GCP Certification Series: 2.3 Planning and configuring data storage options

Prashanta Paudel

Oct 18, 2018 · 15 min read

To begin the topic I think having the background of the storage systems will be required in order to understand things properly. Let us talk about storage in general.

A storage device is any computing hardware that is used to store, retrieve or move files and objects. It can hold data both permanently or temporarily and can be internal or external to the computer, server or similar computing device.

Hard disk drives, SDD, pen drives, zip drive, floppy drives etc are all example of storage devices.

Storage devices are core component of any computing systems as they are required to store the operating system and user data. Storage devices are required to store all the applications used in personal computer or server.

Storage devices are generally found in various capacity to fit for the various purposes. As we know 8 bits form a byte, all the storage devices sizes are usually multiple of 8.

A standard computer can have various storage devices including RAM, cache, Hard disk as well as Optical disks and USB drives.

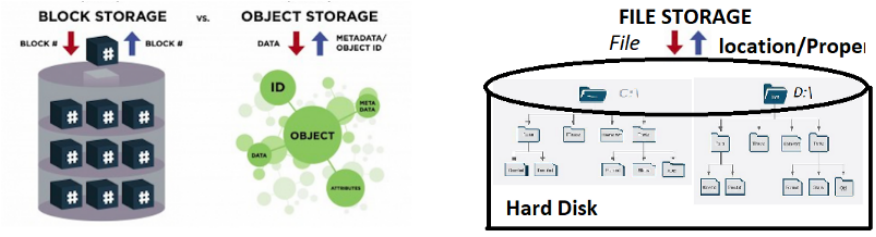
There are two types of storage devices

1. Primary storage devices(RAM): These are in small sizes compared to secondary storage and hold the data temporarily and are internal to the computer. These devices are very fast in data access such as RAM including cache memory.
2. secondary storage devices(HDD): These devices usually have large storage capacity and they store data permanently. These are

usually Hard disks, SSD, Optical disk or USB drive.

Storage Systems

Irrespective of where the storage system is used all the storage devices can be divided into



Type of storage systems

1. File storage————— files and folders
2. Block storage————Databases, OS
3. Object storage————-buckets, multimedia

1. File storage

File storage systems are based on the traditional paper-based file system where files are extracted from folders which are placed in drawers and racks and searched in alphabetical order.

files are hierarchically stored in folders and many subfolders into one root folder. To access the file you have to know the whole path to the last folder and then only you can access the file.

File storage format

The file system can differ from vendor to vendor. For example

- **Windows use FAT,FAT32,NTFS**
- **Mac uses FAT, exFAT, HFS and latest APF**
- **Linux uses ext2, ext3,ext4, XFS, JFS and btrfs.**

FAT- File allocation table

NTFS- New Technology File System

APF- Apple File System

ext2- Extended file system -2

Some file systems are designed for specific applications and work only when used together with the application.

File system and metadata

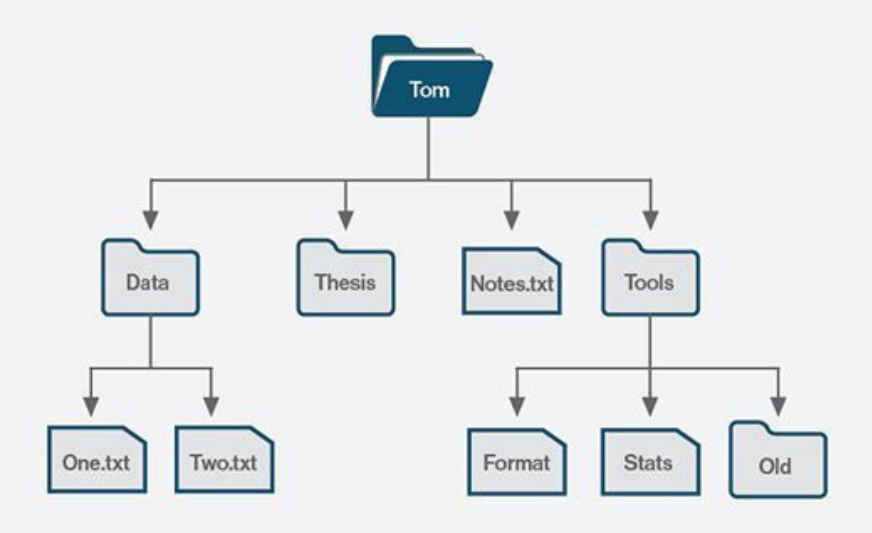
The file system uses metadata to store and pull file information from the storage. Some of the information in metadata could be like

- Creation date
- Modified date
- Owner of the file
- Permissions
- sharing options etc

File storage system

In file storage system you write a file, give it a name, place it inside the folder and nest it under the folder to make it more relevant. In this way, files are organized into a hierarchy, with directories and sub-directories. Each file also has a limited set of metadata associated with it, such as the file name, the date it was created, and the date it was last modified etc.

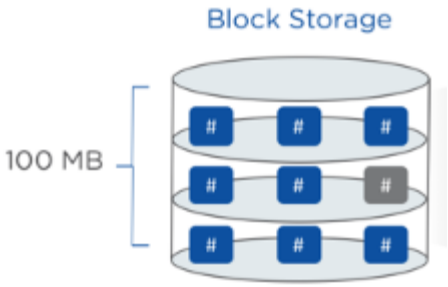
Data is accessed as file IDs (server name + directory path + filename) over a shared network and the storage server manages the data on disk. Network attached storage(NAS) depends on this file system for access and retrieval.



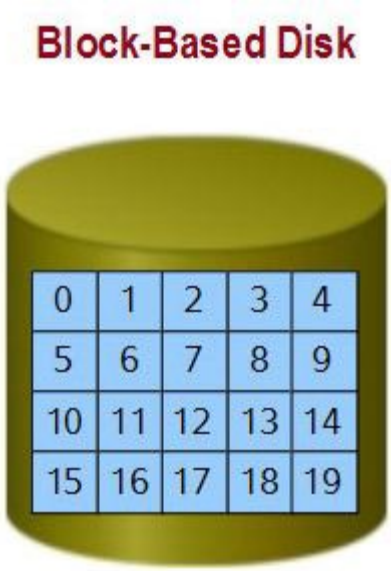
tree file system

This works very well up to a point, but as capacity grows the file model becomes burdensome for two reasons. First, performance suffers beyond a certain capacity. The NAS system itself has the limited processing power, making the processor a bottleneck. The performance also suffers from the massive database—the file lookup tables– that accompany capacity growth.

