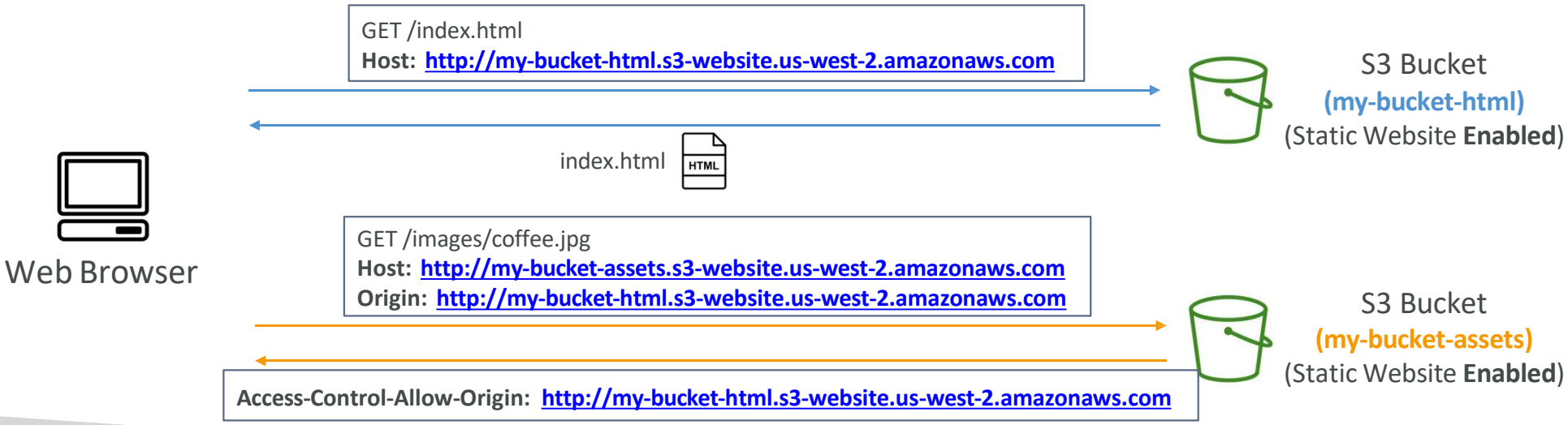
- Cross-Origin Resource Sharing (CORS)
- Origin = scheme (protocol) + host (domain) + port
  - example: https://www.example.com (implied port is 443 for HTTPS, 80 for HTTP)
- Web Browser based mechanism to allow requests to other origins while visiting the main origin
- Same origin: http://example.com/app1 & http://example.com/app2
- Different origins: http://www.example.com & http://other.example.com
- The requests won't be fulfilled unless the other origin allows for the requests, using CORS Headers (example: Access-Control-Allow-Origin)

# What is CORS?


OPTIONS /
Host: www.other.com
Origin: https://www.example.com
Preflight Request
Access-Control-Allow-Origin: https://www.example.com
Access-Control-Allow-Methods: GET, PUT, DELETE
Preflight Response
HTTPS Request
Web Server
(Origin)
https://www.example.com
Web Browser
Web Server
(Cross-Origin)
https://www.other.com
GET /
Host: www.other.com
Origin: https://www.example.com
CORS Headers received already by the Origin
The Web Browser can make requests

# Amazon S3 - CORS

- If a client makes a cross-origin request on our S3 bucket, we need to enable the correct CORS headers
- It's a popular exam question
- You can allow for a specific origin or for * (all origins)



# Amazon S3 - MFA Delete

- MFA (Multi-Factor Authentication) - force users to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
- MFA will be required to:
  - Permanently delete an object version
  - Suspend Versioning on the bucket
- MFA won't be required to:
  - Enable Versioning
  - List deleted versions
- To use MFA Delete, Versioning must be enabled on the bucket
- Only the bucket owner (root account) can enable/disable MFA Delete

