Object Storage -Cloud Storage

- We need to login to google console and in search bar search cloud storage. And select it.
- There we have a option called **create**. Click on it.
- Give name for bucket. Bucket name is unique globally.
- Then click on **continue**.
- In choose **where to store your data**. Select Region, Select **continue**.
- In choose a **storage class for your data**. Select **Standard.** Select **create**.
- In **course download folder**, we have folder **cloud storage** just drag and drop all the folders in to newly created bucket.

Cloud Storage

* Most popular, very flexible & inexpensive storage service.

* Serverless: Autoscaling and infinite scale.

* Store large objects using a key-value approach:

* Treats entire object as a unit (Partial updates not allowed).

* Recommended when you operate on entire object most of the time.

* Access Control at Object level.

* Also called Object Storage.

* Provides REST API to access and modify objects.

* Also provides CLI (gsutil) & Client Libraries (C++, C#, Java, Node.js, PHP, Python & Ruby)

* Store all file types - text, binary, backup & archives:

* Media files and archives, Application packages and logs.

* Backups of your databases or storage devices.

* Staging data during on-premise to cloud database migration.

Cloud Storage - Objects and Buckets

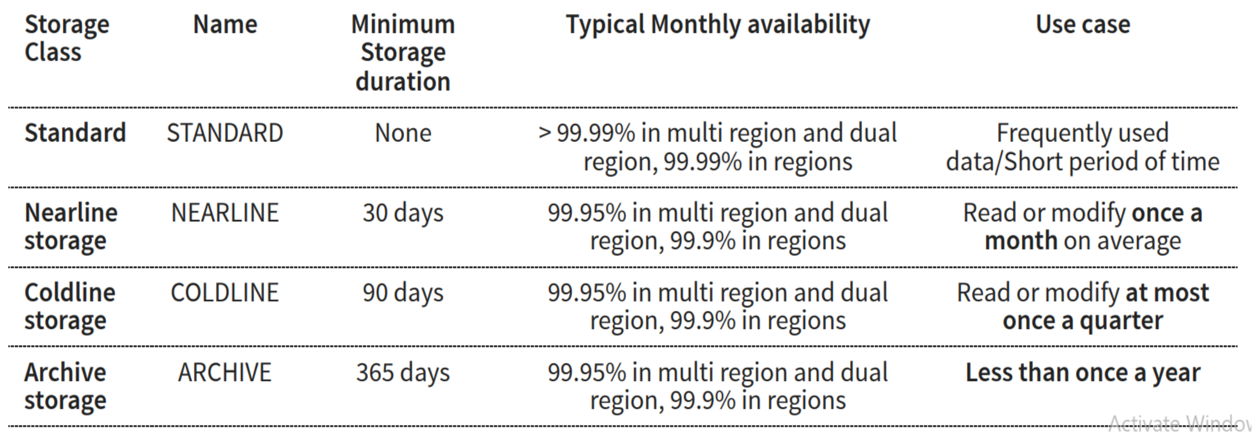
- Objects are stored in buckets.
- Bucket names are globally unique.
- Bucket names are used as part of object URLs => Can contain ONLY lower case letters, numbers, hyphens, underscores and periods.
- 3-63 characters max. Can't start with goog prefix or should not contain google (even misspelled) .
- Unlimited objects in a bucket.
- Each bucket is associated with a project.
- Each object is identified by a unique key.
- Key is unique in a bucket .
- Max object size is 5 TB.
- BUT you can store unlimited number of such objects

Cloud Storage - Storage Classes – Introduction

- Different kinds of data can be stored in Cloud Storage
- Media files and archives.
- Application packages and logs.
- Backups of your databases or storage devices.

- Long term archives.
- Huge variations in access patterns.
- Can I pay a cheaper price for objects I access less frequently?
- Storage classes help to optimize your costs based on your access needs.
- Designed for durability of 99.999999999%(11 9’s)