Block Storage System



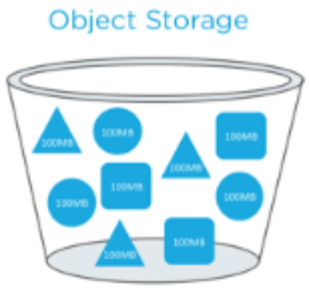
In block storage system raw volumes of storage are created and each block can be controlled as an individual hard disk. Data is organized through block IDs (e.g., sector number) and can be organized as a structure (called a file system) or an application-specific structure. The blocks created are controlled by the server-based operating system and each block can be individually formatted with the required file system and are generally accessed by Fibre Channel, iSCSI or Fibre Channel over Ethernet protocols.



Block level storage is generally deployed in Storage area network (SAN). A client machine connects to a specific volume on the storage service and formats it as if it were a local block device. Block level storage can be used for storing files and can work well for applications like database, Virtual Machine etc. Using traditional file systems on block storage places explicit or practical operational limits on scaling beyond the petabyte range.

Object Storage System

Object storage is a relatively new storage type, designed for unstructured data such as media, documents, logs, backups, application binaries, and VM images. In object storage system the data are stored along with its metadata and unique identifier. This file system doesn't have the hierarchy like the file system or relation to consecutive chunks of hard disk.



Conceptually they are like a persistent key/value store; objects are usually submitted via a REST API call, and an identifier returned. Most object stores allow attaching metadata to objects and aggregating them into containers (or buckets).

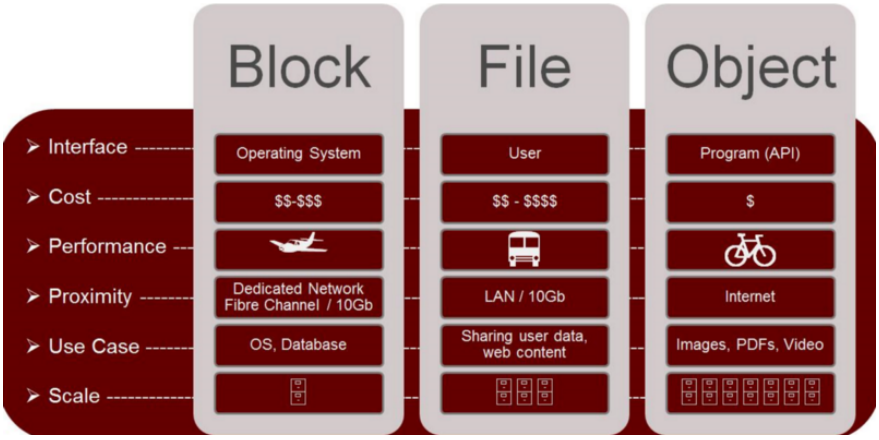
Objects in Object storage are “bundled data” (aka a file) with corresponding metadata. This object gets a unique ID (identifier), that is calculated out of the file content and the metadata.

Applications identify the object via this ID. The many objects inside an object storage system are stored all over the given storage disks. In its pure form, object storage can “only” save one version of a file (object). If a user makes a change another version of the same file is stored as a new object. Because of this reason an object storage is a perfect solution for a backup or archive solution. Or, for example, storage that holds vast amounts of video or movies that are only watched but not changed like for example online movie streaming sites or videos on YouTube.

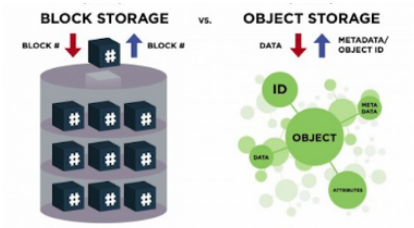
The main difference between the other concepts is that the objects are managed via the application itself that supports Object storage. That means that no real file system is needed here. This layer is obsolete. An application that uses Object storage sends a storage inquiry to the solution where to store the object. The object is then given an address inside the huge storage space and saved there by the application itself.

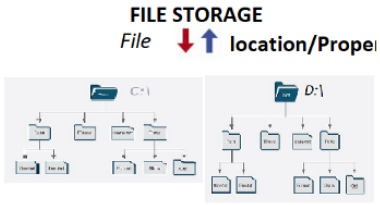
Because of the much simple management of data—with no real file system in place—Object storage solutions can be scaled up much easier than File storage or Block storage-based systems. You just add some disks in the solution and no big management is needed anymore to have more storage space. That´s the main benefit especially in times of exponential data growth.

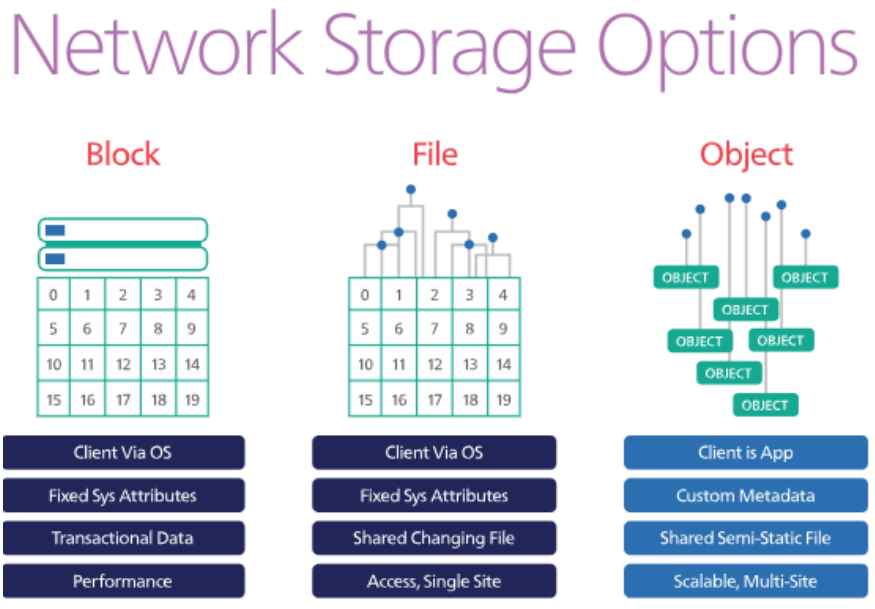
So Object storage is a perfect solution for huge amounts of data and therefore highly used by big cloud service providers like Amazon, Google and others.



Reference: "https://thecustomizewindows.com/wp-content/uploads/2017/09/xObject-Storage-vs-Block-Storage.png.pagespeed.ic.hkN1FTO2ZW.png"







reference: https://www.caringo.com/wp-content/uploads/2017/01/File-Object-Storage.svg

NOW LET'S GET BACK TO GOOGLE !!

There are so many data storage options in Google. One thing to note is that the option you choose also depends on the data you are going to operate. For example, if your data are user input in text and needs to save for a short period of time, creating the bucket could be your solution whereas if you are thinking of database then Cloud SQL could be the solution. Each of the Google solutions has its own significant benefits over the other.

Data could be structured or unstructured, anything from text files, mp3, pictures or database etc.

