**Advance S3**

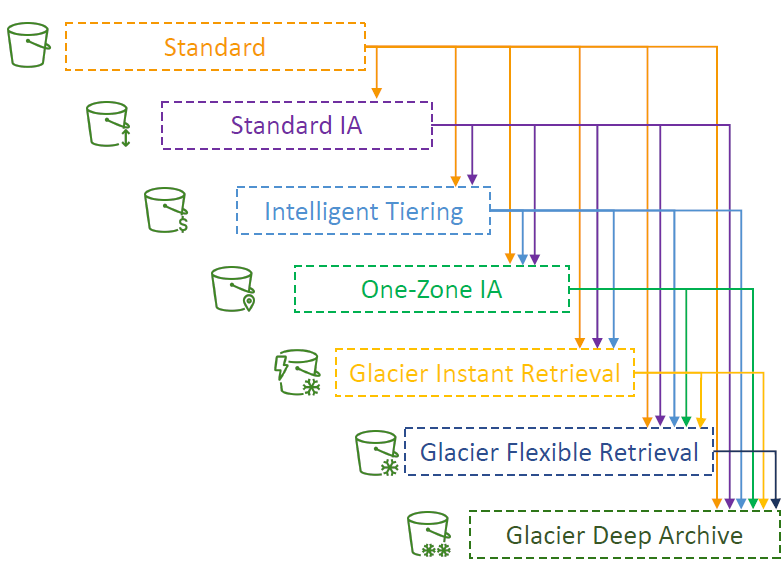
1. **Moving between storage classes**

In S3 you can transition objects between storage classes

- For infrequently accessed object, move them to Standard IA

- For archive objects that don't need fast access, move them to Glacier or Glacier Deep Archive

- Moving object can be done manually or can be automated using Lifecycle Rules



# S3 – Lifecycle Rules

These rules are made of many things include:

* **Transition Actions** – configure the object to transfer to another storage class
  + Ex: 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
  + Ex: 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** can be created for a **certain prefix** (Ex: s3://mybucket/mp4/\*)
* **Rule** can be created for **certain objects** Tags (ex: Department: Finance)

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

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

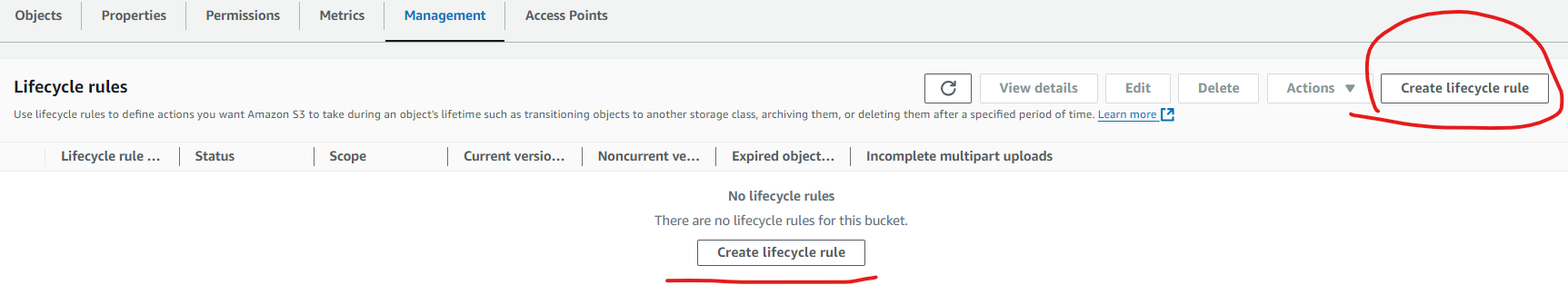
• Transition afterwards the “noncurrent versions” to Glacier Deep Archive

