**Google Cloud Storage:**

* **Google Cloud Storage** is a **RESTful** online file storage webservice for storing and accessing data on Google Cloud Platform infrastructure.
* It is an Infrastructure as a Service (IaaS)

**Cloud Storage:**

* Cloud Storage is unified object storage service.
* Cloud Storage is a persistent storage, it is durable, replicated and 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.
* 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 objects in bucket.

**Key Terms:**

➤ **Buckets:** Basic 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.

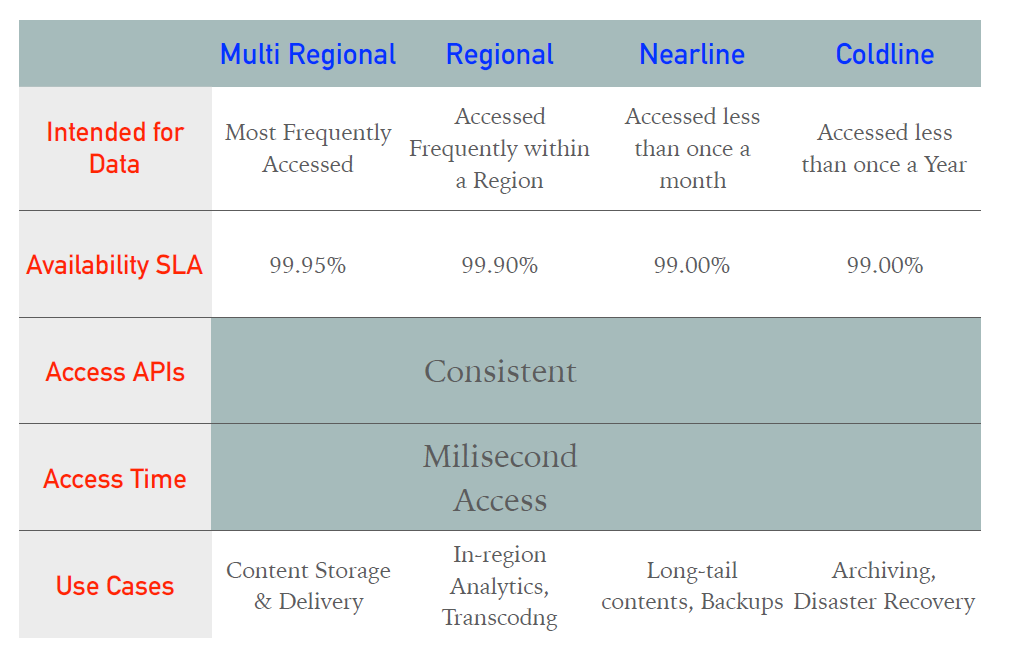
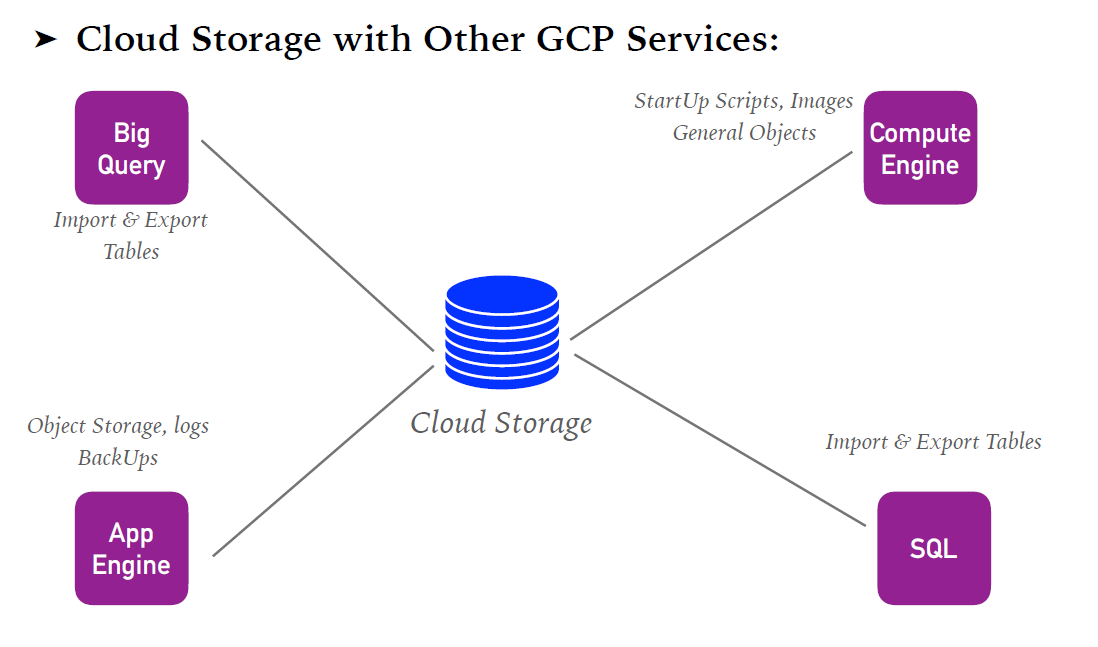
➤ **Bucket names:** Should be unique as the name of the buckets stored in single Cloud Storage namespace. Also, bucket names can be used with a CNAME redirect, which means they need to conform to DNS naming conventions.

