Storage Options in Google Cloud Platform

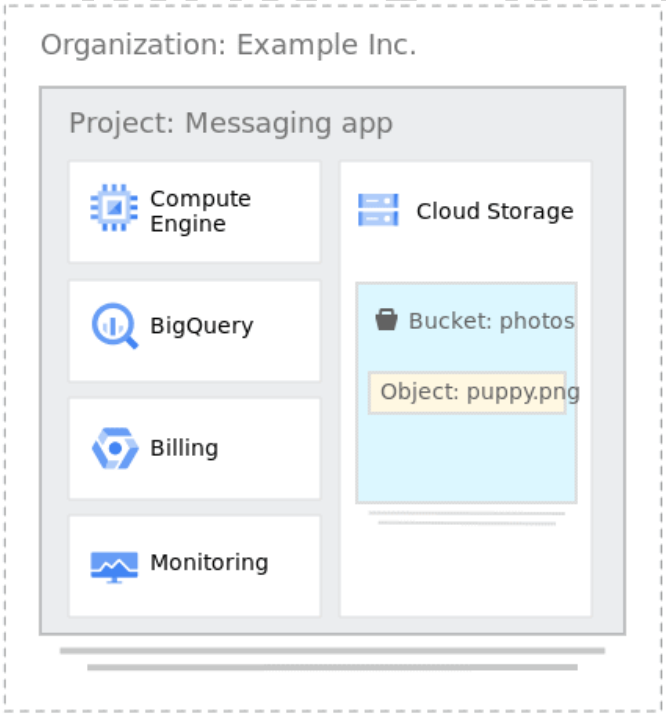
Google Cloud Storage:

- Cloud Storage allows world-wide storage and retrieval of any amount of data at any time.
- It is an Infrastructure as a Service (IaaS), comparable to Amazon S3 online storage service.
- You can use Cloud Storage for a range of scenarios including
  - serving website content - static
  - storing data for archival and disaster recovery
  - distributing large data objects to users via direct download.
- Cloud Storage is a service for storing objects in Google Cloud.
- An object is an immutable piece of data consisting of a file of any format. You store objects in containers called buckets.
- All buckets are associated with a project, and you can group your projects under an organization.

- Google Cloud Storage is a RESTful online file storage web service for storing and accessing data on Google Cloud Platform infrastructure.
- Cloud Storage is unified object storage service.
- Cloud Storage is a persistent storage, it is durable, replicated and also made globally available via HTTP URL.
- Cloud Storage is auto scalable service.
- Cloud Storage is not a File System, because each item in cloud storage have unique URL.

Google Cloud Storage Hierarchy:

Cloud Storage structure can apply to a real-world case:



•Organization:

oYour company, called Cloud & AI Analytics Inc., creates a Google Cloud organization called cloudaIanalytics.org

•Project:

oCloud & AI Analytics Inc. is building several applications, and each one is associated with a project.

oEach project has its own set of Cloud Storage APIs, as well as other resources.

•Buckets:

oBasic containers that hold your data. Everything that you store in Google Cloud Storage must be contained in a bucket. You can use buckets to organize your data and control access to your data, but unlike directories and folders, you cannot nest buckets.

- o When you create a bucket, you give it a globally-unique name and a geographic location where the bucket and its contents are stored.
- o You cannot change the name or location of an existing bucket. Instead, you can create a new bucket with the desired name or in

the desired location and move the contents from the old bucket to the new bucket.

- There is no limit to the number of buckets you can have in a project or location.
- The maximum size of a single upload request is also 5 TiB.

- **Bucket Names:**
  - Should be unique as the name of the buckets stored in single Cloud Storage namespace.
  - Bucket names can be used with a CNAME redirect, which means they need to conform to DNS naming conventions.
  - Every bucket name must be globally unique.
  - Bucket names are publicly visible.
  - Once you delete a bucket, anyone can reuse its name for a new bucket.

- **Bucket labels:**
  - Bucket labels are **key:value** metadata pairs that allow you to group your buckets along with other Google Cloud.
- **Objects:**
  - Objects are the individual pieces of data that you store in Google Cloud Storage such as an image called puppy.png.
  - Objects have two components:
    - **Object data**
      - The object data component is usually a file that you want to store in Google Cloud Storage.
    - **Object metadata**
      - The object metadata component is a collection of name-value pairs that describe various object qualities.
- There is no limit on the number of objects that you can create in a bucket.
- Cloud Storage objects are immutable.Cloud Storage allow to version the stored objects.
- Object Versioning needs to be enable explicitly, in absence of Object Versioning, new objects terminates the old.

- Cloud Storage offers life cycle management policy for the objets in bucket.