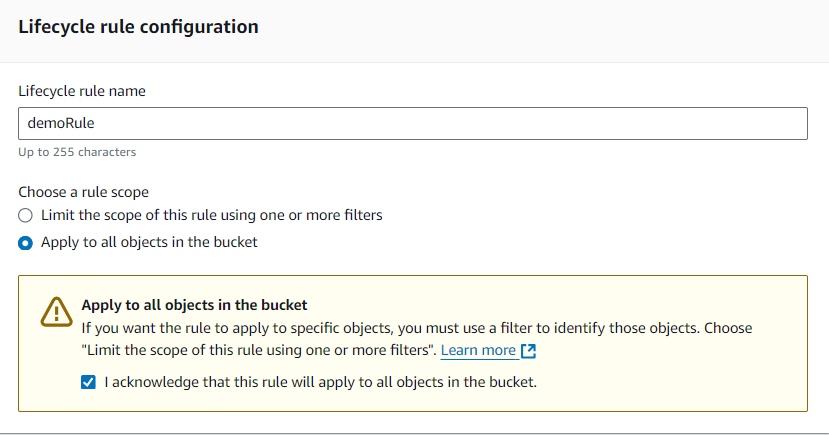
## **S3 Analytics – Storage Class Analysis**

* Help you decide when to transfer objects to the right storage class
* Recommend 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

## **S3 – Lifecycle Rules Hands-on**

In S3 bucket > Management > Create Lifecycle rule





**Life Cycle rule actions**

**Transition current versions of objects between storage classes**

* This action will move current versions.

**Transition noncurrent versions of objects between storage classes**

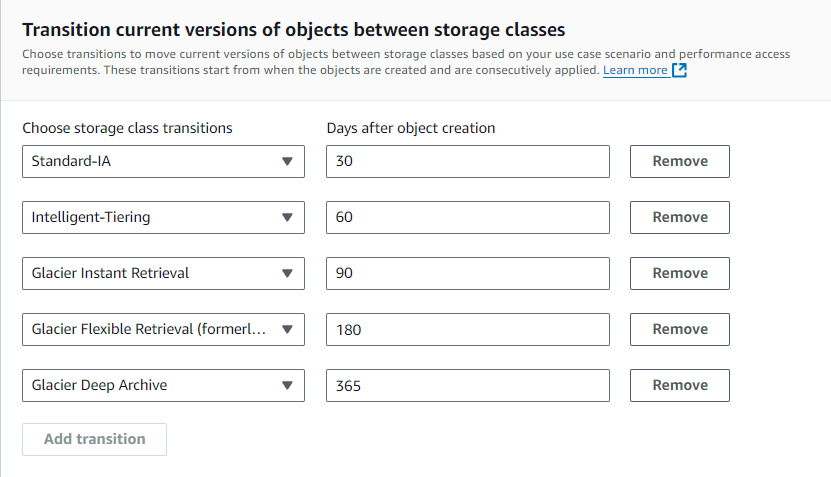
* This action will move noncurrent versions.

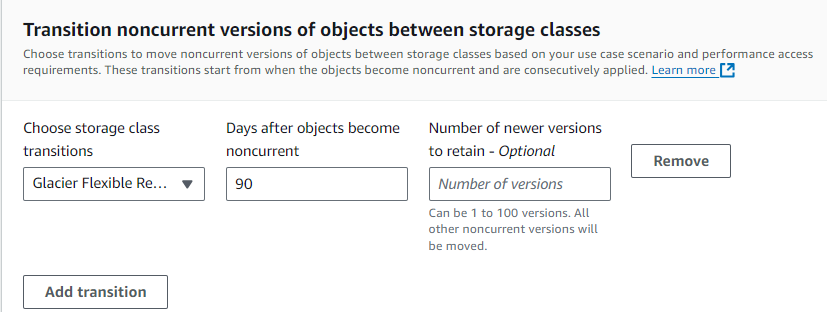
**Expire current versions of objects**

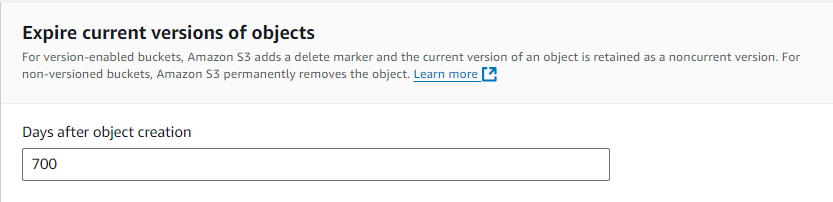
**Permanently delete noncurrent versions of objects**