Google provides

1. **Object storage for different needs and price points**
2. **Block storage for VM's**
3. **File storage for applications that need shared filesystems**
4. **Managed MySQL**
5. **Globally scalable NoSQL**
6. **Archival storage for long-term storage**

Different Storage products from Google

Unstructured Data

1. **Google Cloud Storage**
2. **Cloud Filestore**
3. **Persistent Disk**
4. **Cloud Memorystore**

Structured Data

SQL

1. **Cloud SQL**
2. **Cloud Spanner**

NoSQL

1. **Cloud Bigtable**
2. **Cloud Firestore**
3. **Cloud Datastore**

Let us briefly look at what Google has to say about the storage products so you will have a brief idea about what we are going to learn.

PRODUCT	DESCRIPTION	GOOD FOR	COMMON WORKLOADS
Persistent Disk	Fully-managed, price-performant block storage that is suitable for virtual machines and containers.	• Block storage for Google Compute Engine and Google Kubernetes Engine • Snapshots for data backup	• Disks for virtual machines • Sharing read-only data across multiple virtual machines • Rapid, durable backups of running virtual machines
Google Cloud Storage	A scalable, fully-managed, highly reliable, and cost-efficient object / blob store.	• Images, pictures, and videos • Objects and blobs • Unstructured data	• Storing and streaming multimedia • Storage for custom data analytics pipelines • Archive, backup, and disaster recovery
Google Cloud Bigtable	A scalable, fully-managed NoSQL wide-column database that is suitable for both real-time access and analytics workloads.	• Low-latency read/write access • High-throughput analytics • Native time series support	• IoT, finance, adtech • Personalization, recommendations • Monitoring • Geospatial datasets • Graphs
Google Cloud Datastore	A scalable, fully-managed NoSQL document database for your web and mobile applications.	• Semi-structured application data • Hierarchical data • Durable key-value data	• User profiles • Product catalogs • Game state
Google Cloud SQL	A fully-managed MySQL and PostgreSQL database service that is built on the strength and reliability of Google's infrastructure.	• Web frameworks • Structured data • OLTP workloads	• Websites, blogs, and content management systems (CMS) • Business Intelligence (BI) applications • ERP, CRM, and eCommerce applications • Geospatial applications
Google Cloud Spanner	Mission-critical, relational database service with transactional consistency, global scale and high availability.	• Mission-critical applications • High transactions • Scale + Consistency requirements	• Adtech • Financial services • Global supply chain • Retail
Google BigQuery	A scalable, fully-managed Enterprise Data Warehouse (EDW) with SQL and fast response times.	• OLAP workloads up to petabyte-scale • Big Data exploration and processing • Reporting via Business Intelligence (BI) tools	• Analytical reporting on large data • Data Science and advanced analyses • Big Data processing using SQL

References:https://cloud.google.com/storage-options/

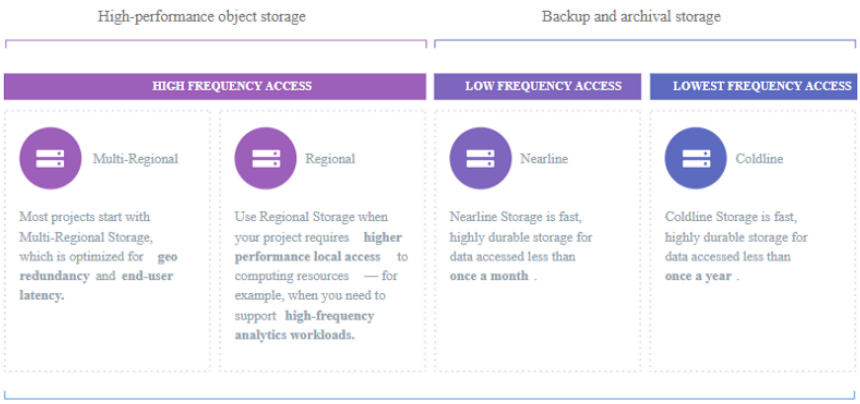
We will discuss all of these options and use cases one by one

1. Google Cloud Storage

Google Cloud Storage

Google cloud storage is the most widely used storage options in GCP. It is Object-based storage and highly accessible in different ways. It Integrates into your app with API. The price depends on the storage class.

Data stored is highly secured and durable with zero carbon emission. The storage is divided into 4 classes and two storage types.



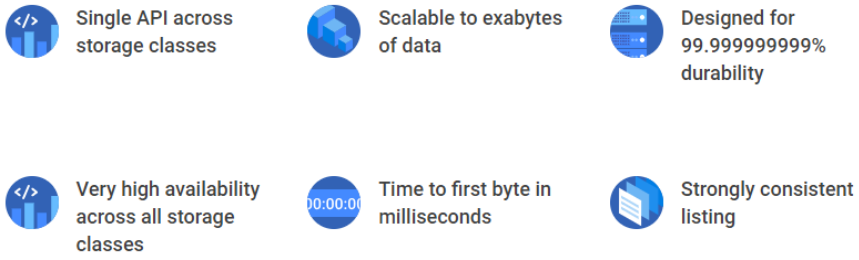
reference: https://cloud.google.com/images/storage/storage-classes-desktop.svg

Storage Class Comparison

Storage Class	Multi-Regional	Regional	Nearline	Coldline
Price	$0.026 - $0.036 per GB/month	$0.02 - $0.035 per GB/month	$0.01 - $0.02 per GB/month	$0.007 - $0.014 per GB/month
Good for	For highest availability of frequently accessed data	For data accessed frequently within a region	For data accessed less than once a month	Data accessed less than once a year
Redundancy	Geo-redundant	Regional, redundant across availability zones	Regional	Regional
Availability	99.95% SLA	99.9% SLA	99% SLA	99% SLA
Durability	99.999999999%			
Responsiveness	Millisecond access consistent API			
Use cases	Content storage and delivery, business continuity	Store data and run data analytics within a region	Store infrequently accessed content	Archive storage, backup and recovery
Applications	Video, Multimedia, Business continuity	Transcoding, Data analytics, Compute intensive data processing	Backup long-tail content, Rarely accessed docs	Archive source file backup, Disaster recovery

References: https://cloud.google.com/storage/features/

Key Features



references: https://cloud.google.com/products/storage/

Use cases

1. Hosting music and videos
2. Hosting images and website content
3. Mobile apps development

4. Video transcoding
5. Genomics
6. General data analytics and compute
7. Archival data
8. Multimedia backup storage
9. Disaster recovery

Google says:{

Choosing a region and zone

Certain Compute Engine resources live in regions or zones. A region is a specific geographical location where you can run your resources. Each region has one or more zones; most regions have three or more zones. For example, the us-central1 region denotes a region in the Central United States that has zones us-central1-a, us-central1-b, us-central1-c, and us-central1-f.