Basic tools for Cloud Storage

Here are some basic ways you can interact with Cloud Storage:

1. **Console:**
    - The Google Cloud Console provides a visual interface for you to manage your data in a browser.
2. **Gsutil:**
    - gsutil is a command-line tool that allows you to interact with Cloud Storage through a terminal. If you use other Google Cloud services, you can download the Google Cloud CLI, which includes gsutil along with the gcloud tool for other services.
        - https://cloud.google.com/storage/docs/gsutil
3. **Client libraries:**
    - The Cloud Storage client libraries allow you to manage your data using one of your preferred languages, including C++, C#, Go, Java, Node.js, PHP, Python, and Ruby.
        - https://cloud.google.com/storage/docs/reference/libraries
4. **REST APIs:**
    - Manage your data using the JSON or XML API.
        - https://cloud.google.com/storage/docs/json_api
        - https://cloud.google.com/storage/docs/xml-api/overview
5. **Terraform:**
    - Terraform is an infrastructure-as-code (IaC) tool that you can use to provision the infrastructure for Cloud Storage

Features of Cloud Storage:

- Object storage solution in GCP
- Unstructured Data storage
    - Image
    - Video
    - Binary File etc
- Automatic storage class transitions

- Continental-scale and SLA-backed replication - region, Dual region and Multi region
- Fast and flexible transfer services
    - **Storage Transfer Service** offers a highly performant, **online pathway** to Cloud Storage—both with the scalability and speed you need to simplify the data transfer process.
    - For **offline data transfer** our Transfer Appliance is a **shippable storage server** that sits in your datacenter and then ships to an ingest location where the data is uploaded to Cloud Storage.
- Scale to Exabyte
- Unlimited data can be stored
- No minimum Object Size
- Max object size is 5 TB
- High Durability
- 99.999999999% annual **(https://uptime.is/)**
- Object can be Globally access
- Object- and bucket-level permissions - IAM allows you to control who has access to your buckets and objects.
- Global unique name for bucket.Example access URL:
    - https://storage.cloud.google.com/[bucket]/[objectname]
- Bucket level lock with data retention policy
- Objects can be versioned
- Object Life Cycle Management - that trigger data deletion or transition to a cheaper storage class.
- Object holds - Place a hold on an object to prevent its deletion.
- Storage event triggers - Cloud functions (FAAS)
- Pub/Sub notifications for Cloud Storage - Send notifications to Pub/Sub when objects are created, updated, or deleted.

Auto Failover:

- Automatic failover is the process of automatically moving an application to a standby server during a failure or service event to preserve its uptime.

- It can be part of a high-availability approach, or part of a disaster recovery (DR) approach, depending on where the second system is, and how it is used.

Region	Dual-Region	Multi-region
• Lowest latency within a single region • Replicated data across multiple zones in single region	• High availability and low latency across 2 regions (Paired region) • Auto-failover	• Highest availability across continent area – US, EU, Asia • Auto-failover