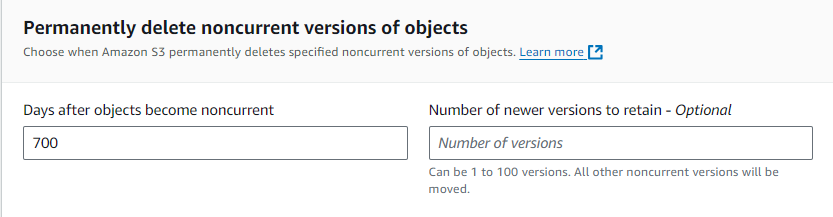
**Delete expired object delete markers or incomplete multipart uploads**

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

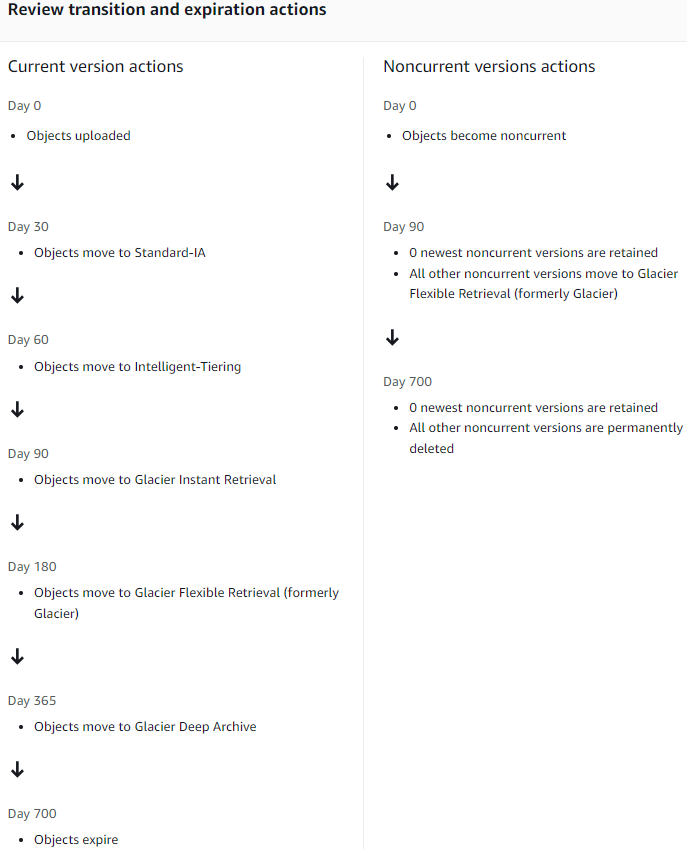








Finally Review



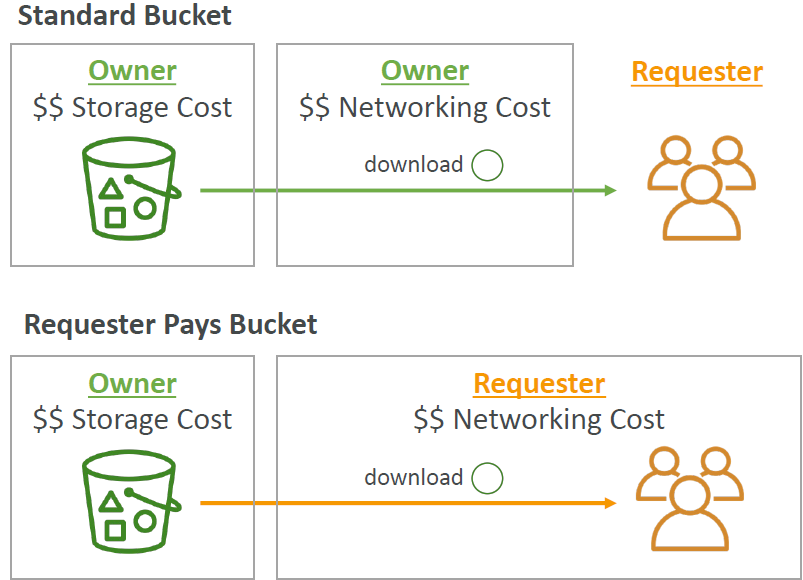
# S3 – Requester Pays

• 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

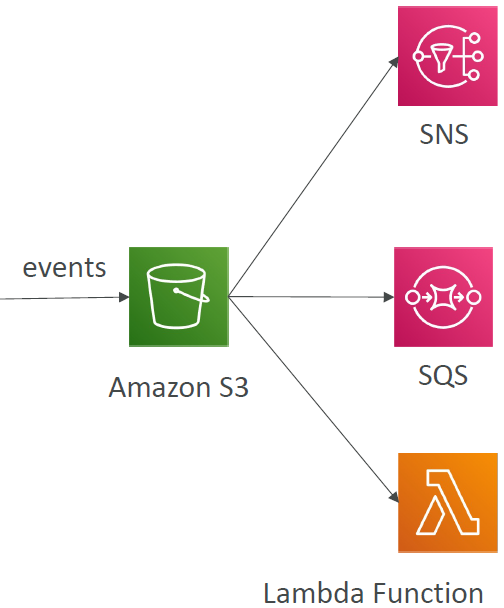
• Helpful when you want to **share large datasets with other accounts**