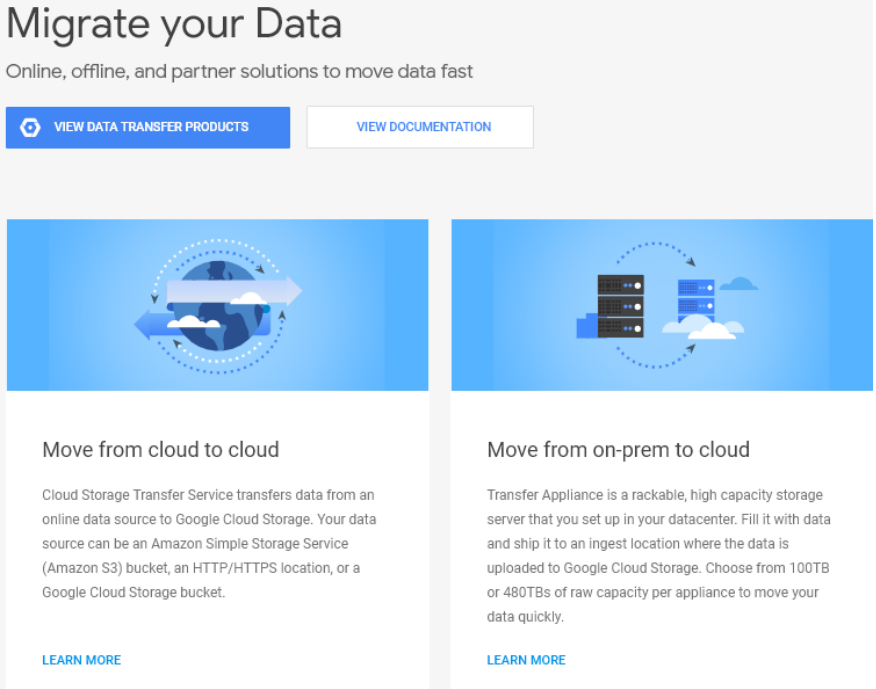
Resources that live in a zone, such as instances or persistent disks, are referred to as zonal resources. Other resources, like static external IP addresses, are regional. Regional resources can be used by any resources in that region, regardless of zone, while zonal resources can only be used by other resources in the same zone.

For example, disks and instances are both zonal resources. To attach a disk to an instance, both resources must be in the same zone. Similarly, if you want to assign a static IP address to an instance, the instance must be in the same region as the static IP.

You choose which region or zone hosts your resources, which controls where your data is stored and used. Choosing a region and zone is important for several reasons:

Handling failuresDistribute your resources across multiple zones and regions to tolerate outages. Google designs zones to be independent of each other: a zone usually has power, cooling, networking, and control planes that are isolated from other zones, and most single failure events will affect only a single zone. Thus, if a zone becomes unavailable, you can transfer traffic to another zone in the same region to keep your

services running. Similarly, if a region experiences any disturbances, you should have backup services running in a different region. For more information about distributing your resources and designing a robust system, see Designing Robust Systems. Decreased network latency decrease network latency, you might want to choose a region or zone that is close to your point of service. For example, if you mostly have customers on the East Coast of the US, then you might want to choose a primary region and zone that is close to that area and a backup region and zone that is also close by.}



References: https://cloud.google.com/storage/migrate/

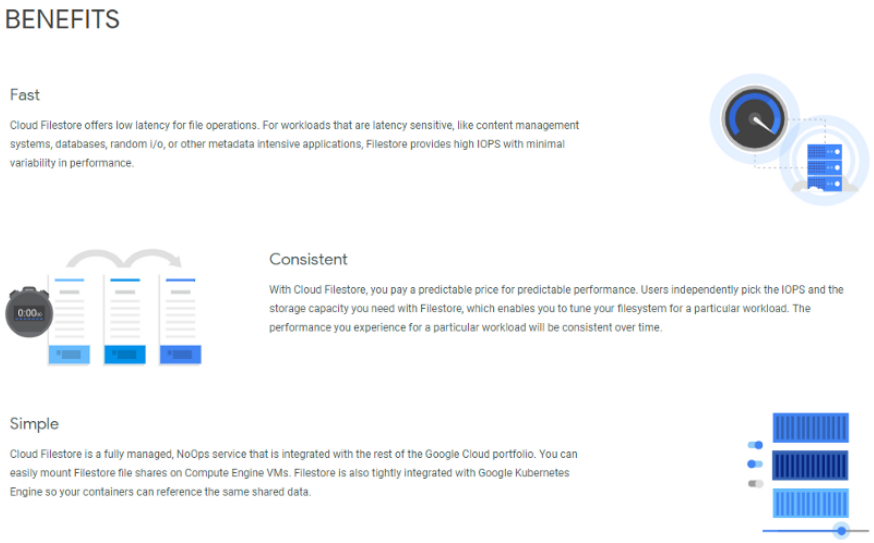
Object lifecycle

Object lifecycles refer to what to do with the data once it is old or doesn't require frequently. There are several options to select like storing in nearline or cold line or delete it after custom time.

2. Cloud Filestore—Beta

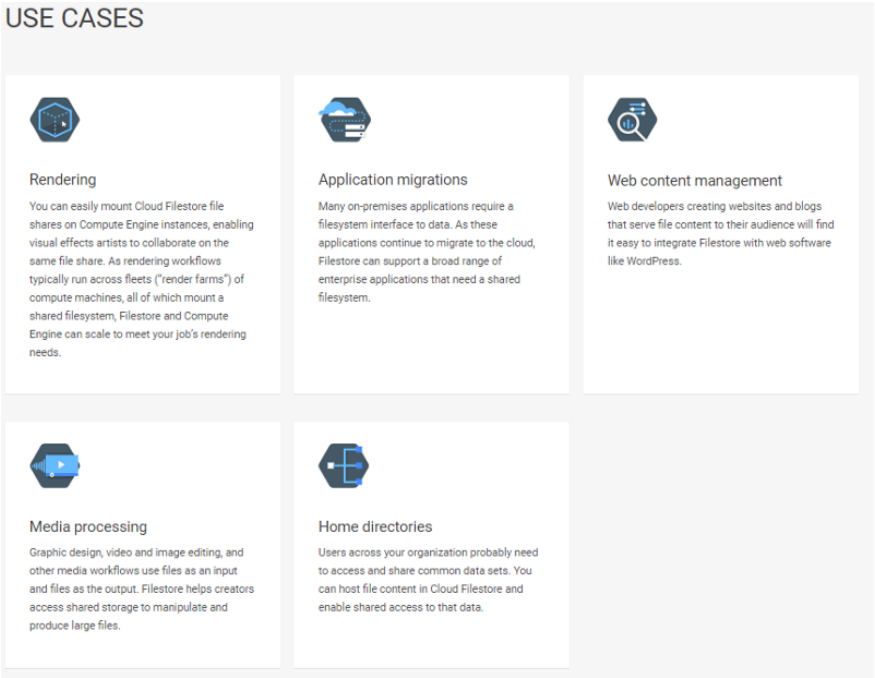
cloud Filestore

The name itself filestore refers to a managed file storage service for the application that requires a filesystem interface and a shared file system for data. It gives native experience for using Network attached storage(NAS) with Google Compute Engine.