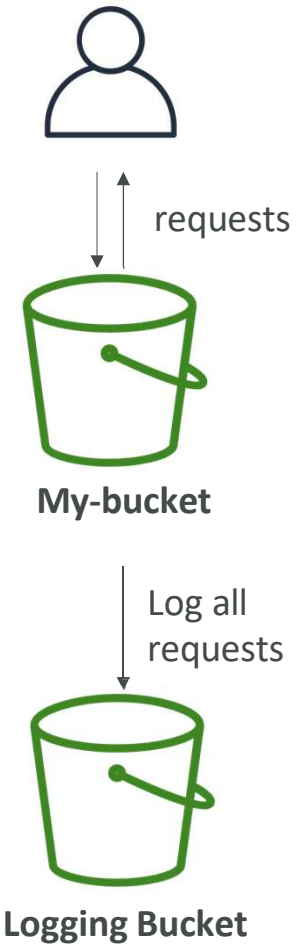
Google Authenticator

MFA Hardware Device

# S3 Access Logs

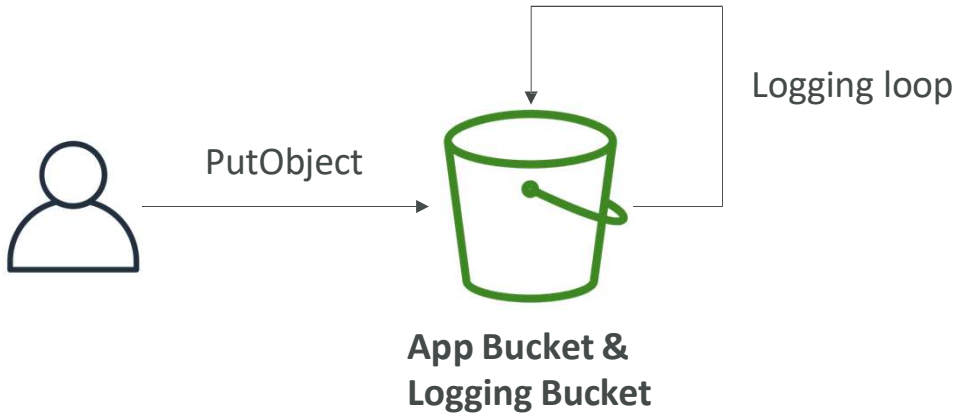
- For audit purpose, you may want to log all access to S3 buckets
- Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
- That data can be analyzed using data analysis tools...
- The target logging bucket must be in the same AWS region

- The log format is at: https://docs.aws.amazon.com/AmazonS3/latest/dev/LogFormat.html



# S3 Access Logs: Warning

- Do not set your logging bucket to be the monitored bucket
- It will create a logging loop, and your bucket will grow exponentially

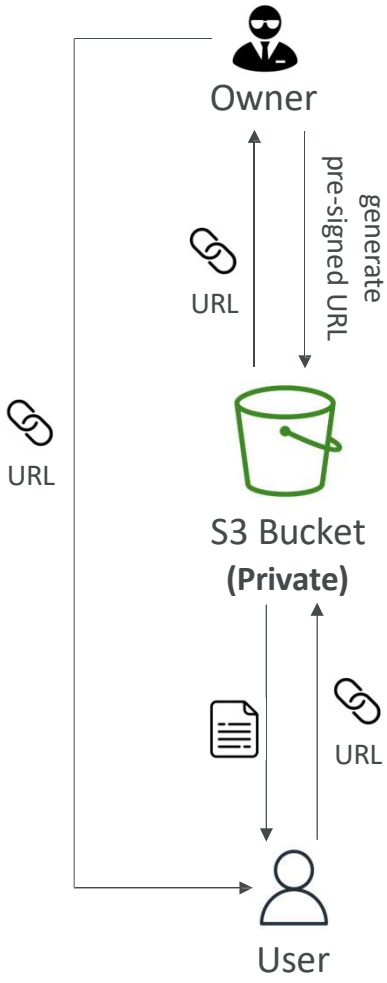



Do not try this at home ©

# Amazon S3 - Pre-Signed URLs

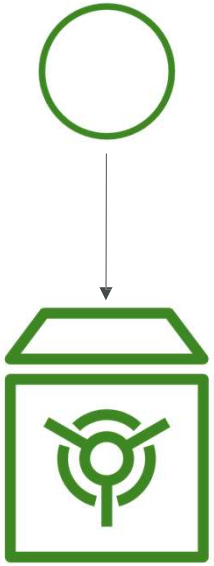
- Generate pre-signed URLs using the S3 Console, AWS CLI or SDK
- URL Expiration
  - S3 Console - 1 min up to 720 mins (12 hours)
  - AWS CLI - configure expiration with *–expires-in* parameter in seconds (default 3600 secs, max. 604800 secs ~ 168 hours)
- Users given a pre-signed URL inherit the permissions of the user that generated the URL for GET / PUT

- Examples:
  - Allow only logged-in users to download a premium video from your S3 bucket
  - Allow an ever-changing list of users to download files by generating URLs dynamically
  - Allow temporarily a user to upload a file to a precise location in your S3 bucket



