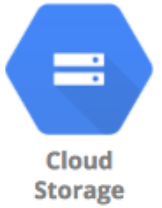
Object Storage - Cloud Storage

Cloud Storage

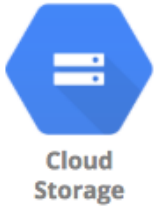
- **Most popular, very flexible & inexpensive** storage service
  - Serverless: Autoscaling and infinite scale
- Store large objects using a **key-value** approach:
  - Treats entire object as a unit (Partial updates not allowed)
    - Recommended when you operate on entire object most of the time
    - 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

In 28 Minutes

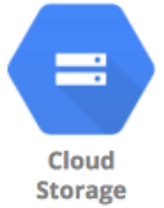
- Objects are stored in buckets
  - Bucket names are **globally unique**
  - Bucket names are used as part of object URLs => Can contain ONLY lower case letters, numbers, hyphens, underscores and periods.
  - 3-63 characters max. Can't start with **goog prefix** or should not contain **google (even misspelled)**
  - 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

In 28 Minutes

- **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

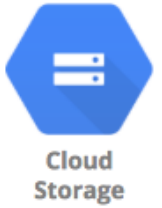
In28 Minutes

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

In 28 Minutes

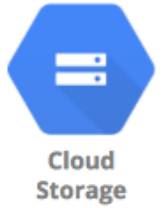
- High durability (99.999999999% annual durability)
- **Low** latency (first byte typically in tens of milliseconds)
- **Unlimited** storage
  - Autoscaling (No configuration needed)
  - **NO minimum** object size
- Same APIs across storage classes
- **Committed SLA** is 99.95% for multi region and 99.9% for single region for Standard, Nearline and Coldline storage classes
  - No committed SLA for Archive storage



Object Versioning

In 28 Minutes

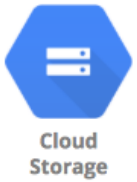
- Prevents **accidental deletion** & provides history
  - Enabled at bucket level
    - Can be turned on/off at any time
  - **Live version** is the latest version
    - If you delete live object, it becomes noncurrent object version
    - If you delete noncurrent object version, it is deleted
  - Older versions are uniquely identified by (object key + a generation number)
  - Reduce costs by deleting older (noncurrent) versions!



Object Lifecycle Management

In28 Minutes

- 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

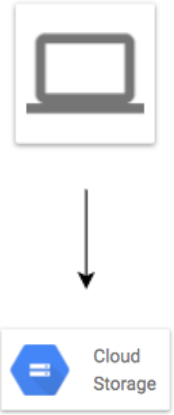
In28 Minutes

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

Cloud Storage - Encryption

In28 Minutes

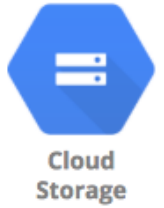
- (Default) Cloud Storage encrypts data on the server side!
- **Server-side** encryption: Performed by GCS after it receives data
  - **Google-managed** - Default (No configuration needed)
  - **Customer-managed** - Keys managed by customer in **Cloud KMS**:
    - GCS Service Account should have access to customer-managed keys in KMS to be able to encrypt and decrypt files
  - **Customer-supplied** - Customer supplies the keys with every GCS operation
    - Cloud Storage does NOT store the key
    - Customer is responsible for storing and using it when making API calls
      - Use API headers when making API calls
        - x-goog-encryption-algorithm, x-goog-encryption-key (Base 64 encryption key), x-goog-encryption-key-sha256 (encryption key hash)
      - OR when using gsutil: In boto configuration file, configure encryption_key under GSUtil section
- (OPTIONAL) **Client-side** encryption - Encryption performed by customer before upload
  - GCP does NOT know about the keys used
  - GCP is NOT involved in encryption or decryption



Understanding Cloud Storage Metadata

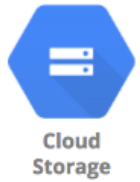
In 28 Minutes

- Each object in Cloud Storage can have **Metadata** associated with it
  - Key Value Pairs ex: `storageClass:STANDARD`
    - Storage class of an object is represented by metadata
  - **Fixed-key metadata**: Fixed key - Changing value
    - `Cache-Control: public, max-age=3600` (Is caching allowed? If so, for how long?)
    - `Content-Disposition: attachment; filename="myfile.pdf"` (Should content be displayed inline in the browser or should it be an attachment, which can be downloaded)
    - `Content-Type: application/pdf` (What kind of content does the object have?)
    - etc..
  - **Custom metadata**: You can define your own keys and values
  - **Non-editable metadata**: You cannot edit these directly
    - Storage class of the object, customer-managed encryption keys etc



Cloud Storage Bucket Lock - Meet Compliance Needs

In 28 Minutes



- How do you ensure that you **comply with regulatory and compliance requirements** around immutable storage in a Cloud Storage bucket?
- Configure **data retention policy** with retention period:
  - How long should objects in the bucket be retained for?
    - "Objects in the bucket can only be deleted or replaced once their age is greater than the retention period"
  - You can set it **while creating a bucket or at a later point in time**
    - Applies automatically to existing objects in the bucket (as well as new objects added in)
  - Once a retention policy is locked:
    - You **CANNOT remove retention policy** or **reduce** retention period (You can increase retention period)
    - You **CANNOT delete the bucket** unless all objects in bucket have age greater than retention period
  - Retention policies and Object Versioning are mutually exclusive features

