Google Cloud

Choosing Storage and Data Solutions

Priyanka Vergadia
Developer Advocate, Google Cloud

In this module, we discuss Google Cloud storage and data solutions and how to select the most suitable one to meet your business and technical requirements.

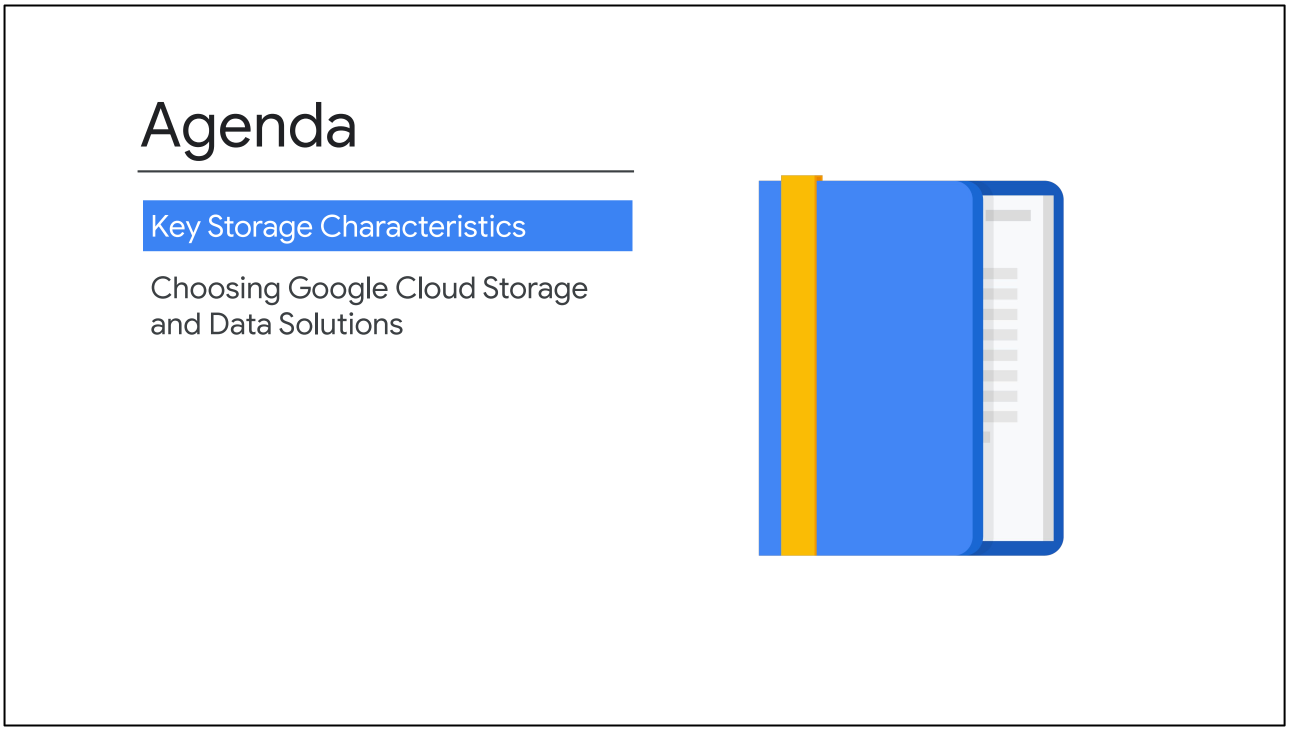
Learning objectives

- Choose the appropriate Google Cloud data storage service based on use case, durability, availability, scalability, and cost.
- Store binary data with Cloud Storage.
- Store relational data using Cloud SQL and Spanner.
- Store NoSQL data using Firestore and Bigtable.
- Cache data for fast access using Memorystore.
- Aggregate data for queries and reports using BigQuery as a data warehouse.

Google Cloud provides a rich set of different storage options that cater to different types of data, sizes of data, lifecycles, and also data access patterns.

We will discuss storing binary data with Cloud Storage, relational data with Cloud SQL or Spanner, and NoSQL or unstructured data using Firestore and Bigtable.

In addition, we will consider caching for fast data access using Memorystore and finally aggregating data for queries and reports using BigQuery as a data warehouse.

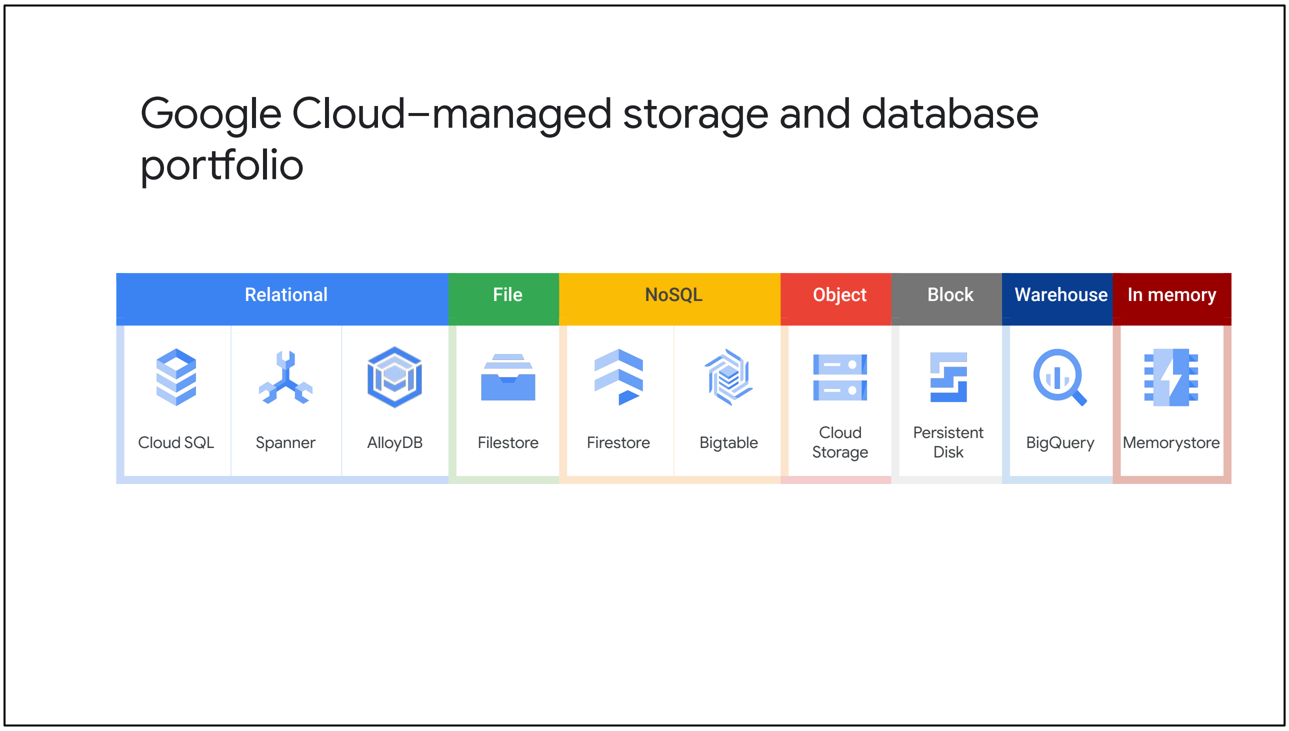


Agenda

Key Storage Characteristics

Choosing Google Cloud Storage and Data Solutions

Let’s get started by considering key storage characteristics.

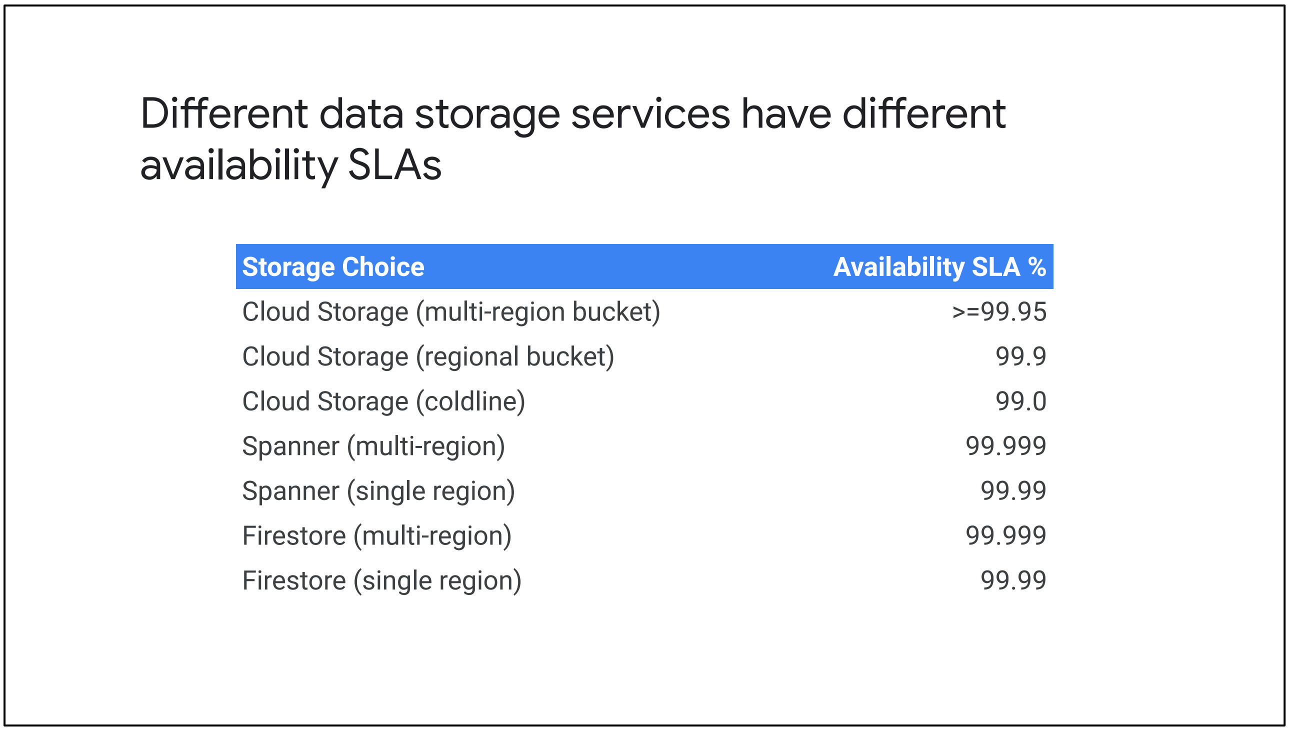


Google Cloud–managed storage and database portfolio

Relational			File	NoSQL		Object	Block	Warehouse	In memory
Cloud SQL	Spanner	AlloyDB	Filestore	Firestore	Bigtable	Cloud Storage	Persistent Disk	BigQuery	Memorystore

Google Cloud has a wide range of managed storage and database options in its portfolio. Knowing the characteristics of each and being able to select a suitable solution is vital as an architect during the design process. From a high level, the services range from relational, NoSQL object storage, data warehouse, to in-memory. These services are fully managed, scalable, and backed by industry leading-SLAs.

Making a decision on which storage solution is right for your requirement is a balance of a number of characteristics, including: type of data, scale, durability, availability, and location requirements. We'll discuss ways in which you can make the best decision based on your requirements in this module.