References: https://cloud.google.com/filestore/

Create and manage Cloud Filestore instances by using the GCP console or the command-line tool, and interact with the NFS file share on the instance by using standard operating system commands.



References: https://cloud.google.com/filestore/

3. Cloud SQL

Cloud SQL

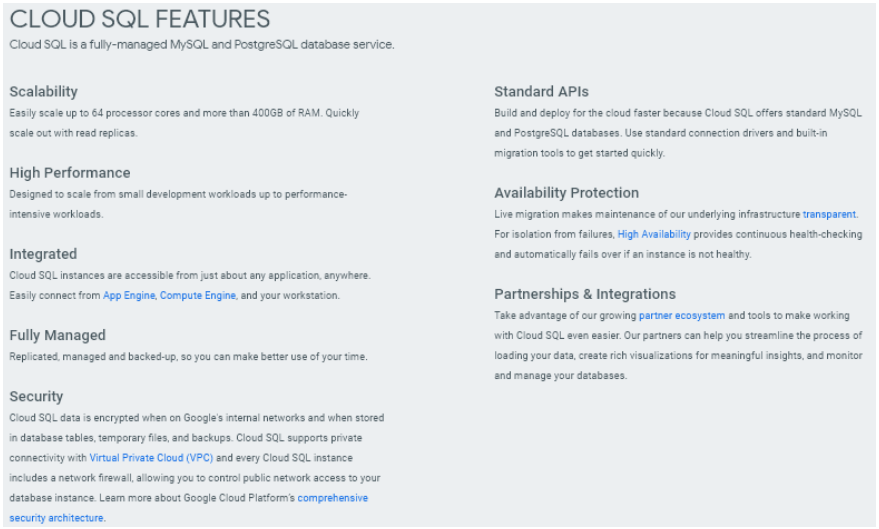
Cloud SQL is a fully managed database service that makes it easy to set up, maintain, manage and administer your relational database. Cloud SQL supports

1. PostgreSQL and
2. MySQL

It offers high performance, scalable and convenience for users to create and implement a database for their applications. A database is

structured data it will be hosted in Google cloud but can be accessed from anywhere.

References: https://cloud.google.com/sql/



References: https://cloud.google.com/sql/

Cloud SQL Pricing depends on Regions and Zones of the GCP

Use cases

Cloud SQL can be used for managing

PRODUCT	DESCRIPTION	GOOD FOR	COMMON WORKLOADS
Google Cloud SQL	A fully-managed MySQL and PostgreSQL database service that is built on the strength and reliability of Google's infrastructure.	• Web frameworks • Structured data • OLTP workloads	• Websites, blogs, and content management systems (CMS) • Business Intelligence (BI) applications • ERP, CRM, and eCommerce applications • Geospatial applications

4. Cloud Bigtable

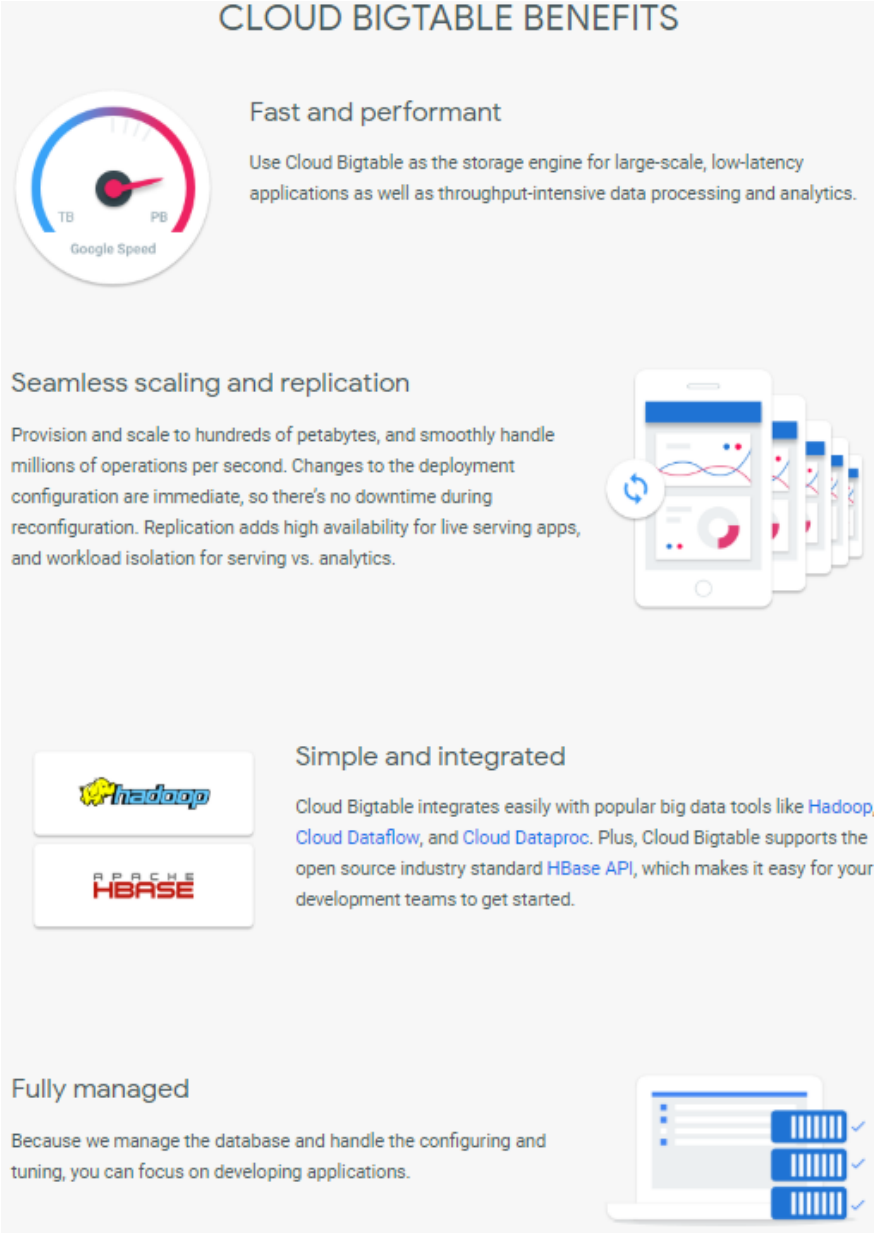
Cloud Bigtable

Cloud Bigtable is a fully managed NoSQL database service that can scale up to petabyte for analytical and operational workloads. It has

very low latency as small as 10ms. It has replication option in case of zonal failures.