Transferring data from on premises to cloud

In 28 Minutes

- **Most popular** data destination is **Google Cloud Storage**
- **Options:**
  - **Online Transfer**: Use gsutil or API to transfer data to Google Cloud Storage
    - Good for one time transfers
  - **Storage Transfer Service**: Recommended for large-scale (petabytes) online data transfers from your private data centers, AWS, Azure, and Google Cloud
    - You can set up a repeating schedule
    - Supports incremental transfer (only transfer changed objects)
    - Reliable and fault tolerant - continues from where it left off in case of errors
  - **Storage Transfer Service vs gsutil**:
    - gsutil is recommended only when you are transferring less than 1 TB from on-premises or another GCS bucket
    - Storage Transfer Service is recommended if either of the conditions is met:
      - Transferring more than 1 TB from anywhere
      - Transferring from another cloud
  - **Transfer Appliance**: Physical transfer using an appliance

Migrating Data with Transfer Appliance

In 28 Minutes

- **Transfer Appliance**: Copy, ship and upload data to GCS
  - **Recommended** if your data size is **greater than 20TB**
    - OR **online transfer takes > 1 week**
  - **Process**:
    - Request an appliance
    - Upload your data
    - Ship the appliance back
    - Google uploads the data
  - Fast copy (upto 40Gbps)
  - AES 256 encryption - Customer-managed encryption keys
  - Order **multiple devices** (TA40, TA300) if need

	Physical Transfer		Physical / Online Transfer			Online Transfer
	1 Mbps	10 Mbps	100 Mbps	1 Gbps	10 Gbps	100 Gbps
1 GB	3 hours	18 minutes	2 minutes	11 seconds	1 second	0.1 seconds
10 GB	30 hours	3 hours	18 minutes	2 minutes	11 seconds	1 second
100 GB	12 days	30 hours	3 hours	18 minutes	2 minutes	11 seconds
1 TB	124 days	12 days	30 hours	3 hours	18 minutes	2 minutes
10 TB	3 years	124 days	12 days	30 hours	3 hours	18 minutes
100 TB	34 years	3 years	124 days	12 days	30 hours	3 hours
1 PB	340 years	34 years	3 years	124 days	12 days	30 hours
10 PB	3,404 years	340 years	34 years	3 years	124 days	12 days
100 PB	34,048 years	3,404 years	340 years	34 years	3 years	124 days

https://cloud.google.com

Understanding Cloud Storage Best Practices

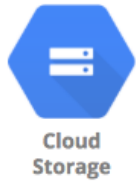
In 28 Minutes

- Avoid use of sensitive info in bucket or object names
- Store data in the **closest region** (to your users)
- Ramp up **request rate gradually**
  - No problems upto 1000 write requests per second or 5000 read requests per second
  - BUT beyond that, take at least 20 minutes to double request rates
- Use **Exponential backoff** if you receive 5xx (server error) or 429 (too many requests) errors
  - Retry after 1, 2, 4, 8, 16, .. seconds
- **Do NOT use sequential** numbers or timestamp as object keys
  - Recommended to use completely random object names
  - Recommended to add a hash value before the sequence number or timestamp
- **Use Cloud Storage FUSE** to enable file system access to Cloud Storage
  - Mount Cloud Storage buckets as file systems on Linux or macOS systems

Cloud Storage - Command Line - gsutil - 1

In28 Minutes

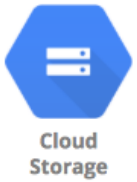
- (REMEMBER) **gsutil** is the CLI for Cloud Storage (**NOT gcloud**)
- Cloud Storage (**gsutil**)
  - *gsutil **mb** gs://BKT_NAME* (Create Cloud Storage bucket)
  - *gsutil **ls -a** gs://BKT_NAME* (List current and non-current object versions)
  - *gsutil **cp** gs://SRC_BKT/SRC_OBJ gs://DESTN_BKT/NAME_COPY* (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

In28 Minutes

- 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 - Scenarios

In28 Minutes

Scenario	Solution
I will frequently access objects in a bucket for 30 days. After that I don't expect to access objects at all. We have compliance requirements to store objects for 4 years.How can I minimize my costs?	Initial Storage Class - Standard Lifecycle policy: Move to Archive class after 30 days. Delete after 4 years.
I want to transfer 2 TB of data from Azure Storage to Cloud Storage	Use Cloud Storage Transfer Service
I want to transfer 40 TB of data from on premises to Cloud Storage	Use Transfer Appliance
Customer wants to manage their Keys	**Customer-managed** - Keys managed by customer in **Cloud KMS**
Regulatory compliance: Object should not modified for 2 years	Configure and lock **data retention policy**