# S3 Glacier Vault Lock

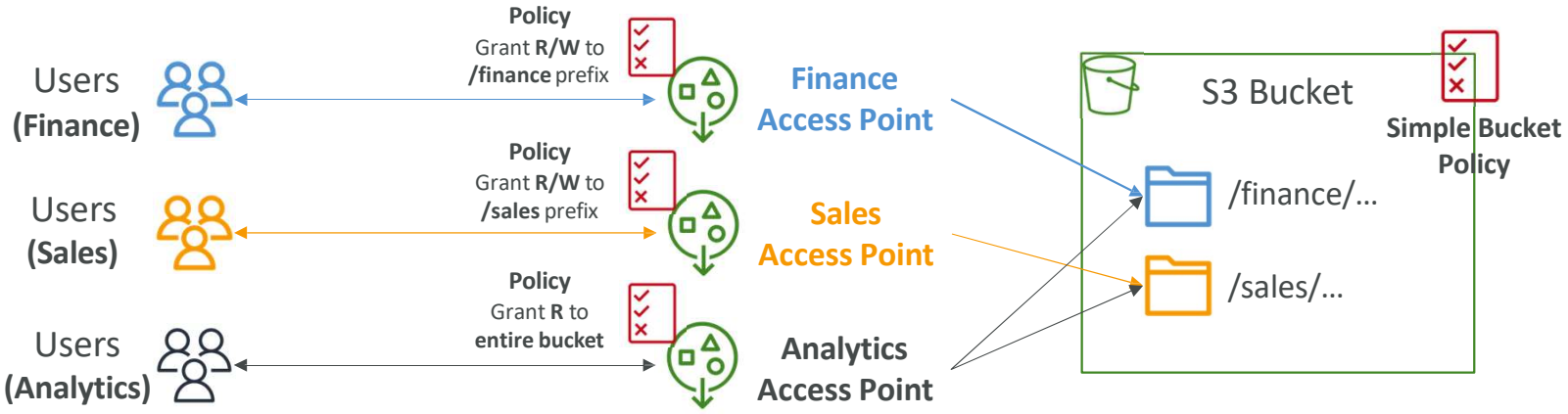
- Adopt a WORM (Write Once Read Many) model
- Create a Vault Lock Policy
- Lock the policy for future edits (can no longer be changed or deleted)
- Helpful for compliance and data retention



# S3 Object Lock (versioning must be enabled)

- Adopt a WORM (Write Once Read Many) model
- Block an object version deletion for a specified amount of time
- Retention mode - Compliance:
  - Object versions can't be overwritten or deleted by any user, including the root user
  - Objects retention modes can't be changed, and retention periods can't be shortened
- Retention mode - Governance:
  - Most users can't overwrite or delete an object version or alter its lock settings
  - Some users have special permissions to change the retention or delete the object
- Retention Period: protect the object for a fixed period, it can be extended
- Legal Hold:
  - protect the object indefinitely, independent from retention period
  - can be freely placed and removed using the *s3:PutObjectLegalHold* IAM permission

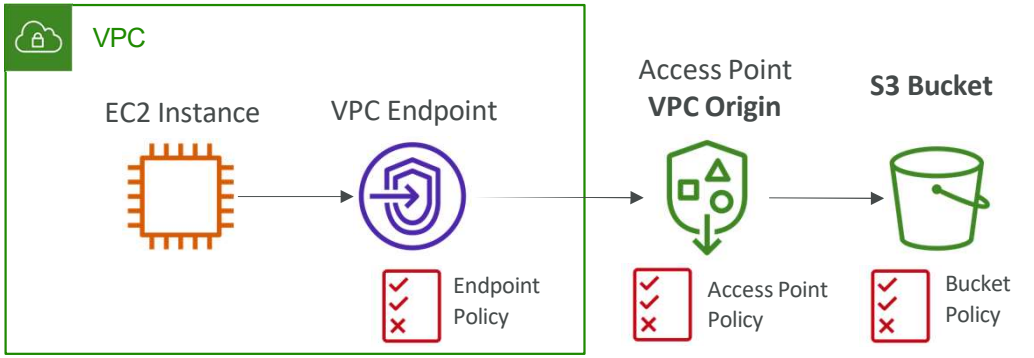
# S3 - Access Points




- Access Points simplify security management for S3 Buckets
- Each Access Point has:
  - its own DNS name (Internet Origin or VPC Origin)
  - an access point policy (similar to bucket policy) - manage security at scale

# S3 - Access Points - VPC Origin

- We can define the access point to be accessible only from within the VPC
- You must create a VPC Endpoint to access the Access Point (Gateway or Interface Endpoint)
- The VPC Endpoint Policy must allow access to the target bucket and Access Point




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
            ]
        }
    ]
}
```

# 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.