Cloud Storage Classes

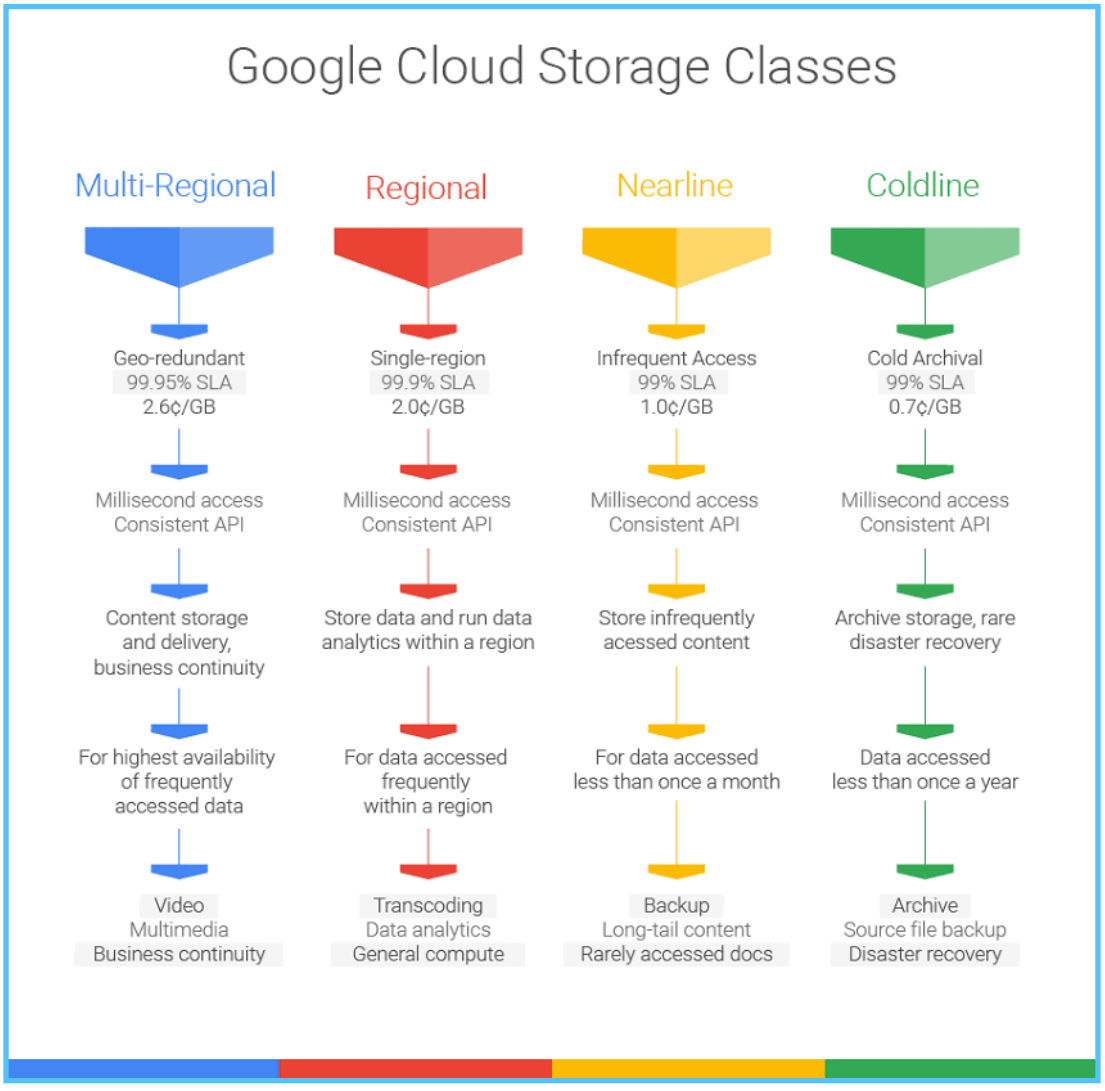
- The following aspects apply to all storage classes:
  - Unlimited storage with unlimited access.
  - No minimum object size and maximum is 5TB.
  - Worldwide accessibility and worldwide storage locations.
  - Low latency (time to first byte typically tens of milliseconds).
  - High durability (99.999999999% annual durability).
  - Redundant across regions if the data is stored in a multi-region or dual-region.
  - A uniform experience with Cloud Storage features, security, tools, and APIs.

Key concepts

- A storage class is a piece of metadata that is used by every object.
- The storage class set for an object affects the object's availability and pricing model.
  - You can change the storage class of an existing object either by rewriting the object or by using Object Lifecycle Management.
  - You can enable the Autoclass feature on a bucket to let Cloud Storage manage storage class transitions for you automatically.
- When you create a bucket, you can specify a default storage class for the bucket. When you add objects to the bucket, they inherit this storage class unless explicitly set otherwise.

- If you don't specify a default storage class when you create a bucket, that bucket's default storage class is set to Standard storage.
- Changing the default storage class of a bucket does not affect any of the objects that already exist in the bucket.