Different data storage services have different availability SLAs

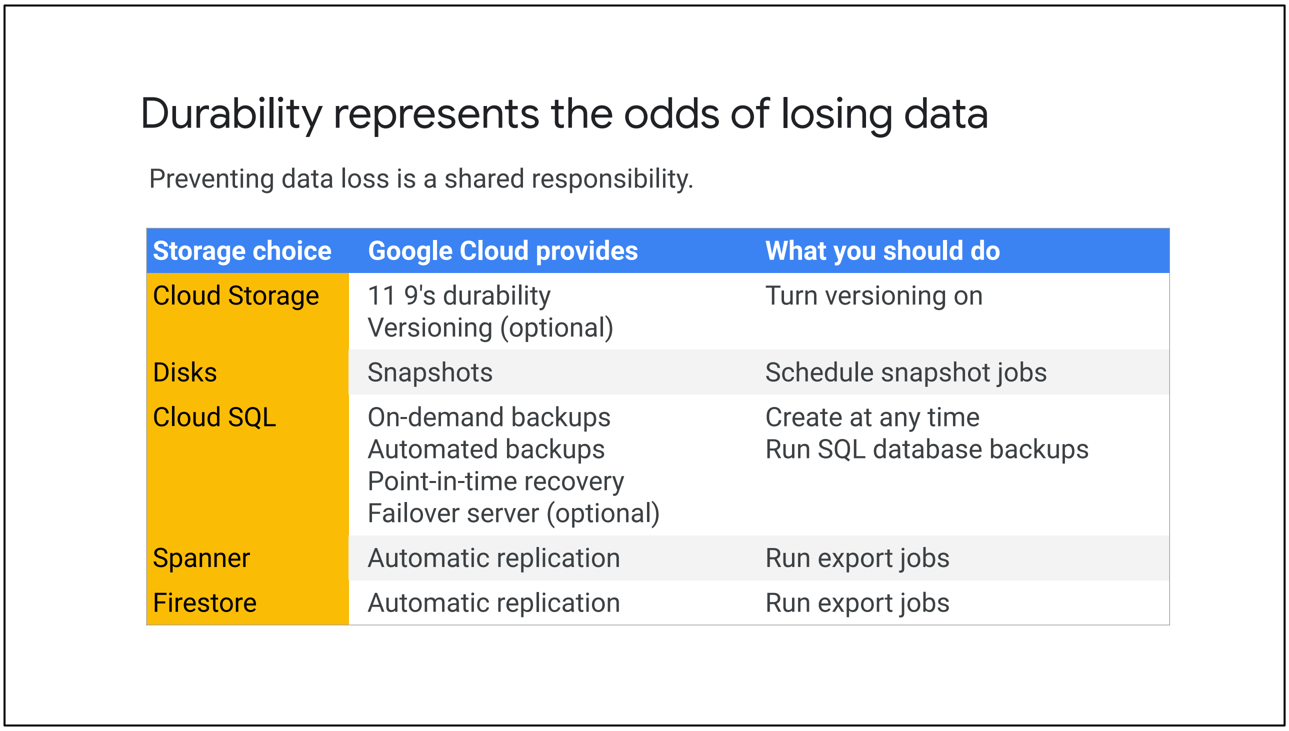
Storage Choice	Availability SLA %
Cloud Storage (multi-region bucket)	>=99.95
Cloud Storage (regional bucket)	99.9
Cloud Storage (coldline)	99.0
Spanner (multi-region)	99.999
Spanner (single region)	99.99
Firestore (multi-region)	99.999
Firestore (single region)	99.99

Different data storage services have different availability SLAs. For a service, the availability SLA is often dependent on the configuration of the service. For example, for Cloud Storage as the slide shows, the availability varies depending on whether multi-regional, regional, or coldline buckets are created.

The same can be seen for Spanner and Firestore, with multi-regional offering higher availability than single region configurations. This is where requirements are extremely important as they will help inform the storage choice.

The availability SLAs are typically defined per month. Monthly Uptime Percentage means the total number of minutes in a month, minus the number of minutes of downtime suffered from all downtime periods in a month, divided by the total number of minutes in a month.

For up-to-date SLA numbers, refer to the documentation.



Durability represents the odds of losing data

Preventing data loss is a shared responsibility.

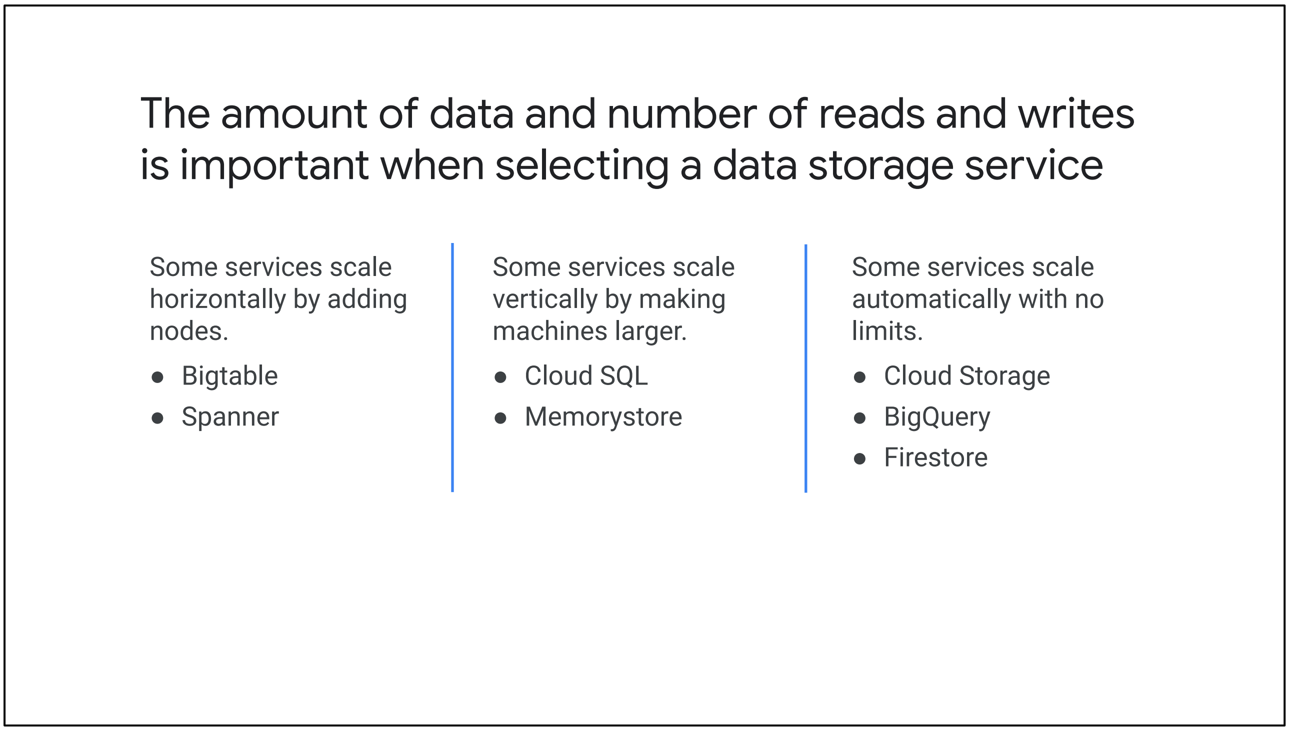
Storage choice	Google Cloud provides	What you should do
Cloud Storage	11 9's durability Versioning (optional)	Turn versioning on
Disks	Snapshots	Schedule snapshot jobs
Cloud SQL	On-demand backups Automated backups Point-in-time recovery Failover server (optional)	Create at any time Run SQL database backups
Spanner	Automatic replication	Run export jobs
Firestore	Automatic replication	Run export jobs

Durability of data represents the odds of losing the data. Depending on the storage solution, the durability is a shared responsibility. Google Cloud's responsibility is to ensure that data is durable in the event of a hardware failure. Your responsibility is performing backups of your data.

For example, Cloud Storage provides you with 11 9's durability, and versioning is a feature. However, it's your responsibility to determine when to use versioning. It is recommended that you turn versioning on and have older versions archived as part of an object lifetime management policy.

For other storage services to achieve durability, it usually means taking backups of data. For disks, this means snapshots, so snapshot jobs should be scheduled. For Cloud SQL, you can create a backup at any time (on-demand). This could be useful if you are about to perform a risky operation on your database, or if you need a backup and you do not want to wait for the backup window. Google Cloud also provides automated backups, point in time recovery, and optionally a failover server. You can create on-demand backups for any instance, whether the instance has automatic backups enabled or not. To improve durability, SQL database backups should also be run.

Spanner and Firestore provide automatic replication, and you should run export jobs with the data being exported to Cloud Storage.



The amount of data and number of reads and writes is important when selecting a data storage service

Some services scale horizontally by adding nodes.

- Bigtable
- Spanner

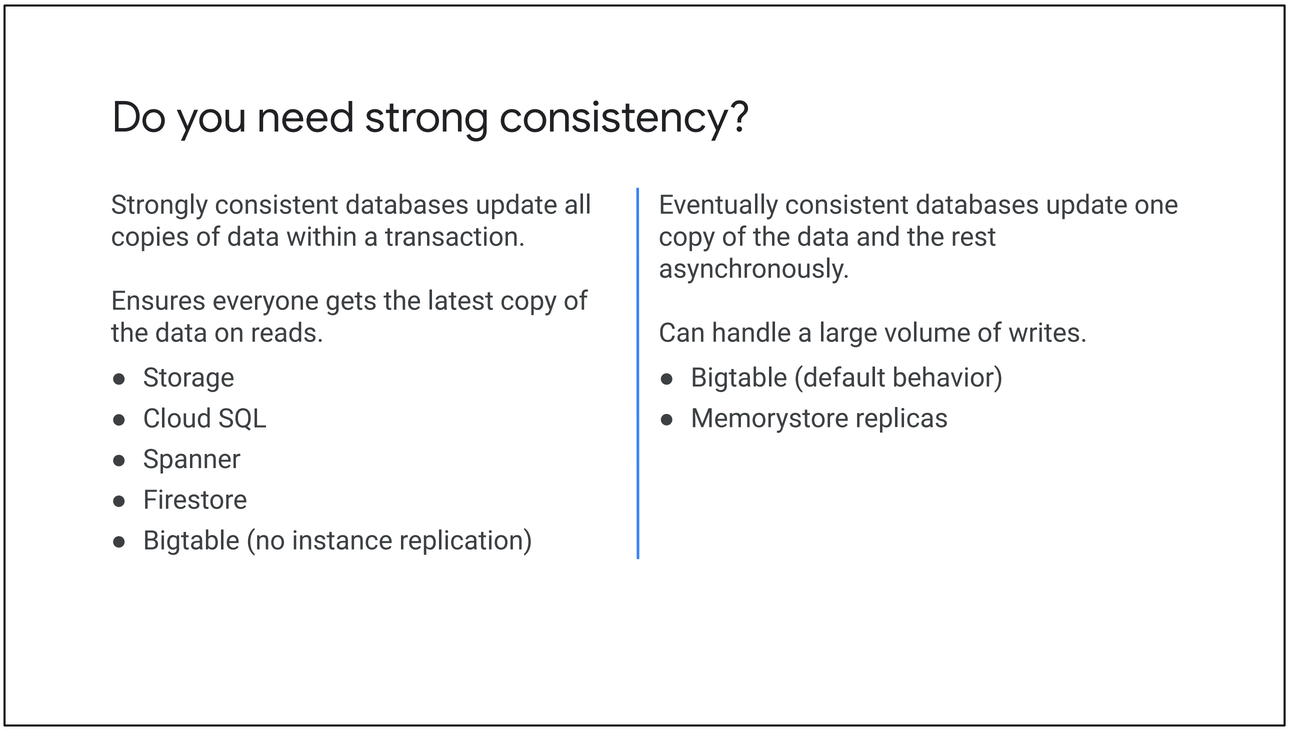
Some services scale vertically by making machines larger.