The most widely used area for Bigtable is Machine Learning and AI.

It is easily integrable with current Big data tools like Hadoop and HBase.



Reference: https://cloud.google.com/bigtable/

Use Cases

Cloud Bigtables are used in Ad tech, Fintech, and IoT

5.Cloud Spanner

Cloud Spanner

Cloud Spanner is a fully managed, mission-critical relational database system. It has special properties like

1. High Availability
2. Transactional consistency
3. global scale

With traditional relational semantics like schemas, ACID transaction, SQL with automatic and synchronous replication cloud spanner is the only product of its kind in the market.

Cloud spanner is first horizontally scalable. strongly consistent, relational database service.

Cloud Spanner is the only enterprise-grade, globally-distributed, and strongly consistent database service built for the cloud specifically to combine the benefits of a relational database structure with non-relational horizontal scale. This combination delivers high-performance transactions and strong consistency across rows, regions, and continents with an industry-leading 99.999% availability SLA, no planned downtime, and enterprise-grade security. Cloud Spanner revolutionizes database administration and management and makes application development more efficient.

In today's always-on, globally-distributed world, IT and developer efficiency, measured in the app downtime and time to market, is one of an organization's most precious resources. The challenge of efficiently managing app database backends while at the same time giving developers the tools they need to build efficiently was previously a challenge.

Cloud Spanner: The best of the relational and non-relational worlds

	CLOUD SPANNER	TRADITIONAL RELATIONAL	TRADITIONAL NON-RELATIONAL
Schema	✓ **Yes**	✓ Yes	✗ No
SQL	✓ **Yes**	✓ Yes	✗ No
Consistency	✓ **Strong**	✓ Strong	✗ Eventual
Availability	✓ **High**	✗ Failover	✓ High
Scalability	✓ **Horizontal**	✗ Vertical	✓ Horizontal
Replication	✓ **Automatic**	↻ Configurable	↻ Configurable

References: https://cloud.google.com/spanner/

Use Cases

Customers across industries can use Cloud Spanner to deliver value to their customers:

USE CASE	BEFORE CLOUD SPANNER	WITH CLOUD SPANNER
Financial trading	Inconsistencies lead to potential monetary loss during reconciliation. Global synchronous replication of trades is not feasible.	Cost savings and a consistent, unified, global view.
Insurance	Inconsistencies lead to incomplete views of customers.	Up-to-date customer views provide more accurate, real-time data.
Global call centers	Eventual and out-of-touch.	Real-time and up-to-date.
Supply-chain management and manufacturing	Global supply chain presents an inconsistent global view and/or data must be shipped in batches.	Global, real-time, consistent view enables real-time decision making.
Telecom and billing	Processing capacity limited to finite scale-up compute resources.	Scale-out allows improved processing speed.
Logistics and Transportation	Regional reach with many systems glued together.	Global reach with lower latency and a consistent view.
Gaming	Each server or cluster is its own universe.	Consistent, global view delivers a unified experience.
E-Commerce (High Availability)	Limited availability SLA or no SLA guarantees. In practice, potential missed sales.	Guaranteed max of 5 minutes of downtime (including planned downtime) on paper and in practice.

References: https://cloud.google.com/spanner/

CLOUD SPANNER FEATURES

The first horizontally scalable, globally consistent, relational database service

Global Scale
Horizontally scalable across rows, regions, and continents, from 1 to hundreds or thousands of nodes.

Fully Managed
Ease of deployment at every scale and every stage. Synchronous replication and maintenance are automatic and built-in.

Relational Semantics
Everything you would expect from a relational database—schemas, ACID transactions, and SQL queries (ANSI 2011).

Multi-Language Support
Client libraries in C#, Go, Java, Node.js, PHP, Python, and Ruby. JDBC driver for connectivity with popular third-party tools.

Transactional Consistency
Purpose-built for external, strong, global transactional consistency.

Enterprise Grade Security
Data-layer encryption, IAM integration for access and controls, and audit logging.

Highly Available
Whenever, wherever, your data is highly available.

References: https://cloud.google.com/spanner/

6. Cloud Firestore—Beta

Cloud Firestore

Cloud Firestore is a fast, fully managed, serverless, cloud-native NoSQL document database that simplifies storing, syncing and querying data for mobile, web, and IoT apps at global scale. It's client libraries provide live synchronization and offline support, as well as its security features and integrations with firebase and GCP, accelerates building truly serverless apps

Build serverless apps

Cloud Firestore in Datastore mode is fully compatible with your Cloud Datastore applications. It delivers massive improvements for Datastore users—up to 99.999% availability SLA, no more eventual consistency, no more entity group limits on writes per second, and no more cross-entity group transaction limits.

You can talk directly to Cloud Firestore from your mobile or web clients for a truly serverless solution. No need to set up an intermediary server to manage access to your data. This is great for prototyping, iterating, and getting a production system up and running quickly. Cloud Firestore is a cloud-native database, which provides an automatically scaling solution built from the ground up to take advantage of Google Cloud Platform's powerful infrastructure.

Sync data across devices, on or offline

With Cloud Firestore, your applications can be updated in near real time when data on the backend changes. This is not only great for

building collaborative multi-user mobile applications but also means you can keep your data in sync with individual users who might want to use your app from multiple devices.

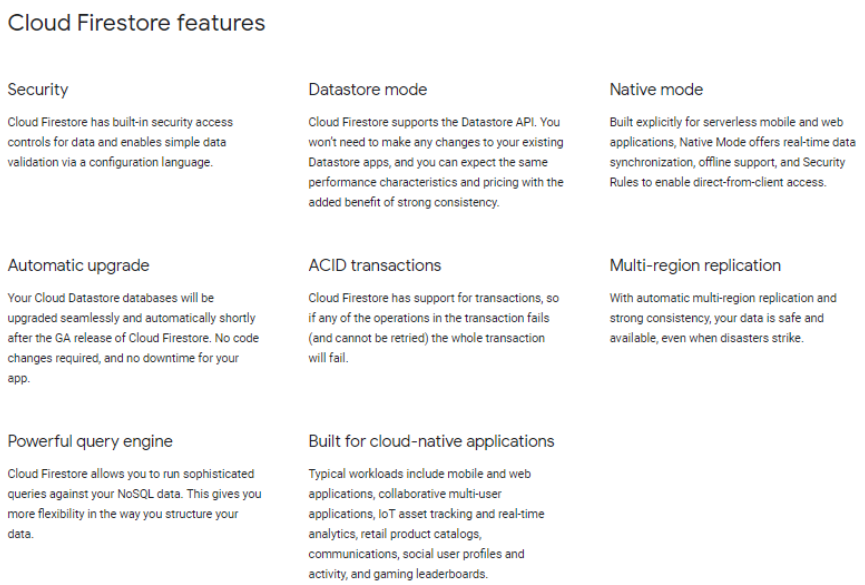
Cloud Firestore has full offline support, so you can access and make changes to your data, and those changes will be synced to the cloud when the client comes back online. Built-in offline support leverages local cache to serve and store data, so your app remains responsive regardless of network latency or internet connectivity.