• The **requester must be authenticated in AWS** (cannot be anonymous)

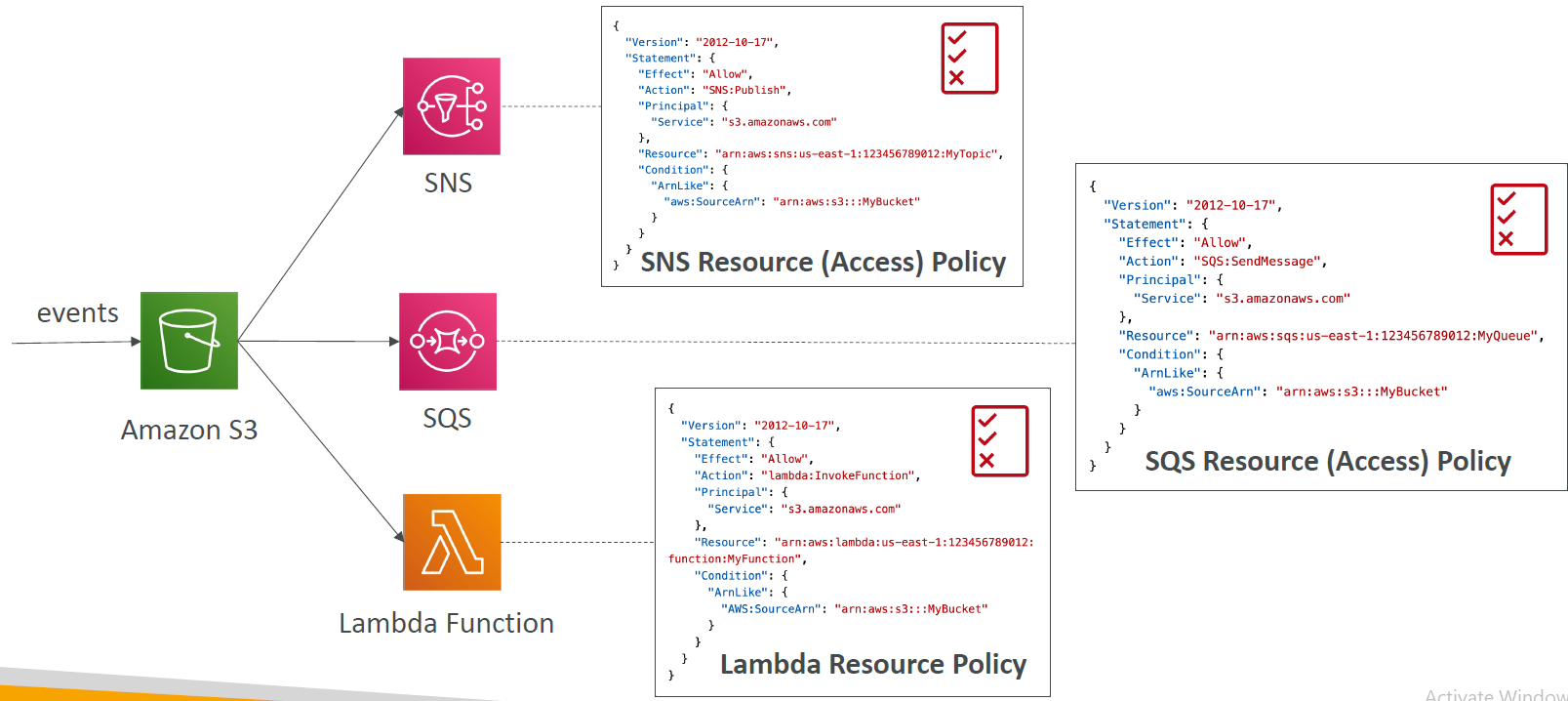


# **S3 Event Notifications**

* 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**

Amazon **EventBridge** **allow S3** bucket to **send notifications over 18 AWS services** as destinations