- Cloud SQL
- Memorystore

Some services scale automatically with no limits.

- Cloud Storage
- BigQuery
- Firestore

The amount of data, and the number of reads and writes, are important to know when selecting a data storage service. Some services scale horizontally by adding nodes; for example, Bigtable and Spanner, which is in contrast to Cloud SQL and Memorystore, which scale machines vertically. Other services scale automatically with no limits; for example, Cloud Storage, BigQuery, and Firestore.



Do you need strong consistency?

Strongly consistent databases update all copies of data within a transaction.

Ensures everyone gets the latest copy of the data on reads.

- Storage
- Cloud SQL
- Spanner
- Firestore
- Bigtable (no instance replication)

Eventually consistent databases update one copy of the data and the rest asynchronously.

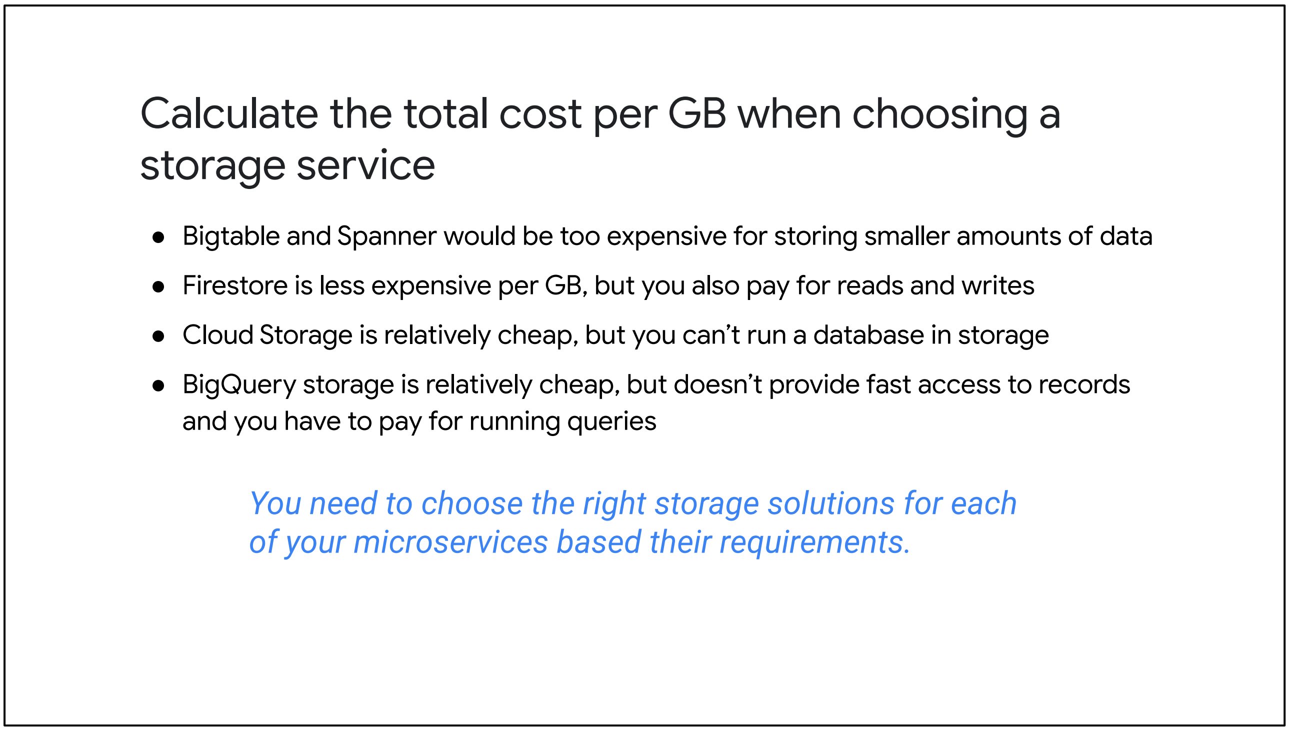
Can handle a large volume of writes.

- Bigtable (default behavior)
- Memorystore replicas

Strong consistency is another important characteristic to consider when designing data solutions. A strongly consistent database will update all copies of data within a transaction and ensure that everybody gets the latest copy of commited data on reads. Google Cloud services providing strong consistency include Cloud Storage, Cloud SQL, Spanner, and Firestore. If an instance does not use replication, Bigtable provides strong consistency, because all reads and writes are sent to the same cluster.

Eventually consistent databases typically have multiple copies of the same data for performance and scalability. They support handling large volumes of writes. They operate by updating one copy of the data synchronously and all copies asynchronously, which means that not all readers are guaranteed to read the same values at a given point in time. The data will eventually become consistent, but not immediately. Bigtable and Memorystore are examples of Google Cloud data services that have eventual consistency.

Replication for Bigtable is eventually consistent by default. This term means that when you write a change to one cluster, you will eventually be able to read that change from the other clusters in the instance, but only after the change is replicated between the clusters.



Calculate the total cost per GB when choosing a storage service

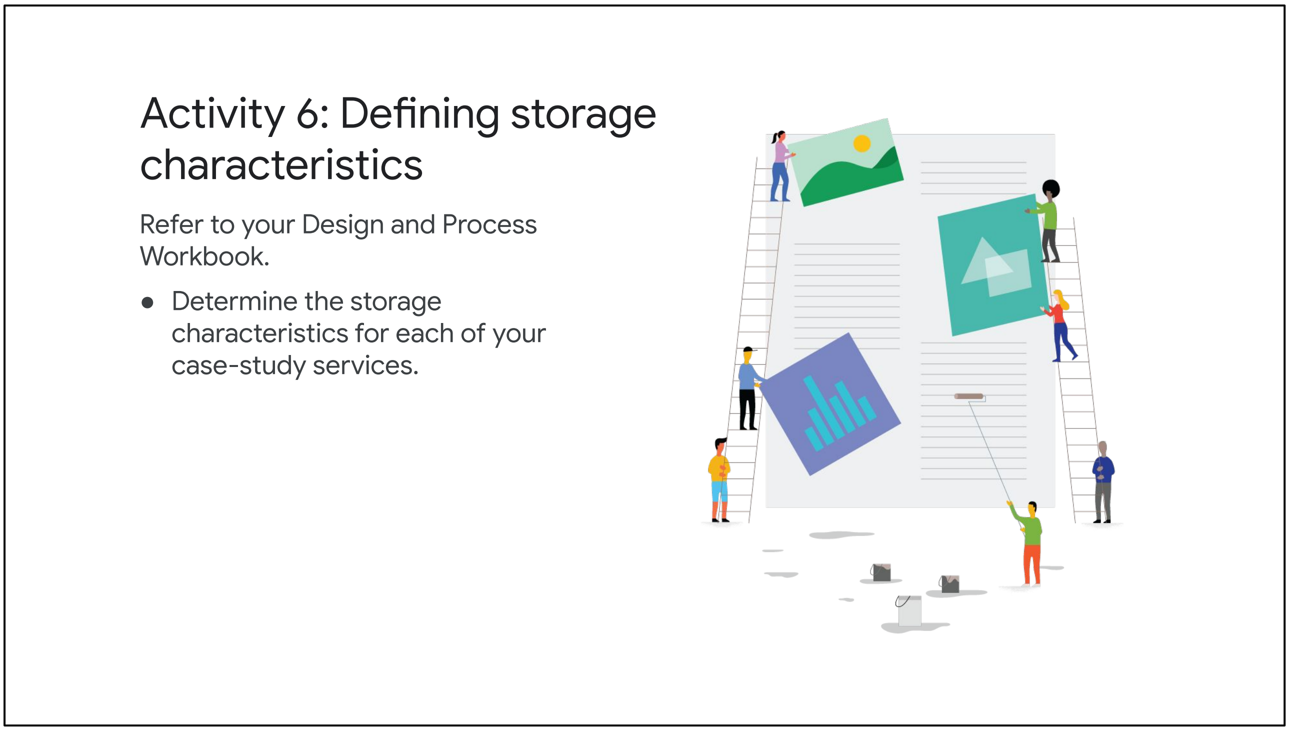
- Bigtable and Spanner would be too expensive for storing smaller amounts of data
- Firestore is less expensive per GB, but you also pay for reads and writes
- Cloud Storage is relatively cheap, but you can't run a database in storage
- BigQuery storage is relatively cheap, but doesn't provide fast access to records and you have to pay for running queries

You need to choose the right storage solutions for each of your microservices based their requirements.

When designing a data storage solution, calculating the total cost per GB is important to help determine the financial implications of a choice.

- Bigtable and Spanner are designed for massive data sets and are not as cost-efficient for small data sets.
- Firestore is less expensive per GB stored, but the cost for reads and writes must be considered.
- Cloud Storage is not as expensive but is only suitable for certain data types.
- BigQuery storage is relatively cheap but does not provide fast access to records, and a cost is incurred for each query.

So as you see, the choice of the right storage solution is not simple. It has to be based on type of data, size of data, and read/write patterns.



Activity 6: Defining storage characteristics

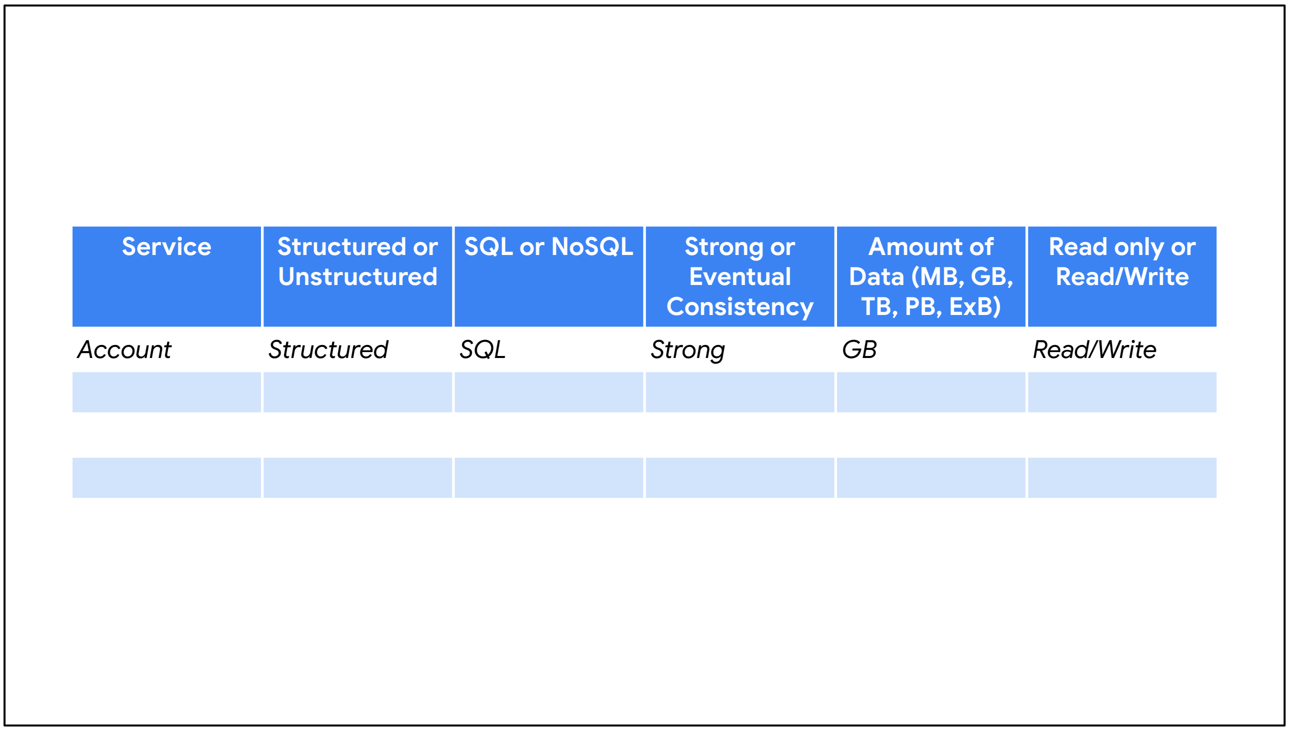
Refer to your Design and Process Workbook.