Cloud Storage - Storage Classes – Comparison



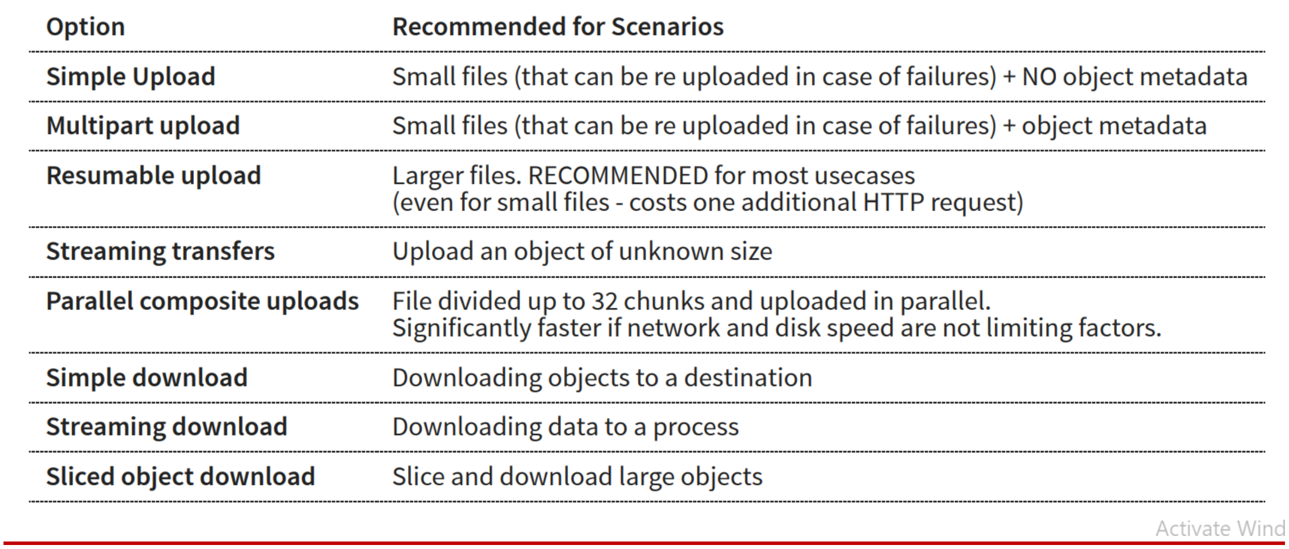
Storage Class	Name	Minimum Storage duration	Typical Monthly availability	Use case
Standard	STANDARD	None	> 99.99% in multi region and dual region, 99.99% in regions	Frequently used data/Short period of time
Nearline storage	NEARLINE	30 days	99.95% in multi region and dual region, 99.9% in regions	Read or modify **once a month** on average
Coldline storage	COLDLINE	90 days	99.95% in multi region and dual region, 99.9% in regions	Read or modify **at most once a quarter**
Archive storage	ARCHIVE	365 days	99.95% in multi region and dual region, 99.9% in regions	**Less than once a year**

Features across Storage Classes

- High durability (99.999999999% annual durability)
- Low latency (first byte typically in tens of milliseconds)
- Unlimited storage .

- Autoscaling (No configuration needed).
- NO minimum object size.
- Same APIs across storage classes.
- Committed SLA is 99.95% for multi region and 99.9% for single region for Standard, Nearline and Coldline storage classes.
- No committed SLA for Archive storage.

Cloud Storage - Uploading and Downloading Objects



Option	Recommended for Scenarios
Simple Upload	Small files (that can be re uploaded in case of failures) + NO object metadata
Multipart upload	Small files (that can be re uploaded in case of failures) + object metadata
Resumable upload	Larger files. RECOMMENDED for most usecases (even for small files - costs one additional HTTP request)
Streaming transfers	Upload an object of unknown size
Parallel composite uploads	File divided up to 32 chunks and uploaded in parallel. Significantly faster if network and disk speed are not limiting factors.
Simple download	Downloading objects to a destination
Streaming download	Downloading data to a process
Sliced object download	Slice and download large objects

Object Versioning

- Prevents accidental deletion & provides history.
- Enabled at bucket level.