* **Advance filtering** option with JSON rules (metadata, object size, name,..)
* **Multiple Destinations** – ex: Step Functions, Kinesis Streams/ Firehose…\_
* **EventBridge Capabilities** – Archive, Replay event, Repliable delivery

## **S3 Event Notifications Hands-on**

In S3 bucket > Properties > Scroll down to Create Event Notification in Event notifications tab; with option to enable Amazon EventBridge

Input Event name

Prefix – optional

Suffix – optional

**Event types**

**Object creation**

Select “All object create events”

s3:ObjectCreated:\*

Put

s3:ObjectCreated:Put

Post

s3:ObjectCreated:Post

Copy

s3:ObjectCreated:Copy

Multipart upload completed

**Object removal – optional**

**Object restore – optional**

**Object ACL – optional**

**Object tagging – optional**

**Reduced Redundancy Storage – optional**

**Replication – optional**

**Lifecycle – optional**

**Destination**

Lambda function

Run a Lambda function script based on S3 events.

SNS topic

Fanout messages to systems for parallel processing or directly to people.

Select “SQS queue”

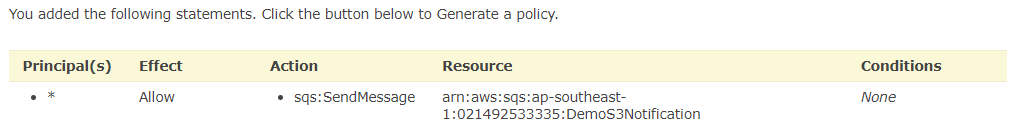
Send notifications to an SQS queue to be read by a server.

Specify SQS queue: Chose SQS queue ARN

Next **create SQS queue** ARN in SQS

Input name for SQS then press create

Access policy > Policy generator



Then copy this JSON policy code paste it back in SQS Access Policy JSON code block

{

"Id": "Policy1731058789370",

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1731058736825",

"Action": [

"sqs:SendMessage"

],

"Effect": "Allow",

"Resource": "arn:aws:sqs:ap-southeast-1:021492533335:DemoS3Notification",

"Principal": "\*"

}

]

}

After that comeback to S3 and Select the SQS queue you just created then press Create.

Test scenario: upload a random file like golang.png to S3

Go To SQS > Send and receive message > Poll message

Result

{"Records":[{"eventVersion":"2.1","eventSource":"aws:s3","awsRegion":"ap-southeast-1","eventTime":"2024-11-08T10:03:10.717Z","eventName":"ObjectCreated:Put","userIdentity":{"principalId":"AWS:AROAQKAIOBBL547A55AR2:dduckhang"},"requestParameters":{"sourceIPAddress":"103.199.7.221"},"responseElements":{"x-amz-request-id":"K2XCX7HKMAQMV043","x-amz-id-2":"sR5fp7v8ZkFcjGd9DIKKi9i/sjnerpS9IoOZBhPOpUEbMUUreH06SSGb7J9r8YGfGZ4NxhiFx9c62JVxCdx/HQWnUjcUacVL"},"s3":{"s3SchemaVersion":"1.0","configurationId":"DemoS3Event","bucket":{"name":"dduckhang-sa","ownerIdentity":{"principalId":"A28OAAU4UVWV30"},"arn":"arn:aws:s3:::dduckhang-sa"},"object":{"key":"golang.png","size":9710,"eTag":"7c5ebc1209d8420d5df47657e074c614","versionId":".kfBA23Th8FXHiPelzyxqc6TZ0VSlAVI","sequencer":"00672DE1DEA05B061F"}}}]}

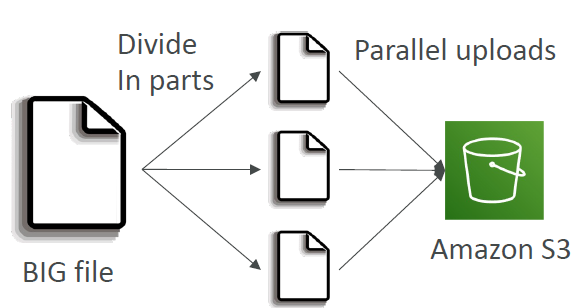
As the SQS message shows: file golang.png size 9710 did exist in S3