➤ **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.
* Objects have two components: object data and object metadata.
* The object data component is usually a file that you want to store in Google Cloud Storage.
* 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 enabled explicitly, in absence of Object Versioning, new objects terminates the old.
* Cloud Storage offers life cycle management policy for the objects in bucket.

**What kind of storage is used for what use case?**

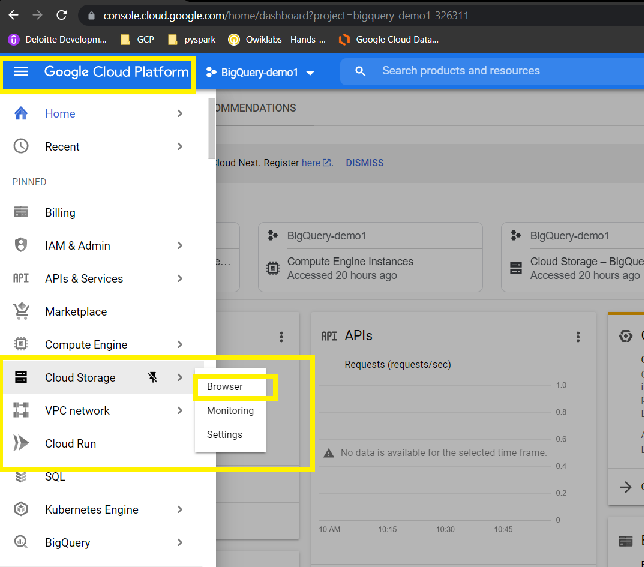
1. Block storage for GCP VMs – **Persistent Disk**
2. Immutable blobs – **Cloud Storage**
3. RDBMS – **CloudSQl & CloudSpanner**
4. NoSQL Database (json, mongo dB) – **Datastore**
5. NoSQL key-value dB - **Bigtable**
6. Import data in cloud – **transfer service**