- Determine the storage characteristics for each of your case-study services.

You will now define storage characteristics for each of your case study services.

In a microservice architecture, there is not one big database to store everything for the entire application. Each service should maintain its own data. Storage characteristics include whether the data is structured or unstructured and relational or NoSQL.

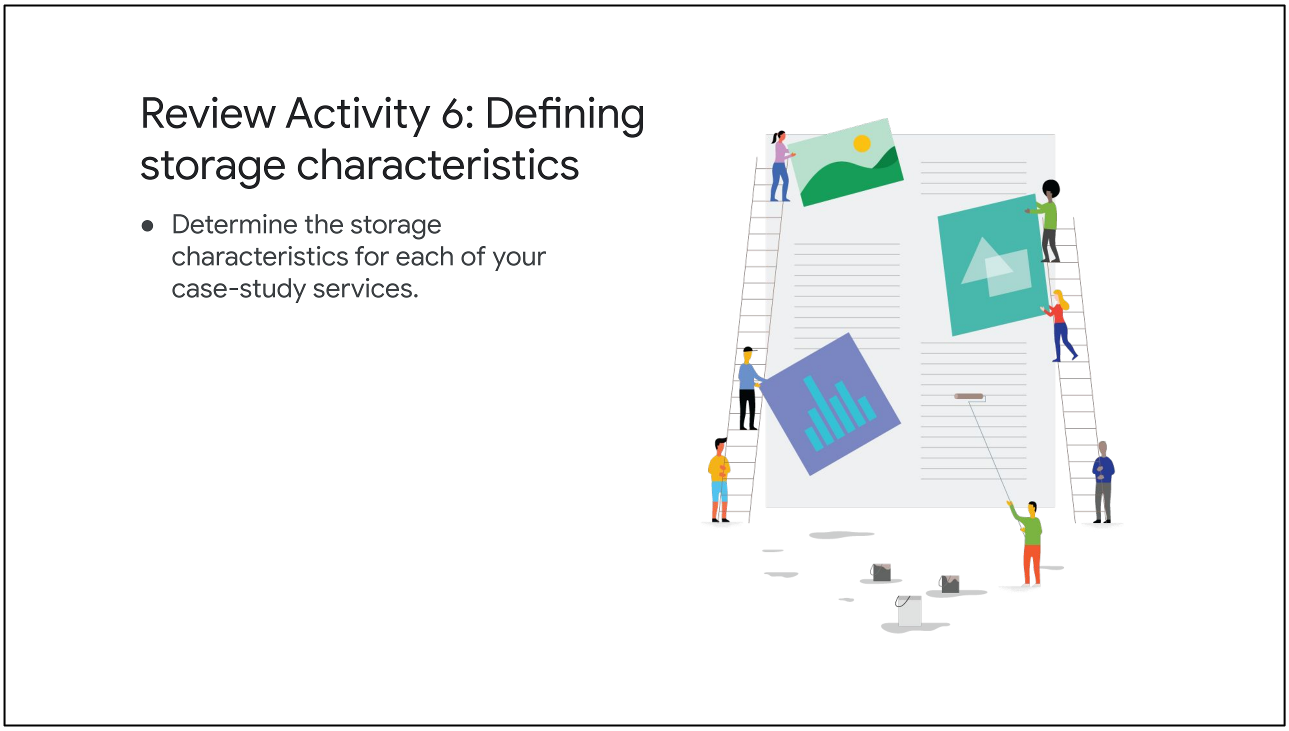
Ask yourself if you require strong consistency or if eventual consistency is good enough. Also think about how much data you have and if you need read/write or read-only.



Service	Structured or Unstructured	SQL or NoSQL	Strong or Eventual Consistency	Amount of Data (MB, GB, TB, PB, ExB)	Read only or Read/Write
Account	*Structured*	*SQL*	*Strong*	*GB*	*Read/Write*

Here's an example table for an account service that specifies all the business and technical requirements that I just mentioned.

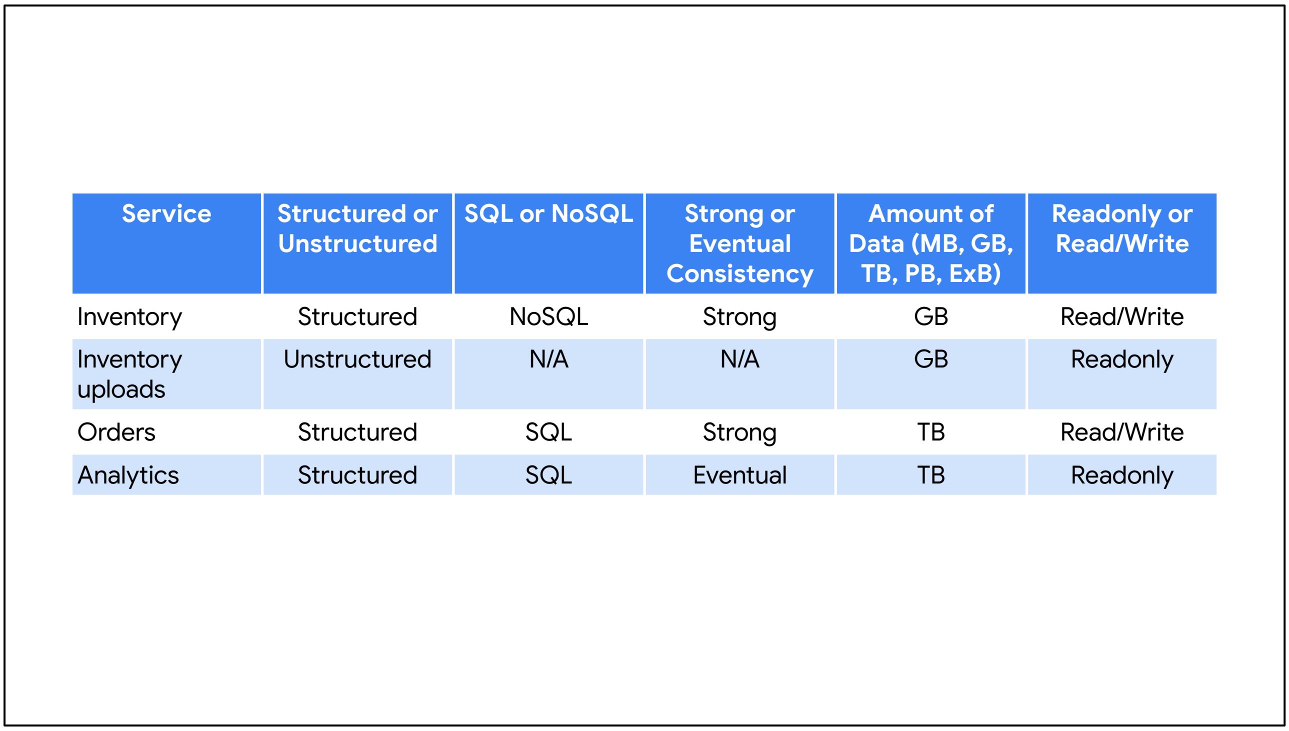
Refer to activity 6 in your design workbook to fill out a similar table for your service.



Review Activity 6: Defining storage characteristics

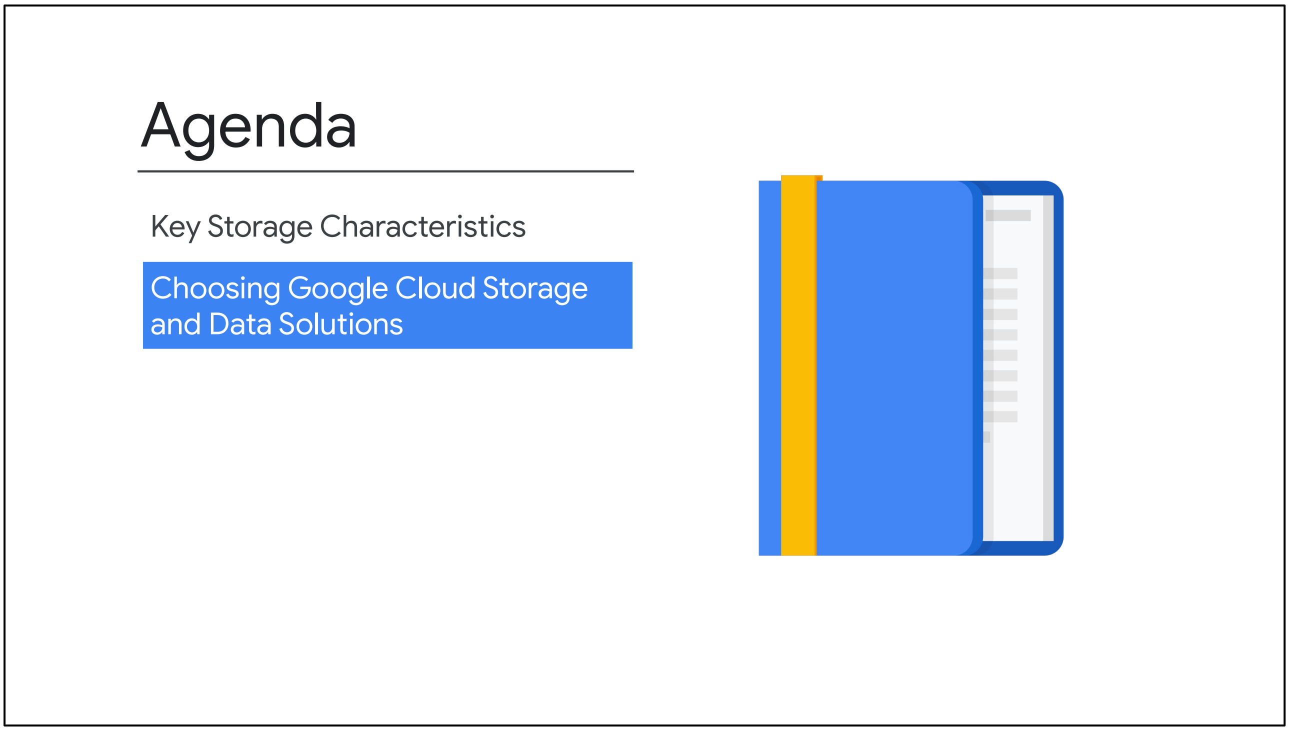
- Determine the storage characteristics for each of your case-study services.

In this activity you were asked to define the storage characteristics for the case study you are designing. These characteristics will help you choose the Google Cloud storage services most appropriate for your application.



Service	Structured or Unstructured	SQL or NoSQL	Strong or Eventual Consistency	Amount of Data (MB, GB, TB, PB, ExB)	Readonly or Read/Write
Inventory	Structured	NoSQL	Strong	GB	Read/Write
Inventory uploads	Unstructured	N/A	N/A	GB	Readonly
Orders	Structured	SQL	Strong	TB	Read/Write
Analytics	Structured	SQL	Eventual	TB	Readonly

Here's an example for our online travel portal, ClickTravel. We focussed on the inventory, inventory uploads, ordering, and analytics services. As you can see, each of these services has different requirements that might result in choosing different Google Cloud services.