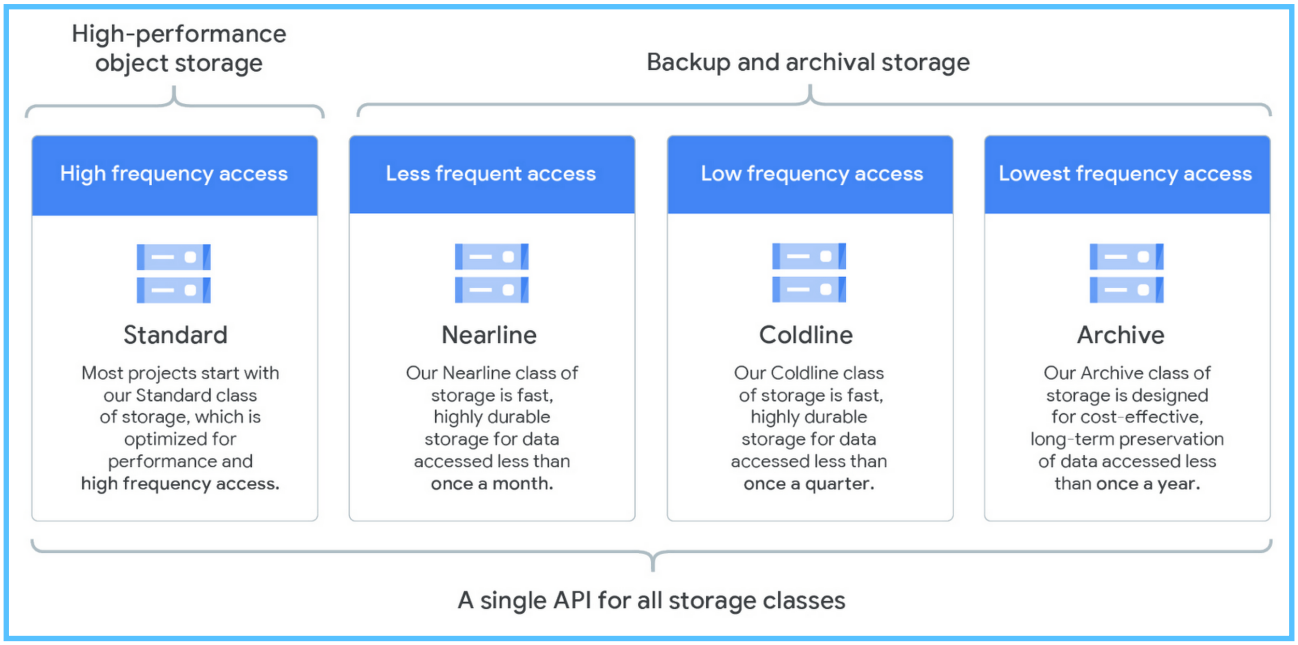
1. **Standard:**
   - Standard storage is best for data that is frequently accessed ("hot" data) and/or stored for only brief periods of time.
   - High frequency access
   - Storage Costliest
   - Access cost is very low
   - Low latency
   - SLA: 99.95% Multi/Dual



- 99.9% Regional

2. **Nearline:**
   - Nearline storage is a low-cost, highly durable storage service for storing infrequently accessed data.
   - Low Frequency access
   - Once in 30 days
   - Access cost will increase
   - Cheaper than standard
   - Back up
   - SLA:
     - 99.9% Multi/Dual
     - 99.0% for Regional

3. **Cold line:**
   - Coldline storage is a very-low-cost, highly durable storage service for storing infrequently accessed data.
   - Very low frequency to access
   - Once in 90 days



   - Cheaper than Nearline
   - SLA:

- 99.9% Multi/Dual
- 99.0% for Regional

4. **Archive:**
   - Archive storage is the lowest-cost, highly durable storage service for data archiving, online backup, and disaster recovery.
   - Cold data storage - Archived data, such as data stored for legal or regulatory reasons, can be stored at low cost as Archive storage, yet still be available if you need it.
   - Disaster recovery - In the event of a disaster recovery event, recovery time is key. Cloud Storage provides low latency access to data stored as Archive storage.
   - Data access is once in year
   - Access cost very high
   - No SLA

Additional classes

- Cloud Storage supports several additional storage classes.
  - **Multi-Regional storage:** Equivalent to Standard storage, except Multi-Regional storage can only be used for objects stored in multi-regions or dual-regions.

Storage Class	Name for APIs and CLIs	Minimum storage duration	Retrieval fees	Typical monthly availability[1]
Standard storage	`STANDARD`	None	None	• >99.99% in multi-regions and dual-regions • 99.99% in regions
Nearline storage	`NEARLINE`	30 days	Yes	• 99.95% in multi-regions and dual-regions • 99.9% in regions
Coldline storage	`COLDLINE`	90 days	Yes	• 99.95% in multi-regions and dual-regions • 99.9% in regions
Archive storage	`ARCHIVE`	365 days	Yes	• 99.95% in multi-regions and dual-regions • 99.9% in regions

  - **Regional storage:** Equivalent to Standard storage, except Regional storage can only be used for objects stored in regions.
  - **Durable Reduced Availability (DRA) storage:** Similar to Standard storage except:

- DRA has higher pricing for operations.
- DRA has lower performance, particularly in terms of availability (DRA has a 99% availability SLA).

Overview on Autoclass - Cloud Storage Bucket

- Autoclass feature automatically transitions objects in your bucket to appropriate storage classes based on each object's access pattern.
- Autoclass simplifies and automates cost saving for your Cloud Storage data. When enabled on a bucket, there are no early deletion charges, no retrieval charges, and no charges for storage class transitions.