Simple and effortless

Cloud Firestore's robust client libraries make it easy for you to update and receive new data while worrying less about establishing network connections or unforeseen race conditions. It can scale effortlessly as your app grows. Cloud Firestore allows you to run sophisticated queries against your data. This gives you more flexibility in the way you structure your data and can often mean that you have to do less filtering on the client, which keeps your network calls and data usage more efficient.

Enterprise-grade, scalable NoSQL

Cloud Firestore is a fast and fully managed NoSQL cloud database. It is built to scale and takes advantage of GCP's powerful infrastructure, with automatic horizontal scaling in and out, in response to your application's load. Security access controls for data are built in and enable you to handle data validation via a configuration language.



References: https://cloud.google.com/firestore/

7. Cloud Datastore

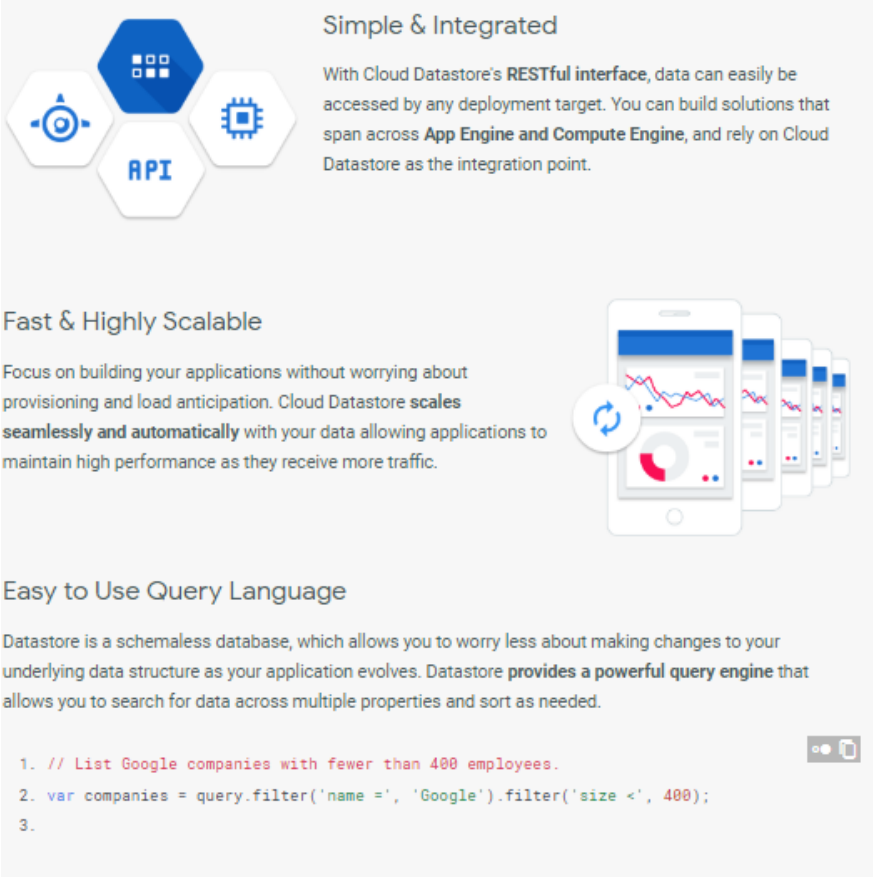
Cloud Datastore

Cloud Datastore is a highly scalable NoSQL database for your web and mobile applications.

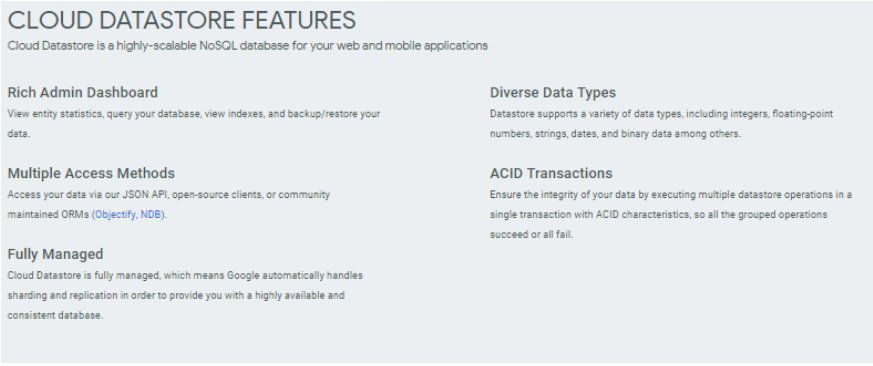
Google Cloud Datastore gives you an elastic, highly available document-oriented database as a service. It is fully managed so you don't need to deploy, update, configure, or manage your database solution. Cloud Datastore comes with a rich admin dashboard, a powerful query engine as well as multiple methods to access the database which makes it ideal for mobile and web workloads.

Highly Scalable NoSQL Database

Cloud Datastore is a highly-scalable NoSQL database for your applications. Cloud Datastore **automatically handles sharding and replication**, providing you with a highly available and durable database that scales automatically to handle your applications' load. Cloud Datastore provides a myriad of capabilities such as **ACID transactions, SQL-like queries, indexes and much more**.