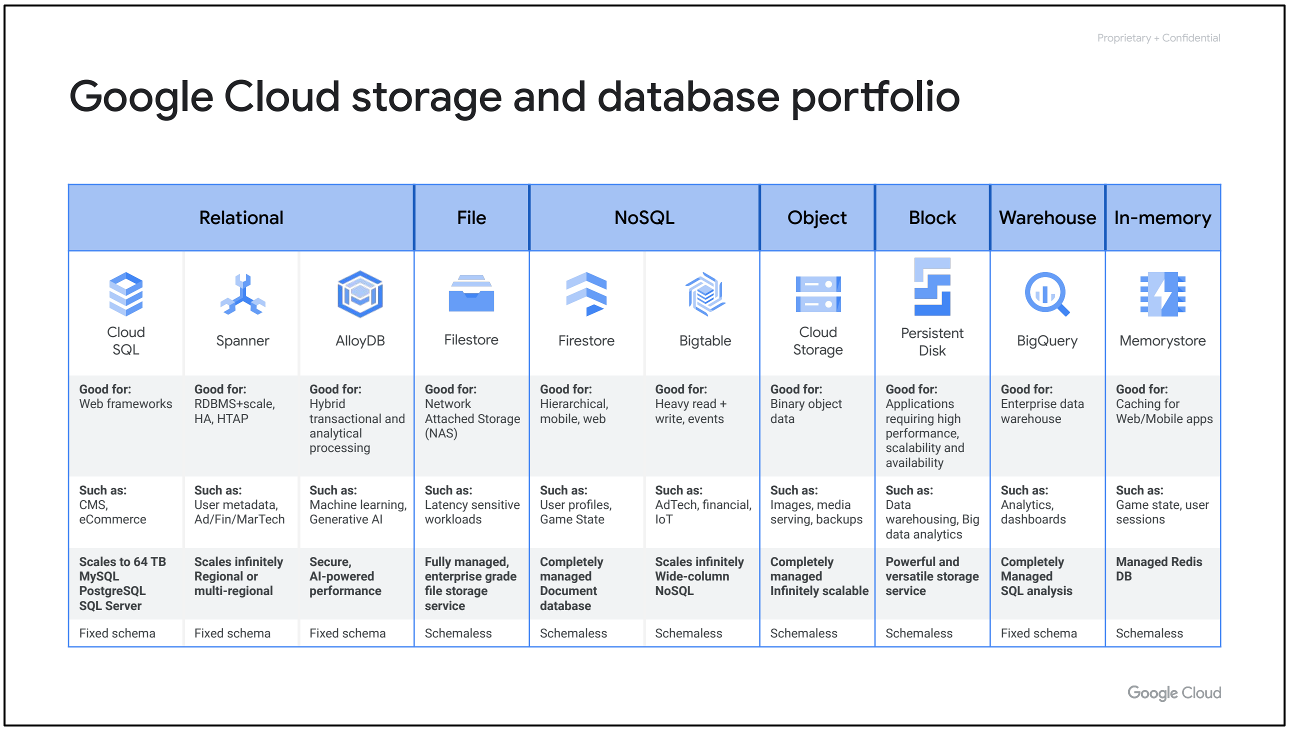


Agenda

Key Storage Characteristics

Choosing Google Cloud Storage and Data Solutions

Now that you have documented the data characteristics of your services, let’s talk about how to select Google Cloud storage and data solutions.



Google Cloud storage and database portfolio

Relational			File	NoSQL		Object	Block	Warehouse	In-memory
Cloud SQL	Spanner	AlloyDB	Filestore	Firestore	Bigtable	Cloud Storage	Persistent Disk	BigQuery	Memorystore
Good for: Web frameworks	**Good for:** RDBMS+scale, HA, HTAP	**Good for:** Hybrid transactional and analytical processing	**Good for:** Network Attached Storage (NAS)	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events	**Good for:** Binary object data	**Good for:** Applications requiring high performance, scalability and availability	**Good for:** Enterprise data warehouse	**Good for:** Caching for Web/Mobile apps
Such as: CMS, eCommerce	**Such as:** User metadata, Ad/Fin/MarTech	**Such as:** Machine learning, Generative AI	**Such as:** Latency sensitive workloads	**Such as:** User profiles, Game State	**Such as:** AdTech, financial, IoT	**Such as:** Images, media serving, backups	**Such as:** Data warehousing, Big data analytics	**Such as:** Analytics, dashboards	**Such as:** Game state, user sessions
Scales to 64 TB MySQL PostgreSQL SQL Server	**Scales infinitely Regional or multi-regional**	**Secure, AI-powered performance**	**Fully managed, enterprise grade file storage service**	**Completely managed Document database**	**Scales infinitely Wide-column NoSQL**	**Completely managed Infinitely scalable**	**Powerful and versatile storage service**	**Completely Managed SQL analysis**	**Managed Redis DB**
Fixed schema	Fixed schema	Fixed schema	Schemaless	Schemaless	Schemaless	Schemaless	Schemaless	Fixed schema	Schemaless

Google Cloud

The Google Cloud storage and database portfolio covers relational, file, NoSQL, object, block, data warehouse, and in-memory stores, as shown in this table. Let's discuss each service from left to right.

Relational

- **Cloud SQL** is a fixed schema database with a storage limit of 64 terabytes. It is offered using MySQL, PostgreSQL, and SQL Server. These services are good for web applications such as CMS or eCommerce.
- **Spanner** is also a relational and fixed schema but scales infinitely and can be regional or multi-regional. Example use cases include scalable relational databases greater than 30 GB with high availability and also global accessibility, like supply-chain management and manufacturing.
- **AlloyDB** is a fixed schema, fully managed PostgreSQL-compatible database. It provides enterprise-grade performance and availability while maintaining 100% compatibility with open-source PostgreSQL.

File

- **Filestore** is a schemaless, high-performance, fully managed file store service for applications that require a filesystem interface and a shared file system for data. Filestore gives users a simple, native experience for standing up managed Network Attached Storage (NAS) with their Compute Engine and Google Kubernetes Engine instances.

NoSQL

Google Cloud’s **NoSQL** datastores are schemaless.

- **Firestore** is a completely managed document datastore with a maximum document size of 1 MB. It is useful for hierarchical data; for example, a game state of user profiles.
- **Bigtable** is also a NoSQL datastore that scales infinitely. It is good for heavy read and write events and use cases including financial services, Internet of Things, and digital advert streams.

Object

- For object storage, Google Cloud offers **Cloud Storage**. Cloud Storage is schemaless and completely managed with infinite scale. It stores binary object data and so it's good for storing images, media serving, and backups.

Block

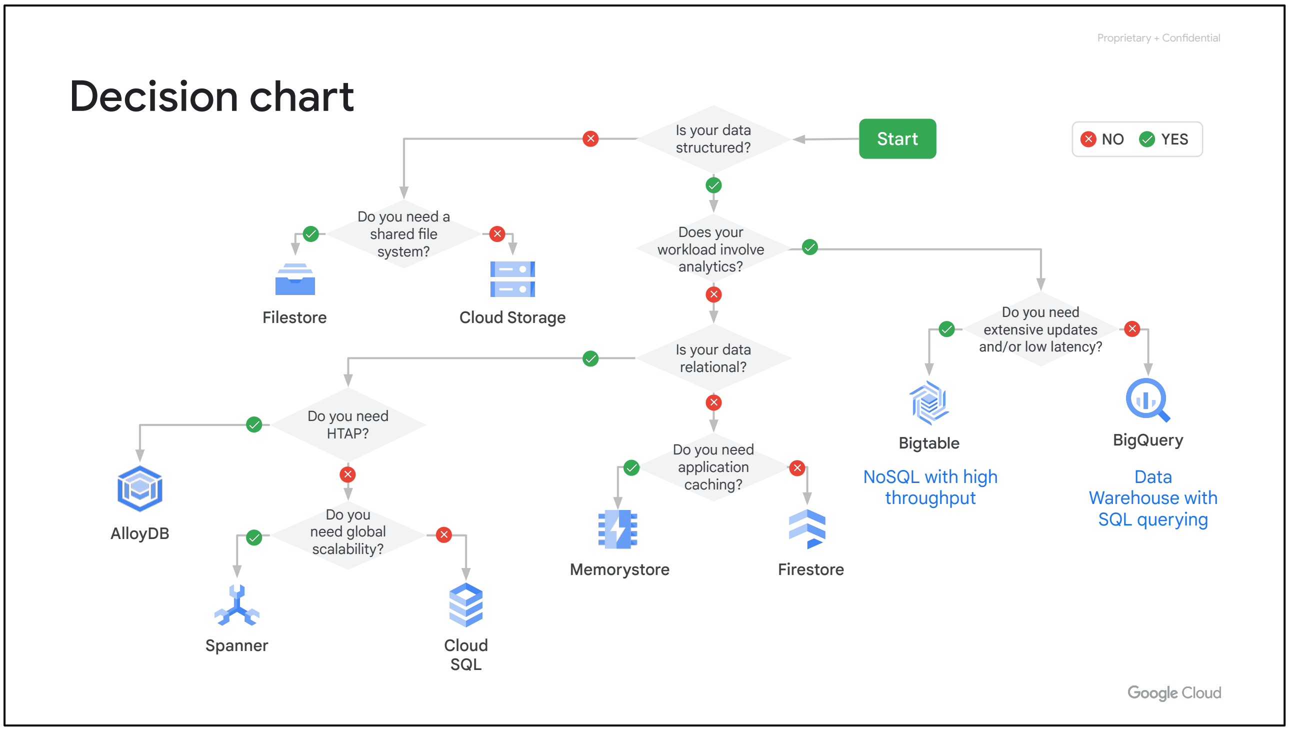
- **Persistent Disk** volumes are schemaless, durable network storage devices that your virtual machine (VM) instances can access like physical disks in a desktop or server. The data on each Persistent Disk volume is distributed across several physical disks.

Warehouse

- Data warehousing is provided by **BigQuery**. The storage uses a fixed schema and supports completely managed SQL analysis of data stored. It is excellent for performing analytics and business intelligence dashboards.

In memory

- For in-memory storage, **Memorystore** provides a schemaless-managed Redis database. It is excellent for caching web and mobile apps and for providing fast access to state in microservice architectures.


Proprietary + Confidential
Decision chart
Start
NO
YES
Is your data structured?
Do you need a shared file system?
Filestore
Cloud Storage
Does your workload involve analytics?
Is your data relational?
Do you need extensive updates and/or low latency?
Bigtable
BigQuery
NoSQL with high throughput
Data Warehouse with SQL querying
Do you need HTAP?
AlloyDB
Do you need global scalability?
Spanner
Cloud SQL
Do you need application caching?
Memorystore
Firestore
Google Cloud

Let’s summarize the services in this module with this decision chart.

- First, ask yourself: Is your data structured, and will it need to be accessed using its structured data format?

- If the answer is no, then ask yourself if you need a shared file system. If you do, then choose Filestore. If you don't, then choose Cloud Storage.