Properties of Autoclass:

- When enabled, Autoclass manages all aspects of storage classes for a bucket:
  - All objects added or rewritten to the bucket begin in Standard storage, even if a different storage class is specified in the request.
  - Objects transition to progressively colder storage classes if they're not accessed.
- **Pricing**
  - Cloud Storage pricing remains the same for Autoclass-enabled buckets, with the following exceptions:
    - Retrieval fees are never charged.
    - Early deletion fees are never charged.
    - There are no operation charges when Autoclass transitions an object.
    - An Autoclass Management fee applies when using Autoclass.
- **Note** that operation and networking charges are determined by the storage class of an object at the time it's requested, not the storage class the object subsequently transitions to.
- **Transition behavior**
  - When Autoclass is enabled, objects are transitioned between storage classes as follows:

        - If an object's data is accessed, the object transitions to Standard storage.
        - Any object that isn't accessed for 30 days transitions to Nearline storage.
        - Any object that isn't accessed for 90 days transitions to Coldline storage.
        - Any object that isn't accessed for 365 days transitions to Archive storage.
- **Restrictions**
    - Currently, Autoclass must be enabled at bucket creation time. Existing buckets can only disable the feature, not enable it.
    - A bucket cannot have both Autoclass enabled and use the SetStorageClass action in an Object Lifecycle Management rule. Requests that would cause both to be set on a bucket fail.

Overview on HMAC keys

- Hash-based message authentication code (HMAC) keys, which you can use to authenticate requests to Cloud Storage.
- HMAC keys are useful when you want to move data between other cloud storage providers and Cloud Storage.
- An HMAC key is a type of credential and can be associated with a service account or a user account in Cloud Storage.
- You use an HMAC key to create *signatures* which are then included in requests to Cloud Storage. Signatures show that a given request is authorized by the user or service account.
- After creation, it can take up to 15 seconds for a service account HMAC key to become useable.
- HMAC keys have two primary pieces, **an access ID** and **a secret**.
    - **Access ID:**
        - An alphanumeric string linked to a specific service or user account.
        - When linked to a service account, the string is 61 characters in length, and when linked to a user account, the string is 24 characters in length.
        - The following shows an example of an access ID:

        - GOOGTS7C7FUP3AIRVJTE2BCDKINBTES3HC2GY5CBFJDCQ2SYHV6A6XXVTJFSA
    - **Secret:**
        - A 40-character Base-64 encoded string that is linked to a specific access ID.
        - A secret is a pre-shared key that only you and Cloud Storage know. You use your secret to create signatures as part of the authentication process.
        - The following shows an example of a secret:
            - bGoa+V7g/yqDXvKRqq+JTFn4uQZbPiQJo4pf9RzJ

- **Best practices**
    - Do not share your HMAC key secret.
    - Regularly change your keys as part of a key rotation.

- **Restrictions**
    - HMAC keys can only be used to make requests to the XML API, not the JSON API.
    - You can have a maximum of 5 HMAC keys per service account.

Lab on Object Life Cycle Management - Cloud Storage Bucket

- Object Lifecycle Management, define a lifecycle configuration, which must be set on a bucket.
- The configuration contains a set of rules which apply to current and future objects in the bucket. When an object meets the criteria of one of the rules, Cloud Storage automatically performs a specified action on the object. Here are some example use cases:
    - Downgrade the storage class of objects older than 365 days to Coldline storage.
    - Delete objects created before January 1, 2019.
    - Keep only the 3 most recent versions of each object in a bucket with versioning enabled.
- Based on condition what action needs to perform on object.
    - Like setting a Time to Live (TTL) for objects, retaining non-current versions of objects, or "downgrading" storage classes of objects to help manage costs, Cloud Storage offers the Object Lifecycle Management feature.

- **Condition**
  - Object age
  - Object file type
  - After some specific date
- **Action**
  - A lifecycle rule specifies exactly one of the following actions:
    - Delete
    - SetStorageClass
    - AbortIncompleteMultipartUpload
  - Transition to different storage class for high performance
    - Like – Standard to Nearline
    - Coldline to Delete

Lab on Object Versioning - Cloud Storage Bucket

- To support the retrieval of objects that are deleted or replaced, Cloud Storage offers the Object Versioning feature.
- Cloud Storage retains a non-current object version each time you replace or delete a live object version, as long as you do not specify the generation number of the live version.
  - Non-current versions retain the name of the object, but are uniquely identified by their generation number.
  - Non-current versions only appear in requests that explicitly call for them to be included.
- You permanently delete versions of objects by including the generation number in the deletion request or by using Object Lifecycle Management.
- Non-current versions of objects exist independently of any live version.
- If you disable Object Versioning:
  - The bucket no longer accumulates new non-current versions of objects.
  - Object versions that already exist in the bucket are unaffected.
- Help to prevent accidental deletion of object
- Enable/Disable versioning at bucket level.

- Get access to older version with (object key + version number)
    - If you don't need earlier version, delete it & reduce storage cost
    - If you don't specify version number, always retrieve latest version

Signed url – temporary access

- A signed URL is a URL that provides limited permission and time to make a request.
- Signed URLs contain authentication information in their query string, allowing users without credentials to perform specific actions on a resource.
- When you generate a signed URL, you specify a user or service account which must have sufficient permission to make the request that the signed URL will make.