# **S3 – Baseline Performance**

* S3 automatically scales to high request rates, latency 100-200 ms
* Your app can achieve at least 3500 PUT/COPY/POST/DELETE or 5500 GET/HEAD requests per second per prefix in a bucket => S3 has really high performance
* There’s no limits to the number of prefixes in a bucket
* Ex: (object path => prefix):
  + bucket/folder1/sub1/file => /folder1/sub1/\*
  + bucket/1/file => 1
* if you spread reads across all 4 prefixes evenly you can achieve 22000 requests per second for GET and HEAD

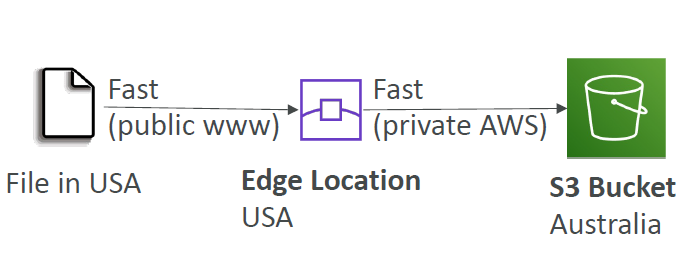
## **S3 Performance**

* Multi-part upload:
  + Recommended for file > 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 data to S3 bucket in target region
* Compatible with multi-part upload

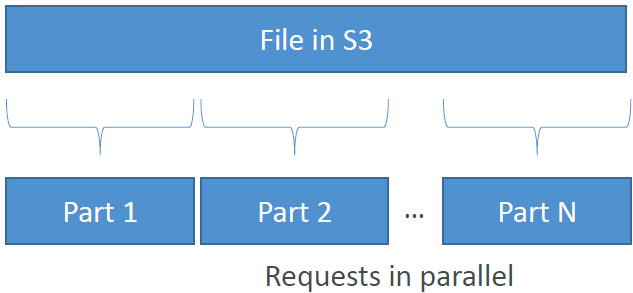


## **S3 Performance – S3 Byte-range Fetches**

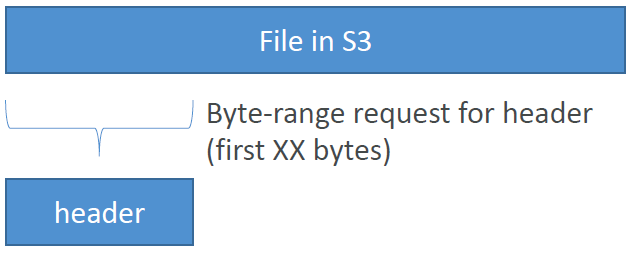
• Parallelize GETs by requesting specific byte ranges

• 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 the file)**

****

# **S3 Batch Operation**

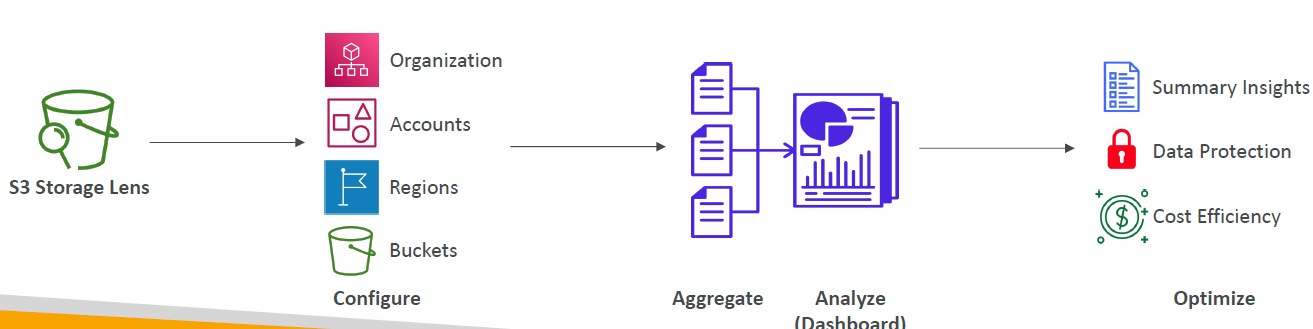
Bulk : mass, huge, large scale

Bulk operation = massive operations at the same time

* Allow perform bulk operations on existing S3 objects with a single request.
* Ex:
  + Modify object metadata & properties
  + Copy object between S3 buckets
  + Encrypt un-encrypted object
  + 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 used for manages retries, track progress, send completion notifications, generate reports….**
* **You can use S3 Inventory to get object list and use S3 Select to filter objects**