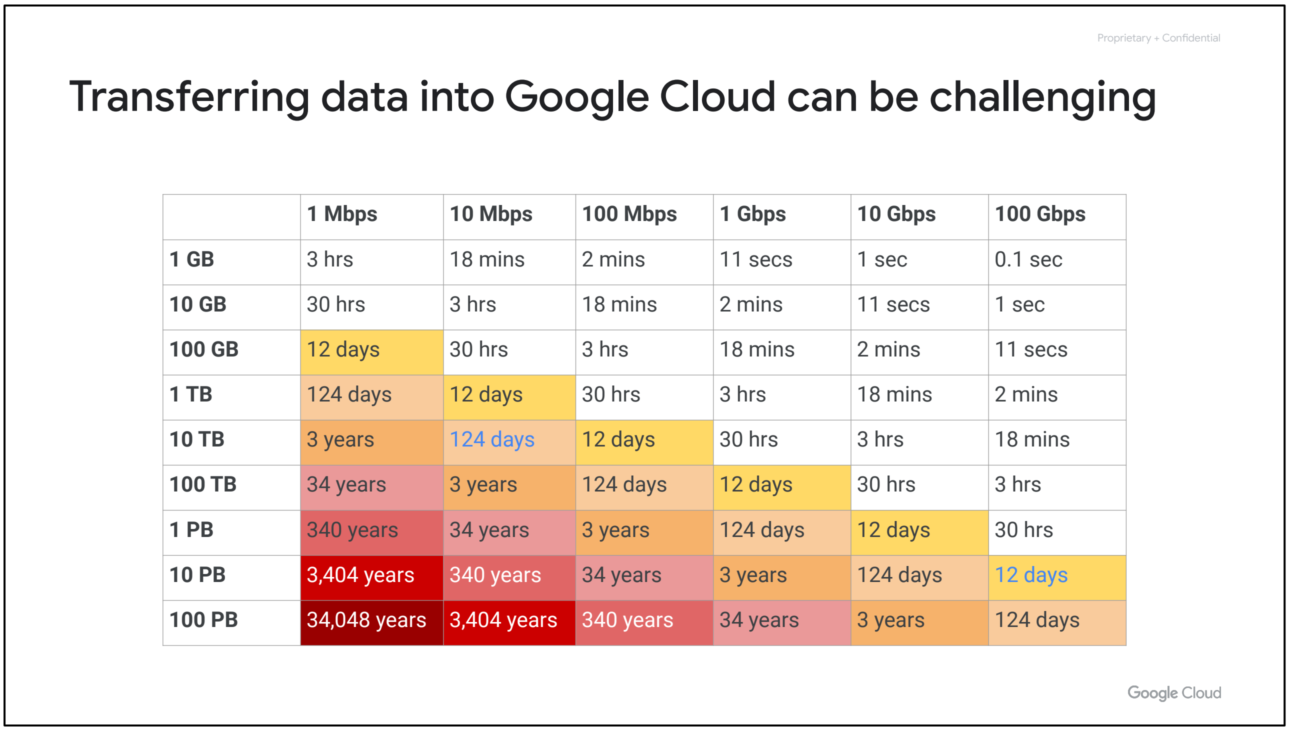
- If your data is structured and needs to be accessed in this way, then ask yourself, does your workload focus on analytics? If it does, you will want to choose Bigtable or BigQuery, depending on your latency and update needs.
  - BigQuery is recommended as a data warehouse. It is the default storage for tabular data, and is optimized for large-scale, ad-hoc SQL-based analysis and reporting. While BigQuery data manipulation language (DML) enables you to update, insert, and delete data from your BigQuery tables, because it has a built-in cache BigQuery works really well in cases where data does not change often.
  - Bigtable is a NoSQL wide-column database. It's optimized for low latency, large numbers of reads and writes, and maintaining performance at scale.
  - In addition to analytics, Bigtable is also suited as a ‘fast lookup’ non-relational database for datasets too large to store in memory, with use cases in areas such as IoT, AdTech and FinTec.

- If your workload doesn’t involve analytics, check whether your data is relational. If it’s not relational, do you need application caching?
    - If caching is a requirement, choose Memorystore, an in-memory database.
    - Otherwise choose Firestore, a document database.

- If your data is relational and you need Hybrid transaction/analytical processing (HTAP) choose AlloyDB.
    - If you don’t need HTAP and don’t need global scalability, choose Cloud SQL.
    - If you don’t need HTAP and need global scalability, choose Spanner.

Depending on your application, you might use one or several of these services to get the job done. For more information on how to choose between these different services, please refer to the links on storage options and databases in the Course Resources.

https://cloud.google.com/storage-options/
https://cloud.google.com/products/databases/


Proprietary + Confidential

Transferring data into Google Cloud can be challenging

	1 Mbps	10 Mbps	100 Mbps	1 Gbps	10 Gbps	100 Gbps
1 GB	3 hrs	18 mins	2 mins	11 secs	1 sec	0.1 sec
10 GB	30 hrs	3 hrs	18 mins	2 mins	11 secs	1 sec
100 GB	12 days	30 hrs	3 hrs	18 mins	2 mins	11 secs
1 TB	124 days	12 days	30 hrs	3 hrs	18 mins	2 mins
10 TB	3 years	124 days	12 days	30 hrs	3 hrs	18 mins
100 TB	34 years	3 years	124 days	12 days	30 hrs	3 hrs
1 PB	340 years	34 years	3 years	124 days	12 days	30 hrs
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

Google Cloud

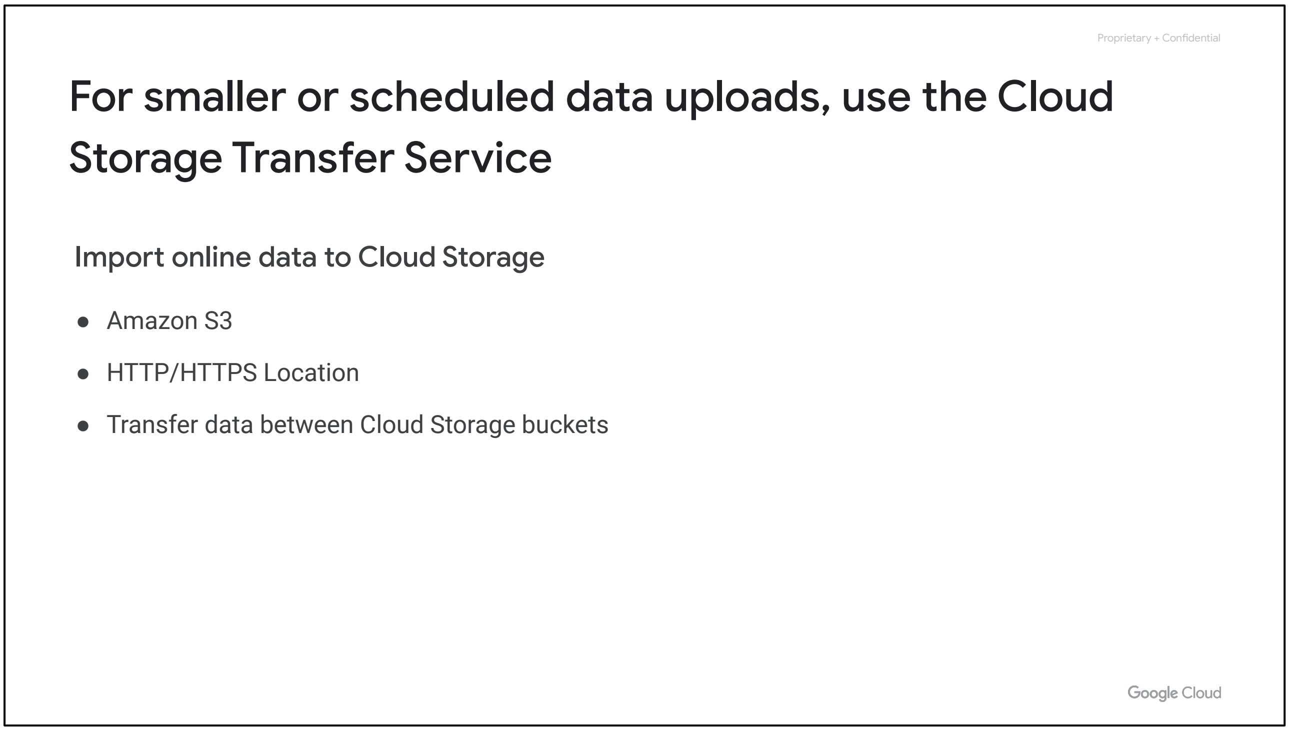
You might also want to consider how to transfer data in Google Cloud. A number of factors must be considered, including cost, time, offline versus online transfer options, and security.

While transfer into Cloud Storage is free, there will be costs with the storage of the data and maybe even appliance costs if a transfer appliance is used or egress costs if transferring from another cloud provider.

If you have huge datasets, the time required to transfer across a network may be unrealistic. Even if it is realistic, the effects on your organization's infrastructure may be damaging while the transfer is taking place.

This table shows the challenge of moving large datasets. The color coded cells highlight unrealistic timelines that require alternative solutions.

Let's go over online and offline data transfer options.



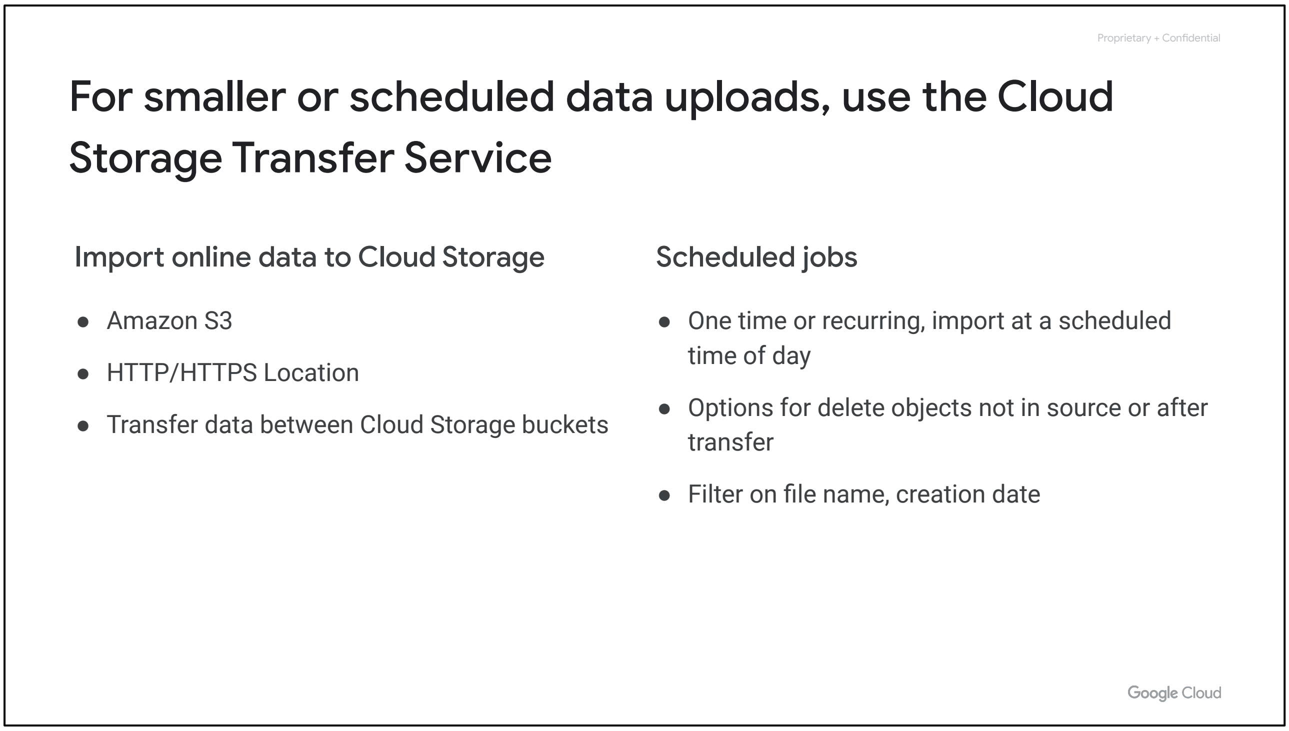
For smaller or scheduled data uploads, use the Cloud Storage Transfer Service

Import online data to Cloud Storage

- Amazon S3
- HTTP/HTTPS Location
- Transfer data between Cloud Storage buckets

For smaller or scheduled data uploads, use Cloud Storage Transfer Service, which enables you to:

- Move or back up data to a Cloud Storage bucket from another cloud storage provider such as Amazon S3, from your on-premises storage, or from any HTTP/HTTPS location.
- Move data from one Cloud Storage bucket to another, so that it is available to different groups of users or applications.
- Periodically move data as part of the data processing pipeline or analytical workflow.