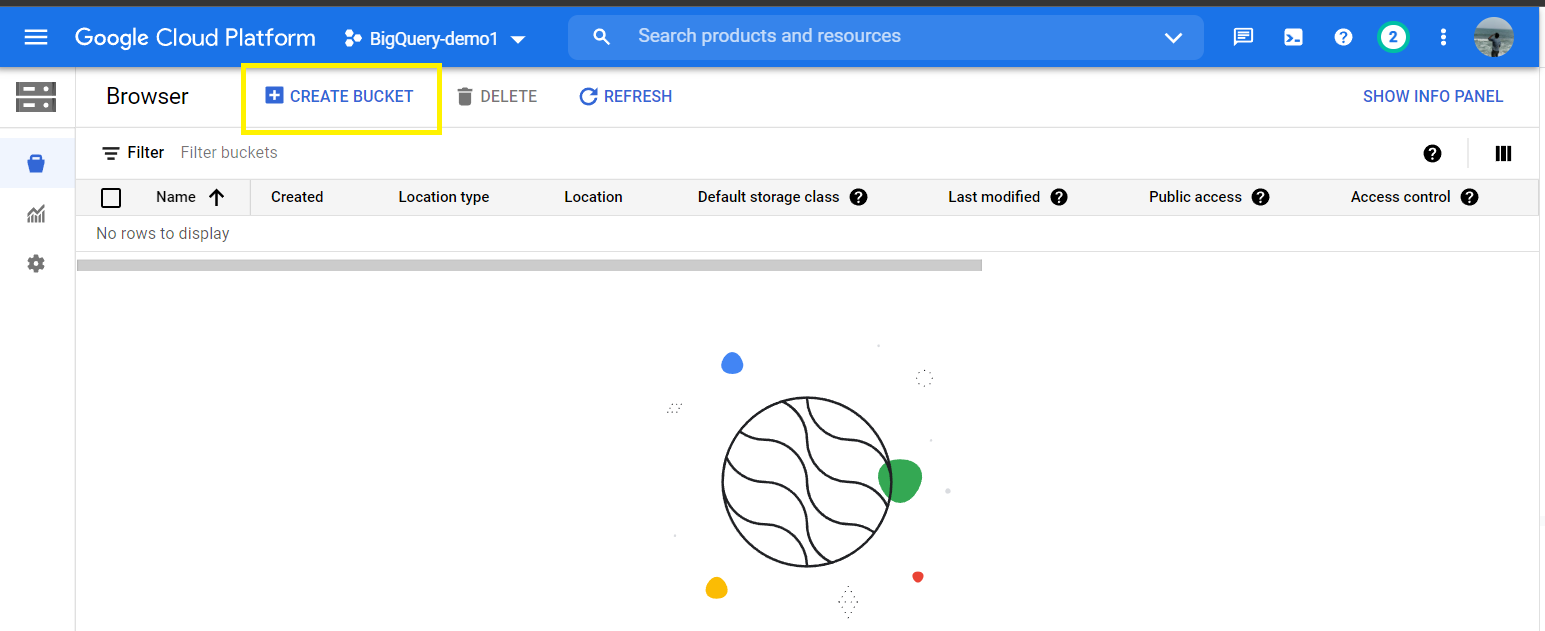
1. **Cloud storage can connect with BQ for importing and exporting tables with low latency.**
2. **It can be used in compute engine using scripts like**
3. **It can be used with app engine for logs, backups and so on.**
4. **And it is used with sql for importing and exporting tables.**

**Working with Cloud Storage buckets:**

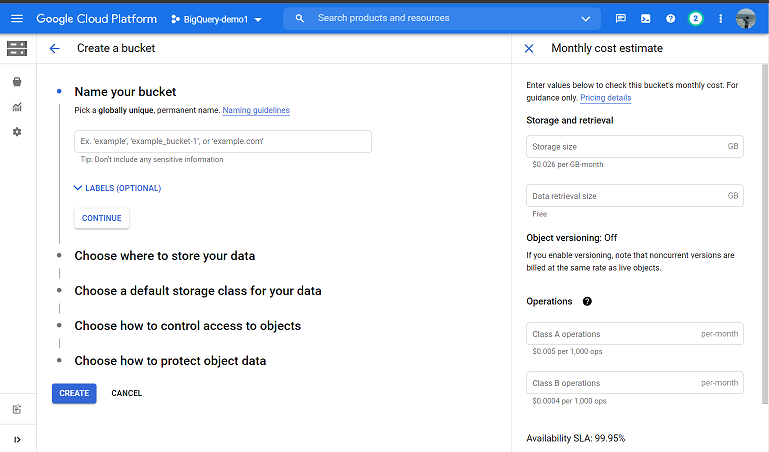
* Create GCS bucket with GUI or UI:
  + GCP organizes resources into projects. This allows you to collect all of the related resources for a single application in one place.
  + **Go to navigation menu.**