References: https://cloud.google.com/datastore/

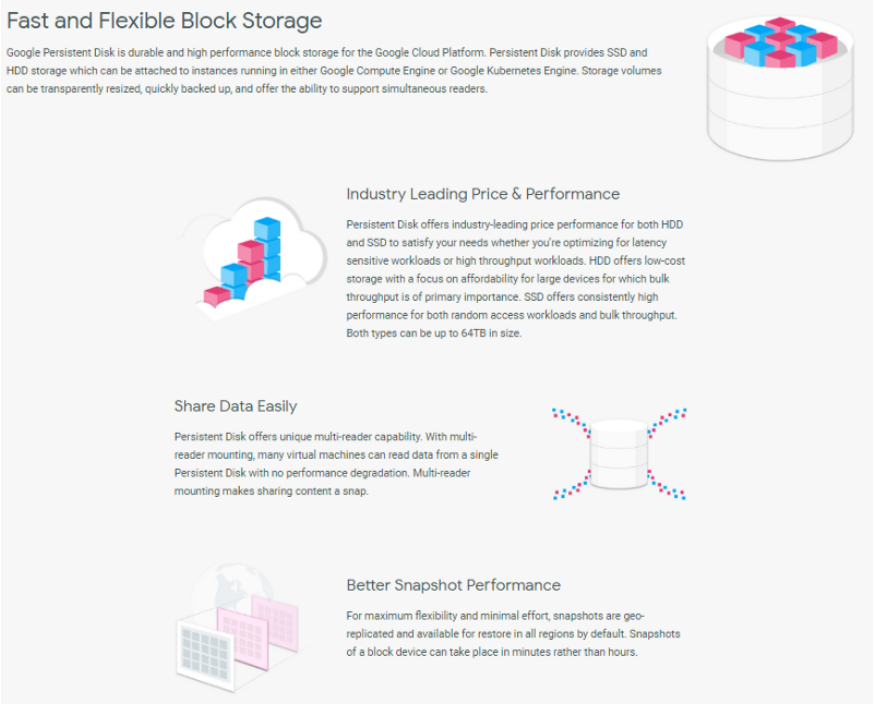


references: https://cloud.google.com/datastore/

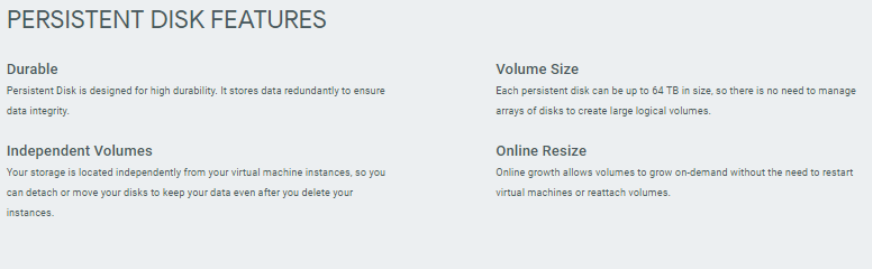
8. Persistent Disk

Persistent Disk

Persistent Disk is a high-performance block storage service suitable for Virtual Machines and container storage. It offers unmatched price/performance. You only pay for capacity and you are never charged for provisioned IOPS. Additionally, Persistent Disk offers multi-reader mounts and on-demand volume resizing to simplify operations.



references: https://cloud.google.com/persistent-disk/



References: https://cloud.google.com/persistent-disk/

9. Cloud Memorystore

Cloud Memstore

Cloud Memorystore is a fully managed in-memory data store service built on scalable, more secure, and highly available infrastructure managed by Google. Use Cloud Memorystore to build application caches that provide sub-millisecond data access. Cloud Memorystore is compatible with the Redis protocol, allowing easy migration with zero code change.

Focus on Building Great Apps

Spend more time writing code and less time managing by taking advantage of the power of open source Redis. Cloud Memorystore automates complex tasks like enabling high availability, failover, patching, and monitoring so that you can spend more time coding.

Scale as Needed

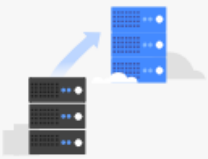
With Cloud Memorystore for Redis, you can easily achieve the sub-millisecond latency and throughput your applications need. Start with the lowest tier and smallest size, and then grow your instance effortlessly with minimal impact to application availability. Cloud Memorystore can support instances up to 300 GB and network throughput of 12 Gbps.

Highly Available

High availability instances are replicated across two zones and provide a 99.9% availability SLA. Instances are monitored constantly, and with automatic failover, applications experience minimal disruption.

Google Grade Security

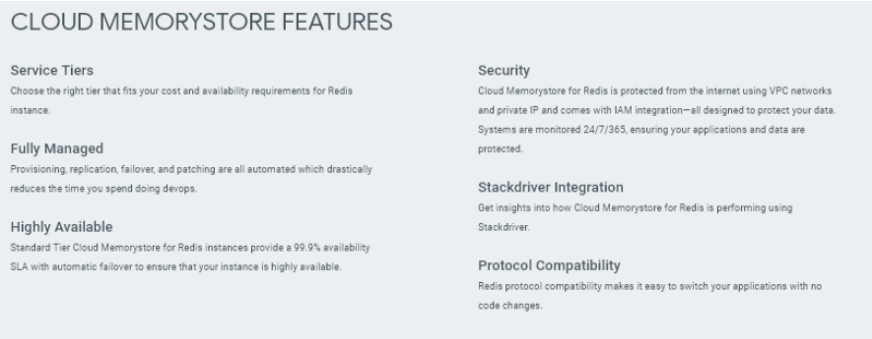
Cloud Memorystore instances are isolated and protected from the internet using private IPs and are further secured using IAM role-based access control.



Easy Lift and Shift

Cloud Memorystore for Redis is fully compatible with the Redis protocol. You can lift and shift your applications from open source Redis to Cloud Memorystore without any code changes. There is no need to learn new tools since all existing tools and client libraries just work.

References: https://cloud.google.com/memorystore/

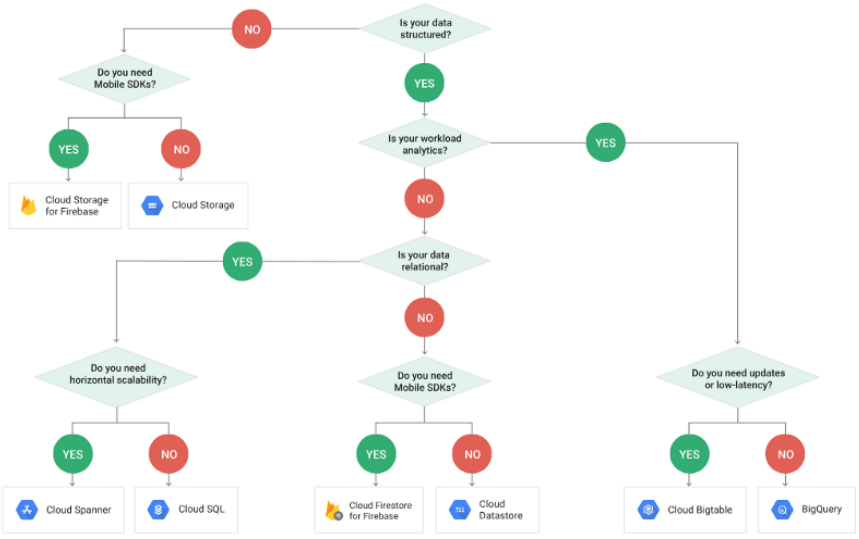


References: https://cloud.google.com/memorystore/

FINALLY, CHOOSING A STORAGE OPTION

Different applications or website or apps have different requirements and different platforms in which they work. Performance requirements are one of the important things to consider while choosing the solution. Google has as we have seen from the above a wide range of storage options whether it is structured or unstructured data.

To determine which solution to take Google has built a flow diagram which will lead to one of the storage options.



Reference: https://cloud.google.com/images/storage-options/flowchart.svg

In this way, we can choose one solution from a large set of options.

See you in 2.4!