**Google Cloud Storage**

**Core Concepts**

* **Buckets and Objects**: Learn how data is stored as objects inside globally unique buckets.
* **Storage Classes**: Understand Standard, Nearline, Coldline, and Archive tiers for cost-effective storage.
* **Bucket Locations**: Choose between regional, dual-region, and multi-region for latency and redundancy.

**Security and Access Control**

* **IAM Roles**: Manage access at bucket and object levels using Identity and Access Management.
* **Uniform Bucket-Level Access**: Simplify permission management across all objects in a bucket.
* **Encryption**: Explore Google-managed, customer-managed, and customer-supplied encryption keys.

**Lifecycle and Versioning**

* **Object Lifecycle Management**: Automate transitions between storage classes or deletions based on age.
* **Object Versioning**: Preserve and recover previous versions of objects.

**Tools and Interfaces**

* **gsutil CLI**: Command-line tool for scripting storage operations.
* **Client Libraries**: SDKs for Python, Java, Go, and more.
* **REST APIs**: Programmatic access to buckets and objects.

**Pricing and Quotas**

* **Storage Pricing**: Based on class, location, and data volume.
* **Data Access Costs**: Retrieval fees for Nearline, Coldline, and Archive.
* **Quotas and Limits**: Understand default limits and how to request increases.

**Integration and Use Cases**

* **BigQuery Integration**: Load data from GCS into BigQuery using operators like GCSToBigQueryOperator.
* **Cloud Functions and Dataflow**: Trigger workflows or transform data directly from GCS.
* **Disaster Recovery and Archiving**: Use Coldline and Archive for long-term storage.

**Core Concepts**

1. **Buckets and Objects**
   1. Buckets are **globally unique containers** that hold your data objects. Think of them like folders, but with cloud-native features.
   2. **Key Properties**
      1. **Name**: Must be globally unique (e.g., ananth-insurance-data)
      2. **Location**: Choose from:
         1. **Regional** (e.g., asia-south1) for low-latency access
         2. **Multi-region** for high availability
         3. **Dual-region** for disaster recovery
      3. **Storage Class**: Controls cost and access frequency:
         1. STANDARD
         2. NEARLINE
         3. COLDLINE
         4. ARCHIVE
      4. Creating a bucket with gsutil command
         1. **gsutil mb -l asia-south1 -c STANDARD gs://ananth-data-bucket/**
   3. **Objects: Your Stored Files -** Objects are the actual **data files** stored inside buckets. They can be anything—CSV, JSON, images, logs, backups.
      1. **Object Structure:** 
         1. **Data** – Immutable binary content.
         2. **Metadata** - Includes name, content type, size, creation time, encryption info, and custom tags.
2. **Storage Classes**
   1. Storage classes in **Google Cloud Storage (GCS)** are designed to help you optimize cost and performance based on how frequently you access your data.
   2. Each object in GCS is assigned a **storage class**, which determines:
      1. **Cost per GB stored**
      2. **Minimum storage duration**
      3. **Retrieval fees**
      4. **Availability SLA**
   3. All classes offer:
      1. 99.999999999% durability
      2. Millisecond access latency
      3. Global accessibility
      4. Uniform API and security features

| **Storage Class** | **Best For** | **Min Duration** | **Retrieval Fee** | **Availability SLA** | **Typical Use Case** |
| --- | --- | --- | --- | --- | --- |
| **Standard** | Frequently accessed data | None | None | ≥99.99% | Web content, active datasets |
| **Nearline** | Accessed ~once/month | 30 days | Yes | ≥99.9% | Monthly reports, backups |
| **Coldline** | Accessed ~once/quarter | 90 days | Yes | ≥99.9% | Quarterly archives, DR snapshots |
| **Archive** | Rarely accessed, long-term data | 365 days | Yes | ≥99.9% | Legal archives, compliance, DR backups |