****

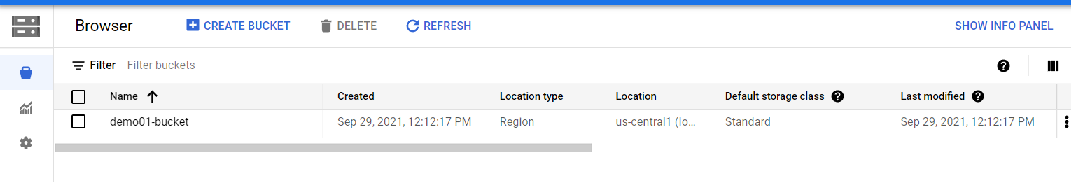
* + Storage browser page opens.

****

* + If you have created any bucket in past, it will be displayed here otherwise it will be empty as shown.
  + Click on **Create bucket** option, to create a new storage in GCS.
  + **Name your bucket:** **Bucket names** must be globally unique (among all buckets ever created by any user).
  + **Choose where to store the data:** Select the **Location Type** for your data.
    - The default, **Multi-region**, delivers the highest availability.
    - For lower latency, you may wish to choose **Regional**.
    - Choosing **Dual-region** strikes a balance between them.
  + **Choose a default storage class for your data: Select a default storage class** for data in this bucket. The default is Standard, but you may wish to choose a different option based on your needs. This decision should be based around how long you plan to store your data and how often it will be accessed.
  + **Choose how to control access to objects:** Specify how **to control access to objects**, whether you want to control access at the bucket level only (Uniform), or to also enable individual stored objects to have additional permission settings (Fine-grained). <https://cloud.google.com/storage/docs/uniform-bucket-level-access>
    - There are some optional settings for **Encryption, Retention policy, and Labels** that you may set. (while creating I have entered retention policy as 1 days so after a day my data will be removed)



* + Click on **create button** to create the storage. You can find SLA for the bucket created along with the monthly cost estimate details.



Upload data to your bucket :

* There are two ways to upload data to your bucket using the
  + Cloud Storage Console
  + gsutil command-line tool.

# Upload data via the Console:

# Click on the storage name (i.e., demo01-bucket ). You can view the below page opening up.

# 

# Click on upload files/ upload folder or create a folder then upload files or folder according to the requirement. I have uploaded a simple json file into my bucket.

# 

# 

# By clicking on name of storage file created. You can view file size, type of file, created date, and details about storage class. By default, public access will be not available to public.

# 

# If you want to change it from not public to available to public, you can use edit permissions

# 

# If you want to delete the data, just click on the storage and delete it as shown.

# 

**Create Google Cloud Storage Bucket Using CLI In Google Cloud Platform**

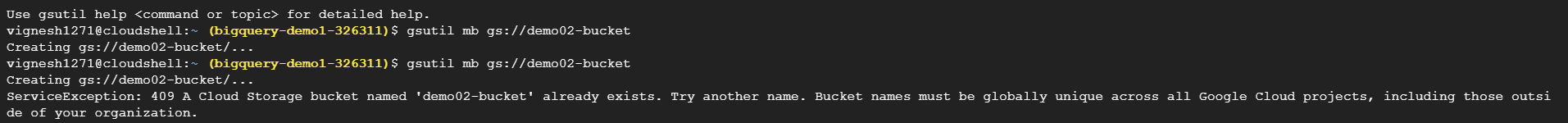
* Create cloud storage bucket using cloud shell
* Copy/upload data in bucket
* Copy one bucket to another
* List bucket contents
* Object versioning in bucket
* Bucket object lifestyle
* Change permission of bucket in command line

Using **gsutil** – command line (Cloud Shell)  Click on this icon and you can see shell similar to this opening up.