For smaller or scheduled data uploads, use the Cloud Storage Transfer Service

Import online data to Cloud Storage

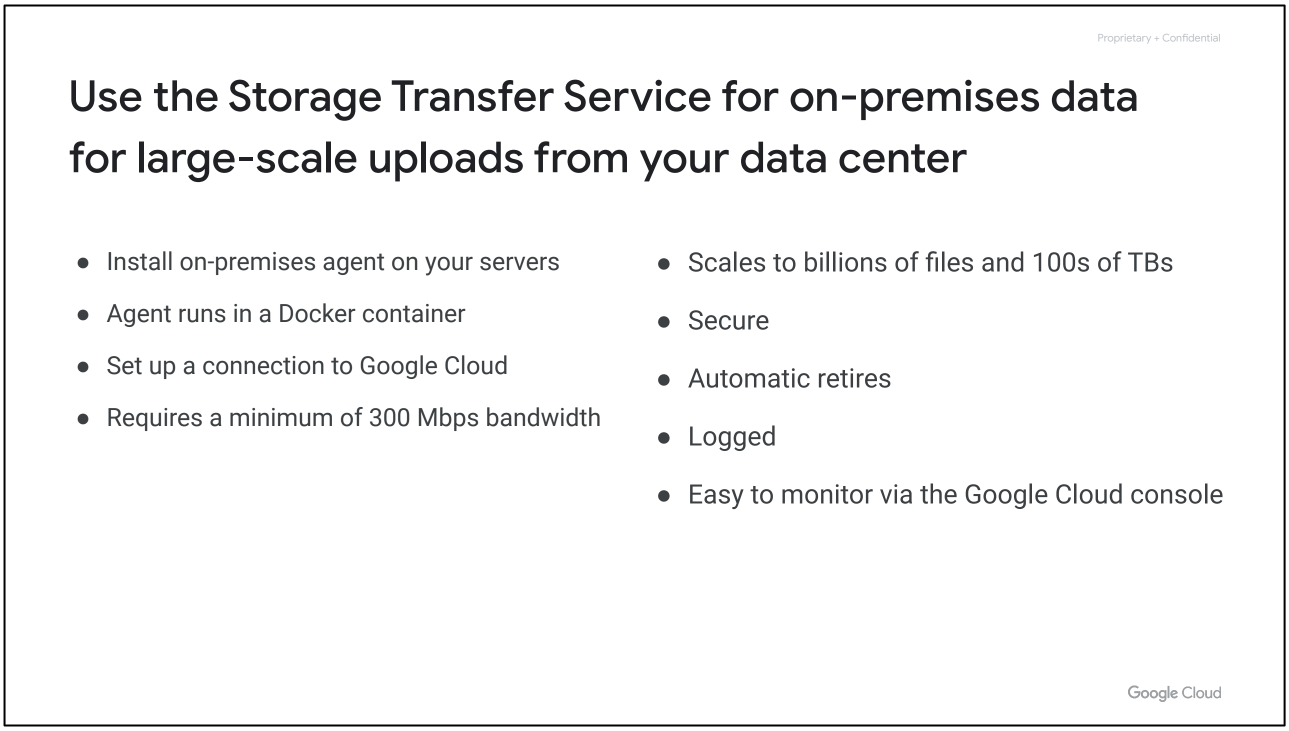
- Amazon S3
- HTTP/HTTPS Location
- Transfer data between Cloud Storage buckets

Scheduled jobs

- One time or recurring, import at a scheduled time of day
- Options for delete objects not in source or after transfer
- Filter on file name, creation date

Storage Transfer Service provides options that can make data transfers and synchronization easier. For example, you can:

- Schedule one-time transfer operations or recurring transfer operations.
- Delete existing objects in the destination bucket if they don’t have a corresponding object in the source.
- Delete data source objects after transferring them.
- Schedule periodic synchronization from a data source to a data sink with advanced filters based on file creation dates, file-name filters, and times of day you prefer to import data.


Proprietary + Confidential

Use the Storage Transfer Service for on-premises data for large-scale uploads from your data center

- Install on-premises agent on your servers
- Agent runs in a Docker container
- Set up a connection to Google Cloud
- Requires a minimum of 300 Mbps bandwidth

- Scales to billions of files and 100s of TBs
- Secure
- Automatic retires
- Logged
- Easy to monitor via the Google Cloud console

Google Cloud

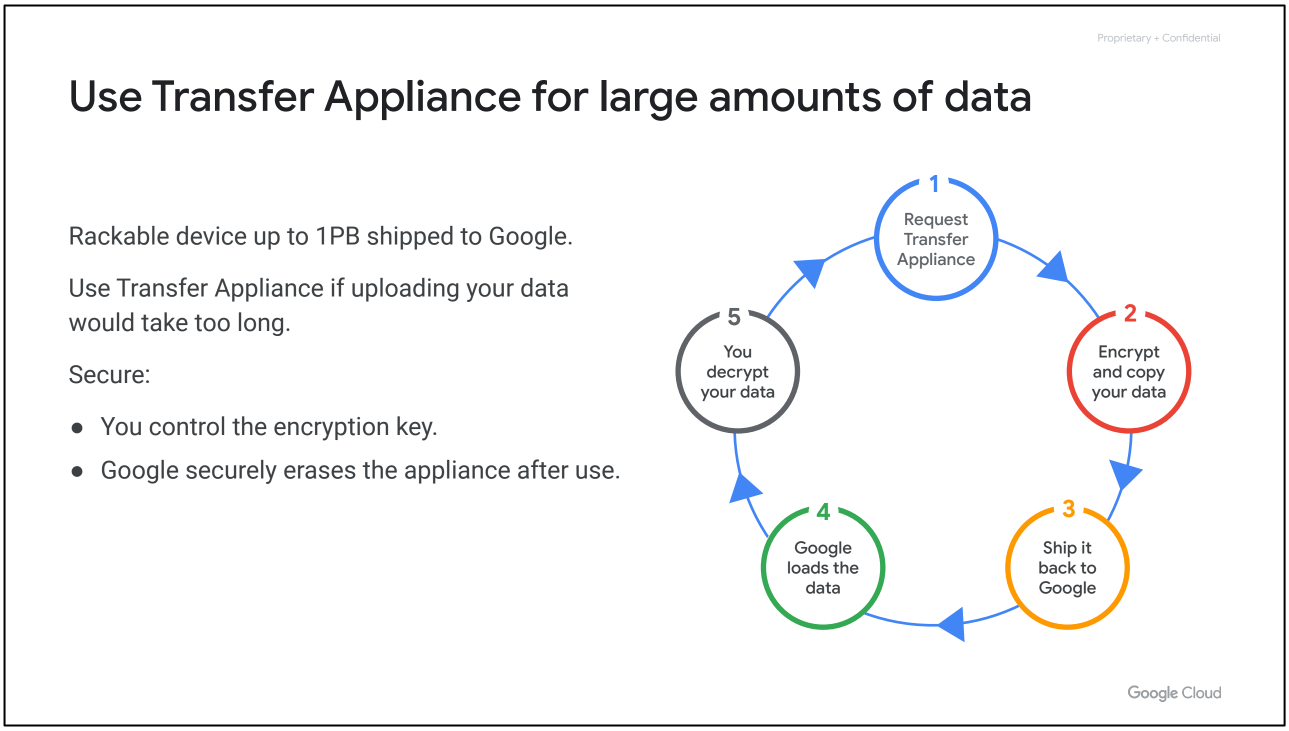
Use the Storage Transfer Service for on-premises data for large-scale uploads from your data center.

The Storage Transfer Service for on-premises data allows large-scale online data transfers from on-premises storage to Cloud Storage. With this service, data validation, encryption, error retries, and fault tolerance are built in. On-premises software is installed on your servers—the agent comes as a Docker container and a connection to Google Cloud is set up. Directories to be transferred to Cloud Storage are selected in the Cloud Console.

Once the data transfer begins, the service will parallelize the transfer across many agents supporting scale up to billions of files and 100s of TBs. Via the Cloud Console, a user can view detailed transfer logs and also the creation, management, and monitoring of transfer jobs.

To use the Storage Transfer Service for on-premises, a Posix-compliant source is required and a network connection of at least 300Mbps. Also, a Docker-supported Linux server that can access the data to be transferred is required with ports 80 and 443 open for outbound connections.

The use case is for on-premises transfer of data whose size is > 1TB.



Proprietary + Confidential

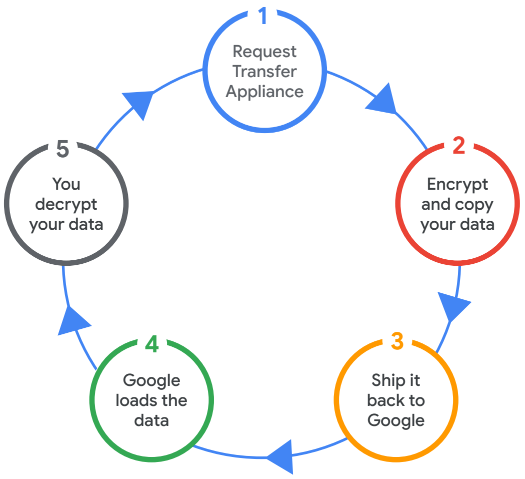
Use Transfer Appliance for large amounts of data

Rackable device up to 1PB shipped to Google.

Use Transfer Appliance if uploading your data would take too long.

Secure:

- You control the encryption key.
- Google securely erases the appliance after use.



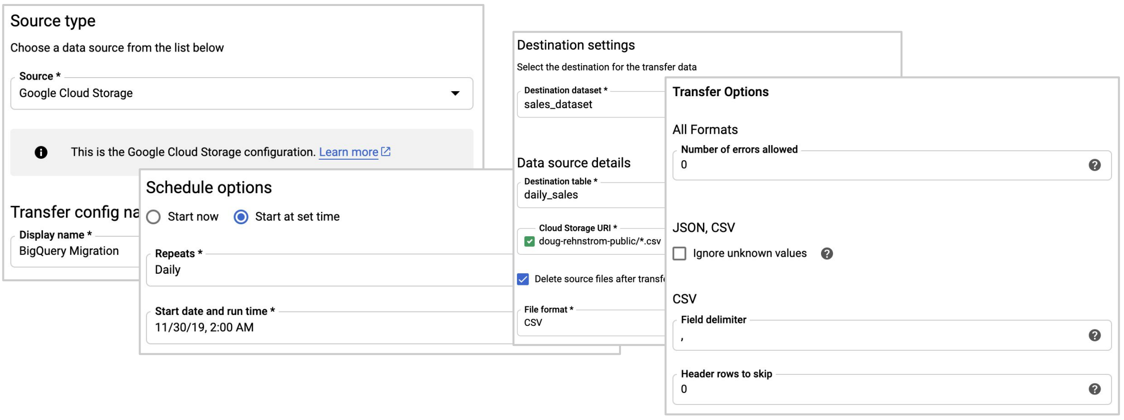
Google Cloud

For large amounts of on-premises data that would take to long to upload, use Transfer Appliance.

Transfer Appliance is a secure, rackable, high-capacity storage server that you set up in your data center. You fill it with data and ship it to an ingest location, where the data is uploaded to Google. The data is secure, you control the encryption key, and Google erases the appliance after the transfer is complete.