1. **Bucket Locations**
   1. A **bucket location** defines the **physical region(s)** where your data is stored. You select this when creating a bucket, and it **cannot be changed later** (unless using relocation features)
   2. The location impacts:
      1. **Latency**: How fast data is accessed
      2. **Availability**: Uptime guarantees
      3. **Redundancy**: How data is replicated
      4. **Pricing**: Storage and transfer costs
   3. **Location Types**
      1. **Regional**
         1. Stores data in a **single region** (e.g., asia-south1 for Mumbai).
         2. Best for **low-latency access** when compute and storage are co-located.
         3. **No replication charges** within the same region.
         4. Ideal for:
            1. Airflow DAG logs
            2. BigQuery staging files
            3. Insurance datasets accessed frequently
      2. **Dual Regional**
         1. Stores data in two specific regions.
         2. Offers Automated failover and turbo replication.
         3. Higher availability than Regional buckets.
         4. Great for:
            1. Disaster recovery setups.
            2. Cross – region analytics.
            3. Insurance backups with geographic redundancy.
      3. **Multi Regional**
         1. Stores data across multiple regions in broad area.
         2. Highest availability but higher cost and replication charges apply.
         3. Best for:
            1. Global content delivery
            2. Public datasets

**Security and Access Control**

1. **IAM Roles**: Manage access at bucket and object levels using Identity and Access Management
   1. IAM in GCS supports two level of controls.
      1. Bucket level – Permission apply to entire buckets.
      2. Object level – fine grained access to individual objects inside buckets.
   2. To enable **object-level IAM**, the bucket must have **Uniform bucket-level access disabled**, allowing ACLs or fine-tuned IAM policies per object
   3. **Types of Identities:**
      1. **Google Accounts –** Individual Users.
      2. **Service Accounts –** For Apps and Automation.
      3. **Google Groups –** group permissions
      4. **G Suite/ Cloud Identity domains –** domain-wide access control.
   4. **Built in IAM roles for GCS**
      1. **roles/storge.admin –** full control over buckets and Objects.
      2. **roles/storage.objectAdmin –** create, delete, update objects only.
      3. **roles/storage.objectViewer –** Read only access to objects.
      4. **roles/storge.objectCreator** – upload objects but can’t read /delete existing objects.
2. **Uniform Bucket-Level Access**: Simplify permission management across all objects in a bucket.Uniform Bucket-Level Access (UBLA) enforces **centralized IAM policy** on **buckets only**, disabling legacy object-level ACLs (Access Control Lists). This means permissions are set **only at the bucket level** using IAM roles—not per object.
   1. **What happens when UBLA is enabled?**
      1. All object level ACLs are disabled. No reader/writer level access at object level.
      2. IAM roles become the only way to manage access.
      3. Tools relying on ACLs need to be updated
      4. Existing ACLs are ignored – even if set before enabling UBLA.
3. **Encryption**:
   1. Summary Comparison Table

| **Feature** | **GMEK** | **CMEK** | **CSEK** |
| --- | --- | --- | --- |
| Key Ownership | Google | Customer via Cloud KMS | Customer |
| Auditability | Limited | Full visibility via IAM & Audit Logs | Manual |
| Key Rotation | Automatic | Manual or scheduled via KMS | Manual |
| Setup Complexity | Minimal | Moderate | High |
| Use Case | General workloads | Regulatory/data-sensitive projects | Legal/data sovereignty constraints |

**Tools and Interfaces**

1. **gsutil CLI**: Command-line tool for scripting storage operations.
   1. gsutil is a Python-based command-line tool that allows you to interact with GCS buckets and objects. It comes bundled with the [Google Cloud SDK](https://cloud.google.com/sdk), making it ideal for scripting tasks like uploading, downloading, syncing, setting permissions, and more.
   2. Key commands:
      1. gsutil cp mydata.csv gs://my-bucket/data/ - **upload**
      2. gsutil cp gs://my-bucket/data/mydata.csv ./local-folder/ - **download**
      3. gsutil rsync -r ./local-folder gs://my-bucket/backup-folder/ - **sync directories**
      4. gsutil rm gs://my-bucket/data/mydata.csv – **Delete objects**
      5. gsutil mb -l ASIA-SOUTH1 gs://my-new-bucket/ - **create bucket**
      6. gsutil iam set bucket-policy.json gs://my-secure-bucket/ - **set IAM policy**
      7. gsutil lifecycle set lifecycle-config.json gs://my-bucket/ - **lifecycle management**
      8. gsutil du -s gs://my-bucket/ - **bucket usage stats**
      9. gsutil -m cp -r ./large-folder gs://my-bucket/ - **multithreaded performance**