- After you generate a signed URL, anyone who possesses it can use the signed URL to perform specified actions, such as reading an object, within a specified period of time.
- When should you use a signed URL?
    - In some scenarios, you might not want to require your users to have a Google account in order to access Cloud Storage, but you still want to control access using your application-specific logic.
    - The typical way to address this use case is to provide a signed URL to a user, which gives the user read, write, or delete access to that resource for a limited time.
    - You specify an expiration time when you create the signed URL. Anyone who knows the URL can access the resource until the expiration time for the URL is reached or the key used to sign the URL is rotated.
- Options for generating a signed URL.Cloud Storage supports several methods for generating a signed URL:
    - V4 signing with service account authentication.
    - Signing with HMAC authentication.
    - V2 signing with service account authentication.

Lab on Cloud IAM roles - Cloud Storage bucket

- IAM allows you to control who has access to the *resources* in your Google Cloud project.
- Resources include Cloud Storage buckets and objects stored within buckets, as well as other Google Cloud entities such as Compute Engine instances.
- Principals are the "who" of IAM. Principals can be individual users, groups, domains, or even the public as a whole.
- Principals are granted roles, which give them the ability to perform actions in Cloud Storage as well as Google Cloud more generally.
- Each role is a collection of one or more permissions.
    - Permissions are the basic units of IAM: each permission allows you to perform a certain action.

  - Who can do what on GCS at what level?
  - Permissions
      - Apply at project level
      - Apply at bucket level
  - Apply Project level
    - IAM
    - Different Predefined Role
        - Storage Admin
        - Storage Object Admin
        - Storage Object Creator
        - Storage Object Viewer
    - Create Custom Role
  - Apply at Bucket level: (https://cloud.google.com/storage/docs/access-control)
    - Assign bucket level roles
        - Select bucket & assign role
        - To user
        - To other GCP services – Service account or product

Lab on Uniform bucket level access - Cloud Storage bucket

- Cloud Storage offers two systems for granting users permission to access your buckets and objects: IAM and Access Control Lists (ACLs).
- These systems act in parallel - in order for a user to access a Cloud Storage resource, only one of the systems needs to grant the user permission.
    - IAM is used throughout Google Cloud and allows you to grant a variety of permissions at the bucket and project levels.
    - ACLs are used only by Cloud Storage and have limited permission options, but they allow you to grant permissions on a per-object basis.
- In order to support a uniform permissioning system, Cloud Storage has uniform bucket-level access. Using this feature on a bucket disables ACLs for all Cloud Storage resources in the bucket; access to Cloud Storage resources then is granted exclusively through IAM.
- After you enable uniform bucket-level access, you can reverse your decision for 90 days.
- Using uniform bucket-level access is recommended, because it unifies and simplifies how you grant access to your Cloud Storage resources.
- Using uniform bucket-level access also enables you to use other Google Cloud security features such as domain restricted sharing, workforce identity federation, and IAM Conditions.
- You want the uploader of an object to have full control over that object, but less access to other objects in your bucket.
- You can enable uniform bucket-level access either when you create a new bucket, or when you explicitly enable uniform bucket-level access on an existing bucket.
- Individual object ownership no longer exists.

Lab on Access Control Lists(ACLs) - Cloud Storage bucket

- An access control list (ACL) is a mechanism you can use to define who has access to your buckets and objects, as well as what level of access they have.
- In Cloud Storage, you apply ACLs to individual buckets and objects. Each ACL consists of one or more entries.

- An entry gives a specific user (or group) the ability to perform specific actions.
- Each entry consists of two pieces of information:
    - A permission, which defines what actions can be performed (for example, read or write).
    - A scope (sometimes referred to as a grantee), which defines who can perform the specified actions (for example, a specific user or group of users).
- As an example, suppose you have a bucket that you want anyone to be able to access objects from, but you also want your collaborator to be able to add or remove objects from the bucket.
    - In this case, your ACL would consist of two entries:
        - In one entry, you would give READER permission to a scope of allUsers.
        - In the other entry, you would give WRITER permission to the scope of your collaborator (there are several ways to specify this person, such as by their email).
- The maximum number of ACL entries you can create for a bucket or object is 100.
- When a user requests access to a bucket or object, the Cloud Storage system reads the bucket or object ACL and determines whether to allow or reject the access request.
    - If the ACL grants the user permission for the requested operation, the request is allowed.
    - If the ACL does not grant the user permission for the requested operation, the request fails and a 403 Forbidden error is returned.