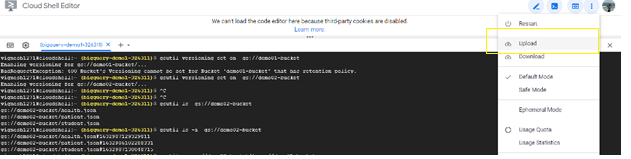
To set your Cloud Platform project in this session use “gcloud config set project [PROJECT\_ID]”

* gcloud config set project bigquery-demo1-326311
* gsutil –help
* To list all in console just use “ls”.
* Create a bucket
  + gsutil mb gs://demo02-bucket

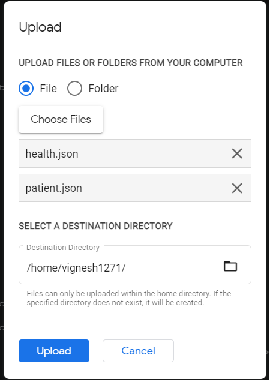
If bucketname already exists we will get error code 409. 409 A Cloud Storage bucket named 'demo02-bucket' already exists. Try another name. Bucket names must be globally unique across all Google Cloud projects, including those outside of your organization.

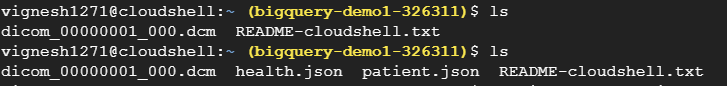


* To upload file from local to GCP cloud storage.

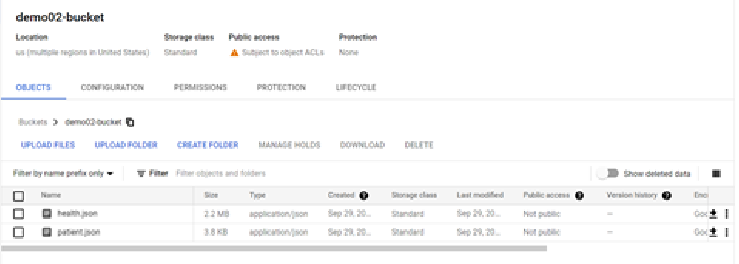


* Click on Upload button and you can choose file to upload into GCP. Here I selected health.json and patient.json file into Cloud storage from cloudshell.

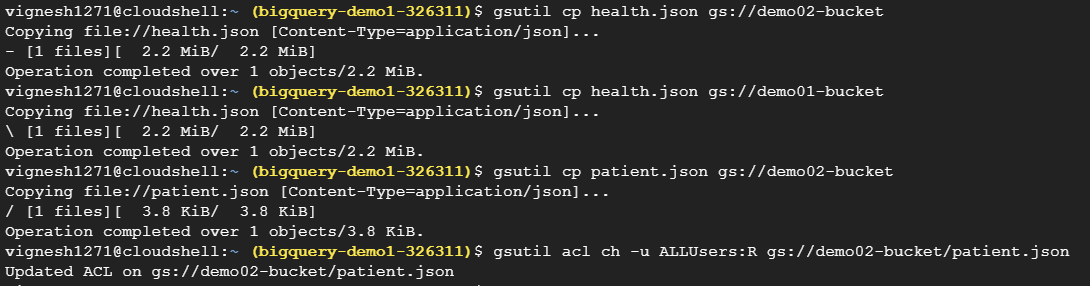




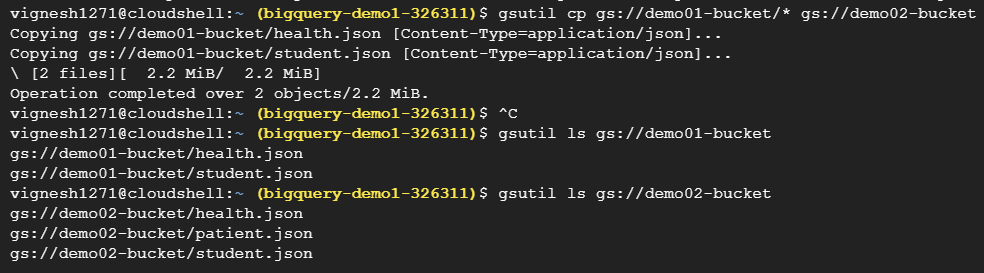
The same looks in GCP UI.