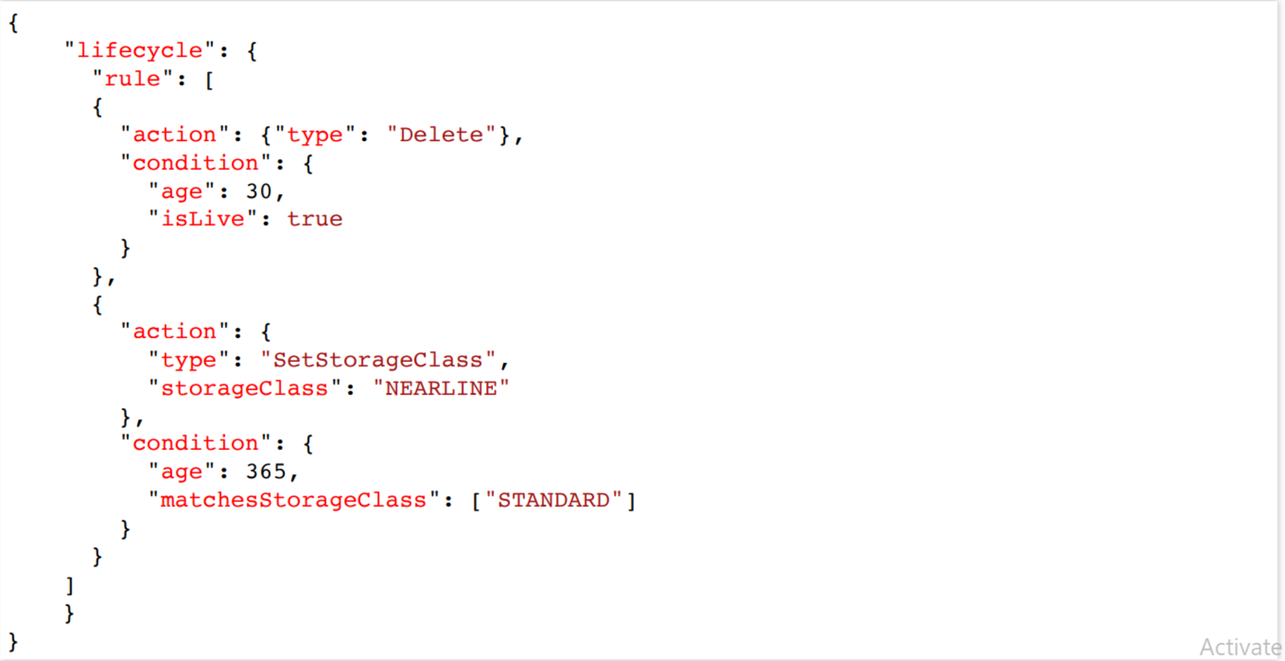
- Can be turned on/off at any time.
- Live version is the latest version.
- If you delete live object, it becomes noncurrent object version.
- If you delete noncurrent object version, it is deleted.
- Older versions are uniquely identified by (object key + a generation number).
- Reduce costs by deleting older (noncurrent) versions!

Object Lifecycle Management

- Files are frequently accessed when they are created.
- Generally usage reduces with time.
- How do you save costs by moving files automatically between storage classes?
- Solution: Object Lifecycle Management.
- Identify objects using conditions based on:
- Age, CreatedBefore, IsLive, MatchesStorageClass, NumberOfNewerVersions etc.
- Set multiple conditions: all conditions must be satisfied for action to happen.
- Two kinds of actions:

- SetStorageClass actions (change from one storage class to another) .
- Deletion actions (delete objects)
- Allowed Transitions:
- (Standard or Multi-Regional or Regional) to (Nearline or Coldline or Archive)
- Nearline to (Coldline or Archive)
- Coldline to Archive

Object Lifecycle Management - Example Rule

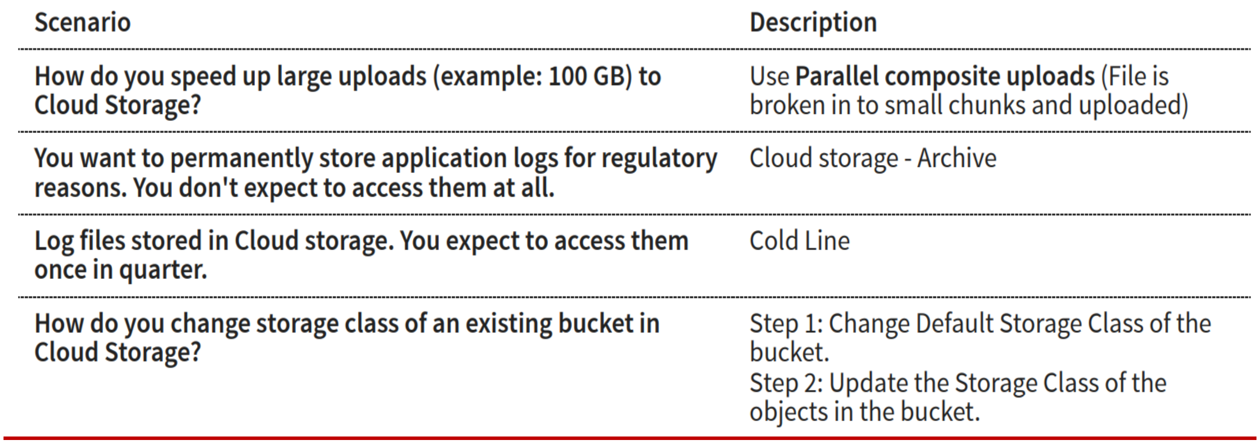


```
{
    "lifecycle": {
      "rule": [
      {
        "action": {"type": "Delete"},
        "condition": {
          "age": 30,
          "isLive": true
        }
      },
      {
        "action": {
          "type": "SetStorageClass",
          "storageClass": "NEARLINE"
        },
        "condition": {
          "age": 365,
          "matchesStorageClass": ["STANDARD"]
        }
      }
    ]
    }
}
```

Cloud Storage – Encryption

- Cloud Storage always encrypts data on the server side!
- Configure Server-side encryption: Encryption performed by Cloud Storage.
- Google-managed encryption key - Default (No configuration required).
- Customer-managed encryption keys - Created using Cloud Key Management Service (KMS). Managed by customer in KMS.
- Cloud Storage Service Account should have access to keys in KMS for encrypting and decrypting using the Customer-Managed encryption key .
- (OPTIONAL) Client-side encryption - Encryption performed by customer before upload.
- GCP does NOT know about the keys used.