Lab on Make data public - Cloud Storage bucket

- How to make objects you own readable to everyone on the public internet.
- Some data stored in Cloud Storage is configured so that it's readable by anyone at any time. This *public data* can be accessed in several ways, depending on how you want to work with the data.
- When an object is shared publicly, any user with knowledge of the object URI can access the object for as long as the object is public.

- https://cloud.google.com/storage/docs/access-control/making-data-public#permissions-cli

Public access prevention

- Public access prevention protects Cloud Storage buckets and objects from being accidentally exposed to the public. When you enforce public access prevention, no one can make data in applicable buckets public through IAM policies or ACLs. There are two ways to enforce public access prevention:
- You can enforce public access prevention on individual buckets.
- If your bucket is contained within an organization, you can enforce public access prevention by using the organization policy constraint storage.publicAccessPrevention at the project, folder, or organization level.

Lab on Creating Cloud Storage Bucket - Client library(Python)

- Google Cloud Storage is almost infinitely scalable and guarantees consistency: when a write succeeds, the latest copy of the object will be returned to any GET, globally.
- Supported Python Versions - Python >= 3.7
- Unsupported Python Versions - Python <= 3.6
- To install Cloud Storage - pip install google-cloud-storage
    - google-cloud-storage 2.7.0
    - https://pypi.org/project/google-cloud-storage/

Lab on Bucket Retention policy - Cloud Storage Bucket

- Cloud Storage allows you to configure object retention with temporary retention, a specific Retention Policy which automatically tracks retention expiration for objects, and even event based holds which allow you determine when a Retention Policy begins.
- With Bucket Lock, you can meet regulatory and compliance requirements, such as those associated with FINRA, SEC, and CFTC. You can also use Bucket Lock to address certain health care industry retention regulations.
- Combined with Cloud Storage Object Lifecycle Management, you can create a complete data retention strategy, for example automatically removing data from buckets when retention policies are met.
- Minimum duration for which bucket will be protected from

- Deletion
- Modification

Pricing on Cloud Storage buckets

- Cloud Storage pricing is based on the following components:
  - **Data storage**
    - the amount of data stored in your buckets. Storage rates vary depending on the storage class of your data and location of your buckets.
  - **Data processing - Operations Usage**
    - processing done by Cloud Storage, which includes operations charges, any applicable retrieval fees, and inter-region replication.
  - **Network usage**
    - amount of data read from or moved between your buckets.
  - **Retrieval and early deletion fees:**
    - It is applicable for data stored in the Nearline Storage, Coldline Storage, and Archive Storage classes.
- https://cloud.google.com/storage/pricing

Host a static website - Cloud Storage Bucket

- Static websites are a good option for sites like blogs — where the page rarely changes after it has been published, or where there isn't any dynamically-generated content.
- Static web pages can contain client-side technologies such as HTML, CSS, and JavaScript.
- They cannot contain dynamic content such as server-side scripts, like PHP.
- Cloud Storage doesn't support custom domains with HTTPS on its own

Encryption and Types in Google Cloud

- Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge.
- Besides this standard, Google-managed behavior, there are additional ways to encrypt your data when using Cloud Storage.

1. **Server-side encryption:** encryption that occurs after Cloud Storage receives your data, but before the data is written to disk and stored.
    1. **Customer-managed encryption keys:**
        - You can create and manage your encryption keys through Cloud Key Management Service.
        - Customer-managed encryption keys can be stored as software keys, in an HSM cluster, or externally.
    2. **Customer-supplied encryption keys:**
        - You can create and manage your own encryption keys.
        - These keys act as an additional encryption layer on top of the standard Cloud Storage encryption.
2. **Client-side encryption:**
    - Encryption that occurs before data is sent to Cloud Storage. Such data arrives at Cloud Storage already encrypted but also undergoes server-side encryption.

Google managed Encryption keys

- Cloud Storage manages server-side encryption keys on your behalf using the same hardened key management systems that we use for our own encrypted data, including strict key access controls and auditing.
- Cloud Storage encrypts user data at rest using AES-256, in most cases using Galois/Counter Mode (GCM).
- There is no setup or configuration required, no need to modify the way you access the service, and no visible performance impact.
- Data is automatically decrypted when read by an authorized user.

Customer managed Encryption keys

- If you need more control over key operations than, Google-managed encryption keys allows to use customer-managed encryption keys.
- These keys are created and managed using **Cloud Key Management Service (Cloud KMS)**, and you store the keys as software keys, in an HSM cluster, or externally.