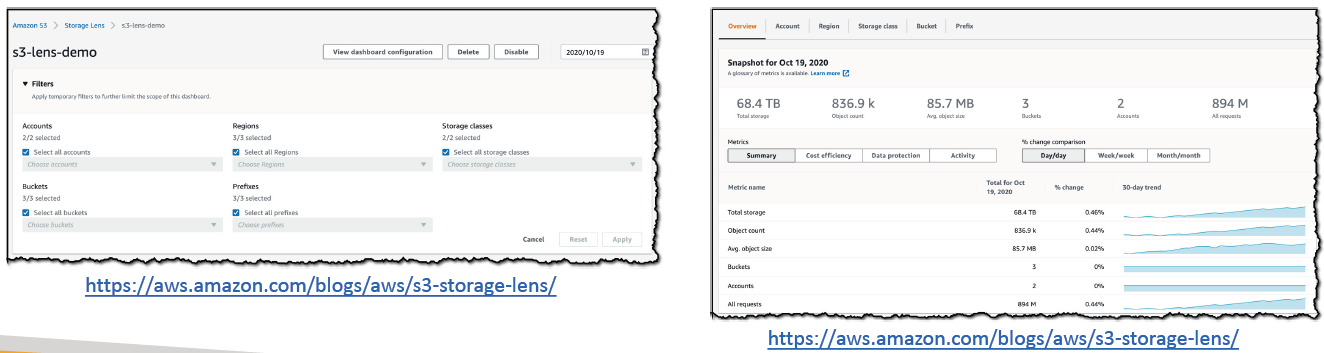
# **S3 – Storage Lens**

* Help understand, analyze and optimize storage across entire AWS Organisation
* **Discover anomalies, identify cost efficiencies and apply data protection best practice across entire AWS Organization** (30 days usage & activity metrics)
* **Aggregate (summary) 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**)



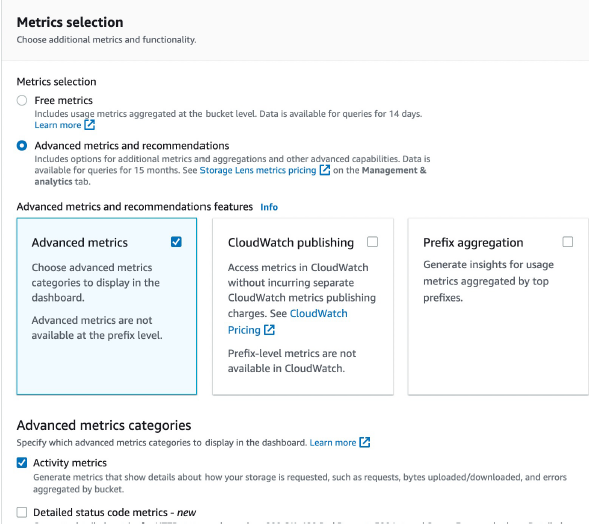
## **Storage Lens – Default Dashboard**

* **Visualize** summarized **insights** and **trends** **for** both **free and advance metrics**
* Default dashboard show Multi-region and Multi-account data
* Preconfigured by S3
* Can not be deleted but can be disabled