Cloud Storage – Scenarios



Scenario	Description
How do you speed up large uploads (example: 100 GB) to Cloud Storage?	Use **Parallel composite uploads** (File is broken in to small chunks and uploaded)
You want to permanently store application logs for regulatory reasons. You don't expect to access them at all.	Cloud storage - Archive
Log files stored in Cloud storage. You expect to access them once in quarter.	Cold Line
How do you change storage class of an existing bucket in Cloud Storage?	Step 1: Change Default Storage Class of the bucket. Step 2: Update the Storage Class of the objects in the bucket.

Cloud Storage - Command Line - gsutil – 1

- (REMEMBER) gsutil is the CLI for Cloud Storage (NOT gcloud) .
- Cloud Storage (gsutil) .
- gsutil mb gs://BKT_NAME (Create Cloud Storage bucket)
- gsutil ls -a gs://BKT_NAME (List current and non-current object versions)
- gsutil cp gs://SRC_BKT/SRC_OBJ gs://DESTN_BKT/NAME_COPY (Copy objects)
- -o 'GSUtil:encryption_key=ENCRYPTION_KEY' - Encrypt Object
- gsutil mv (Rename/Move objects)

- gsutil mv gs://BKT_NAME/OLD_OBJ_NAME gs://BKT_NAME/NEW_OBJ_NAME
- gsutil mv gs://OLD_BUCKET_NAME/OLD_OBJECT_NAME gs://NEW_BKT_NAME/NEW_OBJ_NAME
- gsutil rewrite -s STORAGE_CLASS gs://BKT_NAME/OBJ_PATH (Ex: Change Storage Class for objects)
- gsutil cp : Upload and Download Objects
- gsutil cp LOCAL_LOCATION gs://DESTINATION_BKT_NAME/ (Upload)
- gsutil cp gs://BKT_NAME/OBJ_PATH LOCAL_LOCATION (Download)

Cloud Storage - Command Line - gsutil – 2

- Cloud Storage (gsutil)
- gsutil versioning set on/off gs://BKT_NAME (Enable/Disable Versioning)
- gsutil uniformbucketlevelaccess set on/off gs://BKT_NAME
- gsutil acl ch (Set Access Permissions for Specific Objects)
- gsutil acl ch -u AllUsers:R gs://BKT_NAME/OBJ_PATH (Make specific object public)

- gsutil acl ch -u john.doe@example.com:WRITE gs://BKT_NAME/OBJ_PATH
- Permissions - READ (R), WRITE (W), OWNER (O)
- Scope - User, allAuthenticatedUsers, allUsers(-u), Group (-g), Project (-p) etc
- gsutil acl set JSON_FILE gs://BKT_NAME
- gsutil iam ch MBR_TYPE:MBR_NAME:IAM_ROLE gs://BKT_NAME (Setup IAM role)
- gsutil iam ch user:me@myemail.com:objectCreator gs://BKT_NAME gsutil iam ch allUsers:objectViewer gs://BKT_NAME (make the entire bucket readable)
- gsutil signurl -d 10m YOUR_KEY gs://BUCKET_NAME/OBJECT_PATH (Signed URL for temporary access).

Cloud Storage - Command Line - gcloud storage

- Earlier, gsutil was the recommended CLI for Cloud Storage.
- GCLOUD STORAGE is now the recommended CLI for Cloud Storage
- Advantages:
- Upto 94% faster storage transfers
- Better parallel processing

- Do NOT worry about options/parameters/flags
- gcloud storage will decide the optimal storage transfer approach for you
- Provides very simple to remember commands (consistent with gcloud):
- gcloud storage buckets create gs://BKT_NAME (Create Cloud Storage bucket)
- options: --default-encryption-key, --default-storage-class
- gcloud storage buckets delete gs://BKT_NAME
- gcloud storage buckets list gs://B*
- gcloud storage buckets describe gs://BKT_NAME
- gcloud storage buckets update gs://BKT_NAME
- options: --default-encryption-key, --default-storage-class, --[no-]versioning
- If you have existing scripts that make use of gsutil commands AND
- You do NOT want to change the scripts AND
- You want the performance benefits offered by new features in gcloud storage.
- Check out shim (In boto configuration file, configure use_gcloud_storage=True under GSUtil section)