- You can use customer-managed encryption keys on individual objects, or configure your bucket to use a key by default on all new objects added to a bucket.
- When using a customer-managed encryption key, an object is encrypted with the key by Cloud Storage at the time it's stored in a bucket, and the object is automatically decrypted by Cloud Storage when the object is served to requesters.
    - Create keyring in Cloud KMS
    - key will be managed by customer. Like Key rotation
    - Key Management service – GCP services
- **Cloud Key Management Service (Cloud KMS)** use, rotate, and manage cryptographic keys.
- A cryptographic key is a resource that is used for encrypting and decrypting data or for producing and verifying digital signatures.
- To perform operations on data with a key, use the Cloud KMS API.

Customer supplied Encryption keys

- As an additional layer on top of Google-managed encryption keys, you can choose to provide your own AES-256 encryption key, encoded in standard Base64. This key is known as a customer-supplied encryption key.
- If you provide a customer-supplied encryption key, Cloud Storage does not permanently store your key on Google's servers or manage your key.
- Instead, when you provide your key for each Cloud Storage operation, and your key is purged from Google's servers after the operation is complete.
- Cloud Storage stores only a cryptographic hash of the key so that future requests can be validated against the hash. Your key cannot be recovered from this hash, and the hash cannot be used to decrypt your data.
- When is the key used?
    - When you apply a customer-supplied encryption key to an object, Cloud Storage uses the key when encrypting:
        - The object's data.
        - The object's CRC32C checksum.

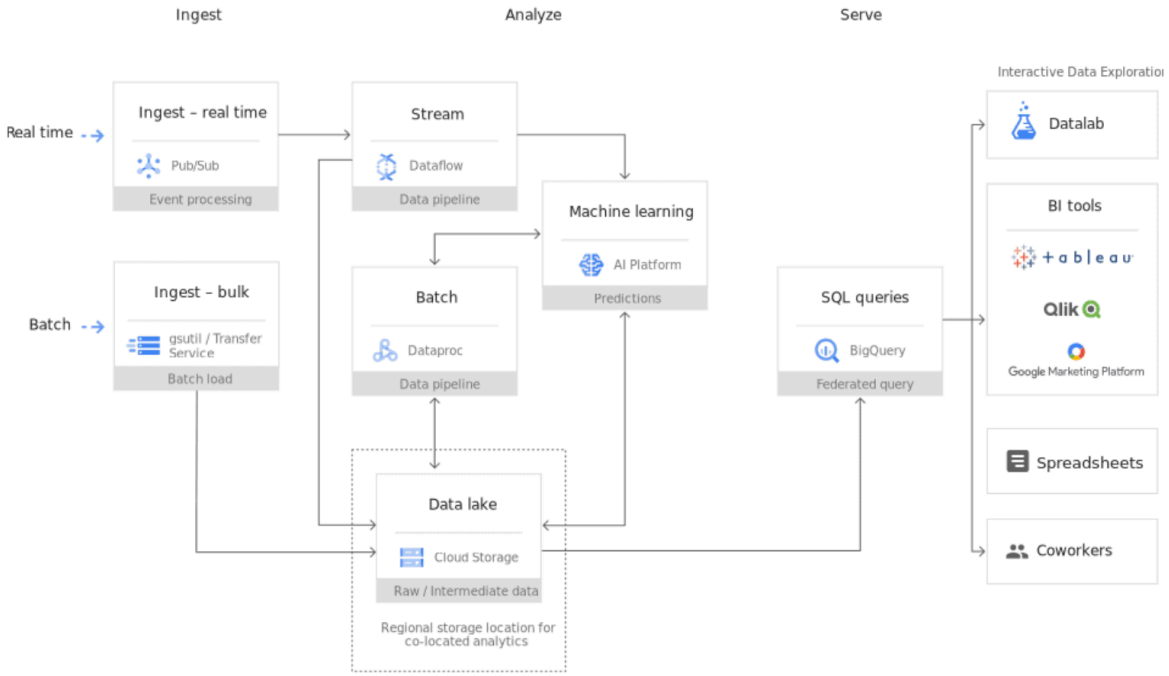
- The object's MD5 hash.

- only in CLI commands (https://cloud.google.com/storage/docs/encryption/customer-supplied-keys)

Use Cases For Google Cloud Storage

1.) Backups and archives: It provides fast, low-cost, highly durable storage for data accessed less than once a month. It is perfect for reducing the cost of backups and archives while still retaining immediate access. Backup data in Cloud Storage can be used for more than just recovery because all storage classes have ms latency and are accessed through a single API.

2.) Integrated repository for analytics and ML: The highest level of availability and performance within a single region is ideal for compute, analytics, and machine learning workloads in a particular region. Cloud Storage is also strongly consistent, giving you confidence and accuracy in analytics workloads.



3.) Media content storage and delivery: Geo-redundant storage with the highest level of availability and performance is ideal for low-latency, high-QPS content serving to users distributed across geographic regions. Google Cloud

Storage service provides the availability and throughput needed to stream audio or video directly to apps or websites.