## Storage Lens – Metrics

* **Summary metrics**:
  + General insights of your S3 storage
  + StorageBytes, ObjectCOunt,...
  + **Use cases: identify fastest-growing or not used buckets and prefixes**
* **Cost-optimization metric**:
  + Provide insights to manage and optimize your storage expenditure
  + NonCurrentVersionStorageBytes, IncompleteMultipartUploadStorageBytes…..
  + Use cases: identify buckets with incomplete multipart upload operation older than 7 days, point out which objects could be transferred to lower-cost storage class
* Data-protection metric:
  + Provide insights for data protection features
  + VersioningEnabledBucketCount, MFADeleteEnabledBucketCount, SSEKMSEnabledBucketCount, CrossRegionReplicationRuleCount…
  + **Use cases: identify buckets that aren’t following data-protection best pratices**
* Access-management metrics:
  + Provide insight for S3 Object ownership
  + ObjectOwnershipBucketOwnerEnforcedBucketCount…..
  + Use cases: identify Object Ownership settings your bucket use
* Event metrics:
  + Provide insight for S3 Event Notifications
  + EventNotificationEnabledBucketCount (identify which buckets have S3 Event Notifications configured)
* Performance metrics:
  + Provide insights for S3 Transfer Acceleration
  + TransferAccelerationEnabledBucketCount (identify which buckets have S3 TransferAcceleration enabled)
* Activity metrics:
  + Provide observation about how your storage is requested
  + AllRequests, GetRequests, PutRequests, ListRequests, BytesDownloaded…
* Detailed status code metrics:
  + Provide vision for HTTP status codes
  + 200OKStatusCount, 403ForbiddenErrorCount, 404NotFoundErrorCount…
* Free metrics:
  + Automatically available to all customers
  + Contains around 28 usage metrics
  + Data is available for queries for 14 days
* Advanced metrics and recommendations
  + Additional paid metrics and features
  + Advance metrics include: 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



# **QA**

Question 1:

How can you be notified when there's an object uploaded to your S3 bucket?

Answer: S3 Event Notifications since it can send notification through your email or devices

Question 2:

You have an S3 bucket that has S3 Versioning enabled. This S3 bucket has a lot of objects, and you would like to remove old object versions to reduce costs. What's the best approach to automate the deletion of these old object versions?

Answer: S3 Lifecycle Rules – Expiration Actions

Explain: since this can automatically remove your objects after predefined period

Question 3:

How can you automate the transition of S3 objects between their different tiers?

Answer: By using S3 Lifecycle Rules

Question 4:

While you're uploading large files to an S3 bucket using Multi-part Upload, there are a lot of unfinished parts stored in the S3 bucket due to network issues. You are not using these unfinished parts and they cost you money. What is the best approach to remove these unfinished parts?

Answer: Use an S3 Lifecycle Policy to automate delete old/ unfinished parts

Explain: in Life CycleRule Actions select “Delete expired object delete markers or incomplete multipart uploads”

Question 5:

You are looking to get recommendations for S3 Lifecycle Rules. How can you analyze the optimal number of days to move objects between different storage tiers?

Answer: Use S3 Analytics to help give you recommendation on how to set optimal LifeCycle Rule

Question 6:

You are looking to build an index of your files in S3, using Amazon RDS PostgreSQL. To build this index, it is necessary to read the first 250 bytes of each object in S3, which contains some metadata about the content of the file itself. There are over 100,000 files in your S3 bucket, amounting to 50 TB of data. How can you build this index efficiently?

Answer: Create an app that will traverse (go through) the S3 bucket, issue a Byte Range Fetch for the first 250 bytes, and store that information in RDS

Explain: Byte Range Fetch in S3 Performance especially useful for speed up download or download the first xx size header of the file to save the file metadata in this case “, it is necessary to read the first 250 bytes of each object in S3, which contains some metadata about the content of the file itself” so it is a great choice for using Byte Range Fetch

Question 7:

You have a large dataset stored on-premises that you want to upload to the S3 bucket. The dataset is divided into 10 GB files. You have good bandwidth but your Internet connection isn't stable. What is the best way to upload this dataset to S3 and ensure that the process is fast and avoid any problems with the Internet connection?

Answer: Combine both S3 Multi-part Upload & S3 Transfer Acceleration

Explain: Multi-part upload help uploading your files in parallel which increase upload speed and S3 Transfer Acceleration help Increase transfer speed by transferring file to an AWS edge location which will forward data to S3 bucket in target region

Question 8:

A company is preparing for compliance and regulatory review on its infrastructure on AWS. Currently, they have their files stored on S3 buckets encrypted using S3 Default Encryption, which must be encrypted using KMS as required for compliance and regulatory review. Which S3 feature allows them to encrypt all files in their S3 buckets in the most efficient and cost-effective way?

Answer: Use S3 Batch Operation

Explain: S3 Batch Operation allow encrypt and un-encrypt files and it use for massive operation on existing S3 objects in a single request.

Question 9:

You have a 25 GB file that you're trying to upload to S3 but you're getting errors. What is a possible solution for this?

Answer: Use Multi-Part upload when uploading files larger than 5G

Note: Multi-part Upload is recommended as soon as the file is over 100 MB