* Then to load the uploaded file into bucket created, follow below commands. To give global access to a file in a bucket
  + gsutil acl ch -u ALLUsers:R gs://demo02-bucket/patient.json



* Copy data from one bucket to another bucket. Error: AccessDeniedException: 403 Object 'demo01-bucket/health.json' is subject to bucket's retention policy and cannot be deleted, overwritten or archived until 2021-09-30T02:01:14.331731-07:00
  + gsutil -D ls gs://demo01-bucket/\* (not resolved)



NOTE: If retention policy is present in bucket, you are likely to face this error “Error: 403, subject to bucket's retention policy and cannot be deleted, overwritten or archived until 2021-09-30T02:01:14.33”

* How to enable version in gsutil
  + gsutil versioning get gs://demo01-bucket

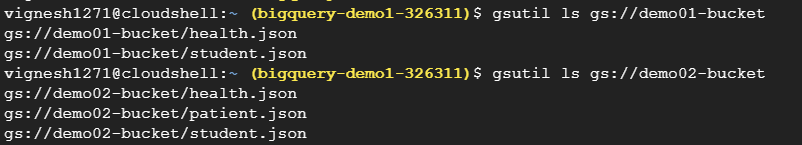


NOTE: response: gs://demo01-bucket: Suspended (BadRequestException: 400 Bucket's Versioning cannot be set for Bucket 'demo01-bucket' that has retention policy.)

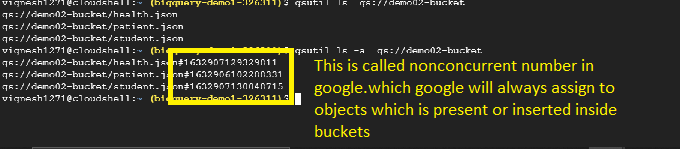
* Enabling the versioning in demo01-bucket
  + gsutil versioning set on gs://demo01-bucket



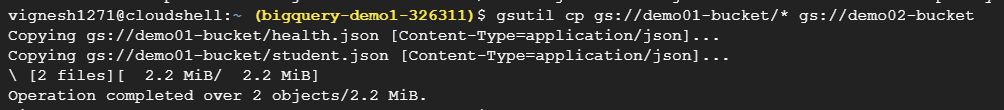
* To list all the files in the bucket
  + gsutil ls gs://<BUCKET\_NAME>



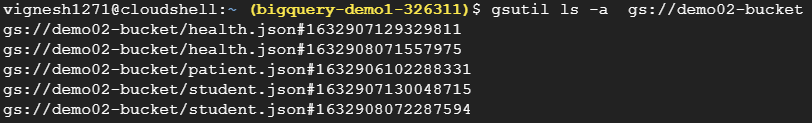
* To list all hidden files in the bucket
  + gsutil ls -a gs://<BUCKET\_NAME>



* After enabling version in gs://demo02-bucket, Try running the command
  + gsutil cp gs://demo01-bucket/\* gs://demo02-bucket



* To list all the files in the bucket
  + gsutil ls -a gs://demo02-bucket



* Lifecycle of any object in the bucket
  + gsutil lifecycle get gs://demo02-bucket
* Created json file(bucket2\_lifecycle.json) and upload into the bucket using cloudshell.dev. After that go to cloudshell and execute the below commands.
  + gsutil lifecycle set bucket2\_lifecycle.json gs://demo02-bucket

Sources:

* <https://cloud.google.com/storage/docs/gsutil/commands/mb>
* <https://cloud.google.com/storage/docs/quickstart-gsutil>