The process for using a transfer appliance is that you request an appliance, and it is shipped in a tamper-evident case. Data is transferred to the appliance, and the appliance is shipped back to Google, data is loaded to Cloud Storage, and you are notified that it is available. Google uses tamper-evident seals on shipping cases to and from the data ingest site. Data is encrypted to AES256 standard at the moment of capture. Once the transfer is complete, the appliance is erased per NIST-800-88 standards. You decrypt the data when you want to use it.

There's also a transfer service for BigQuery



Source type

Choose a data source from the list below

Source *: Google Cloud Storage

This is the Google Cloud Storage configuration. Learn more

Transfer config na

Display name *: BigQuery Migration

Schedule options

Start now [✓ Start at set time]

Repeats *: Daily

Start date and run time *: 11/30/19, 2:00 AM

Destination settings

Select the destination for the transfer data

Destination dataset *: sales_dataset

Data source details

Destination table *: daily_sales

Cloud Storage URI *: ✓ doug-rehnstrom-public/*.csv

✓ Delete source files after transf

File format *: CSV

Transfer Options

All Formats

Number of errors allowed: 0

JSON, CSV

☐ Ignore unknown values

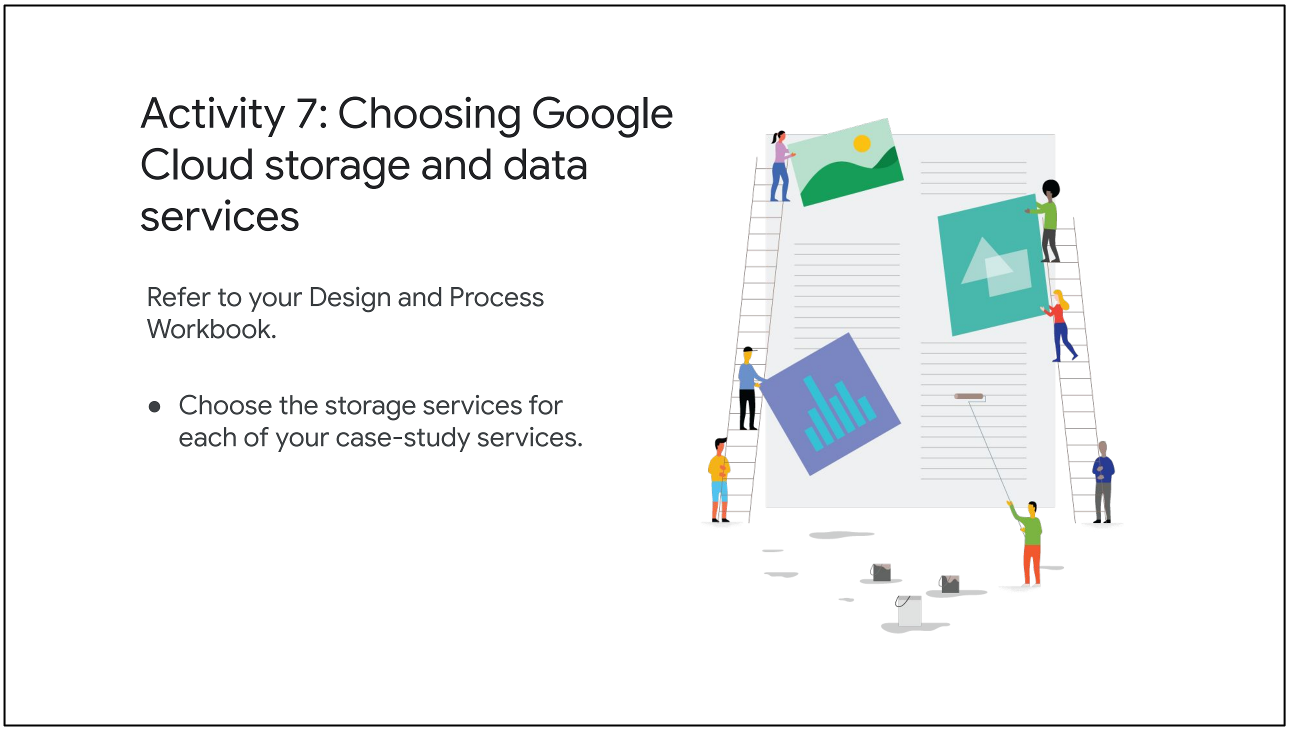
CSV

Field delimiter: ,

Header rows to skip: 0

There is also a transfer service for BigQuery. The BigQuery Data Transfer Service automates data movement from SaaS applications to BigQuery on a scheduled, managed basis. Data Transfer Service initially supports Google application sources like Google Ads, Campaign Manager, Google Ad Manager, and YouTube. There are also data connectors that allow easy data transfer from Teradata, Amazon Redshift, and Amazon S3 to BigQuery.

The screenshots on the slide show a source type is selected for a transfer, a schedule is configured, and a data destination is selected. For the transfer, the data formats are also configured.



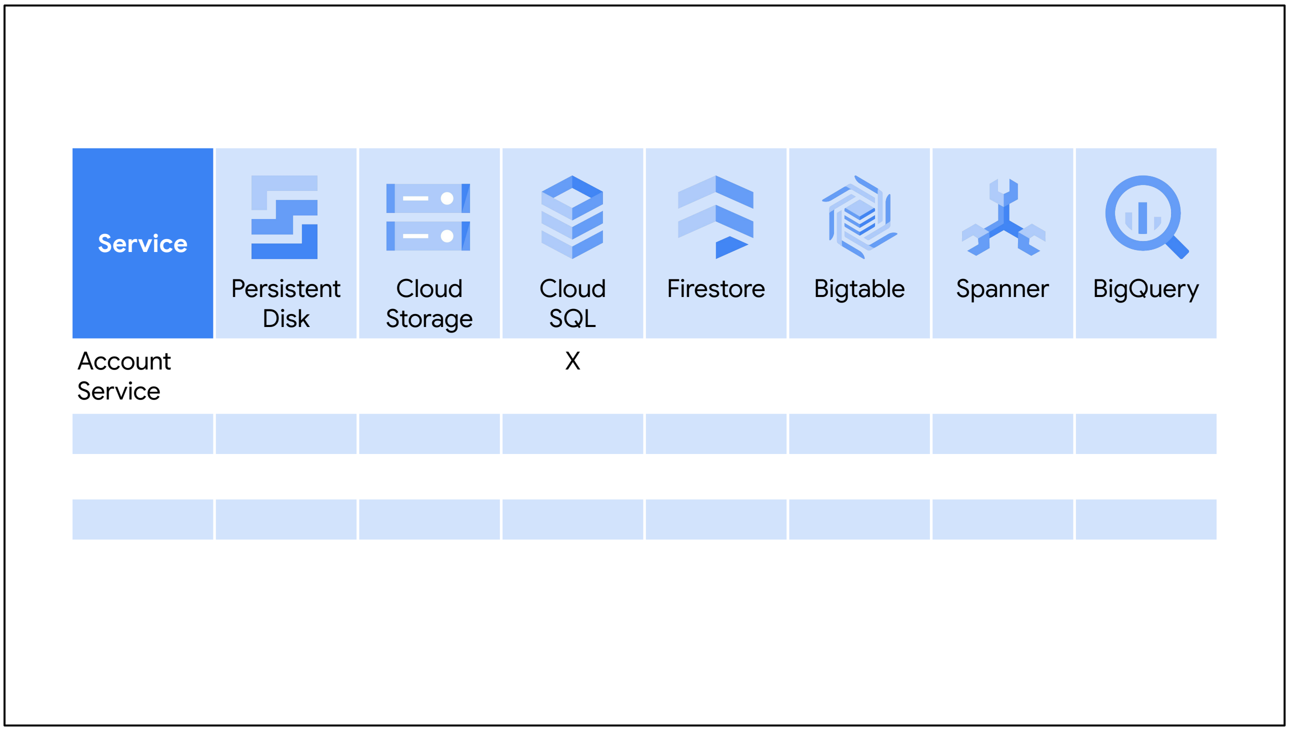
Activity 7: Choosing Google Cloud storage and data services

Refer to your Design and Process Workbook.

- Choose the storage services for each of your case-study services.

You will now select storage products for each of your case study services.

Use the storage characteristics and your understanding of Google Cloud's various services that we covered in this module to help you choose which storage services would be most appropriate for each of your microservices.



Service	Persistent Disk	Cloud Storage	Cloud SQL	Firestore	Bigtable	Spanner	BigQuery
Account Service			X				

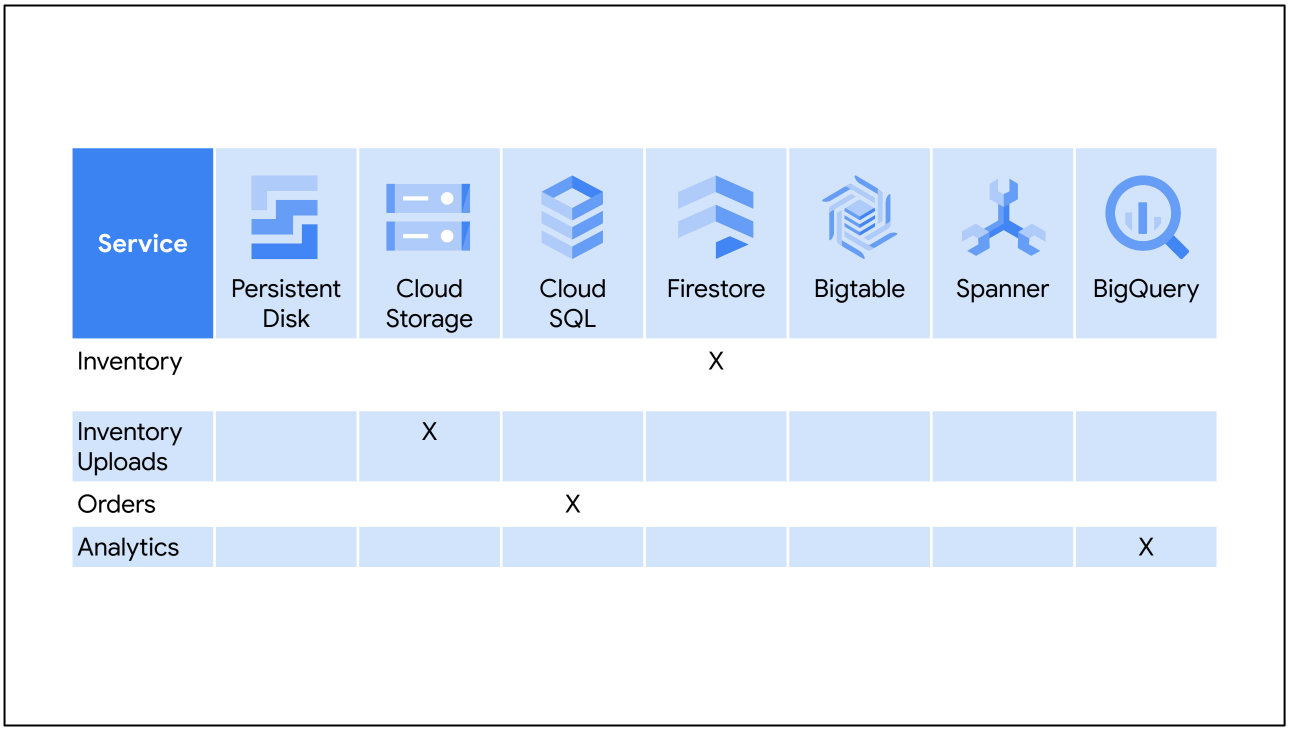
Here’s an example table for an account service that should leverage Cloud SQL because we require a relational database that’s strongly consistent with GB of data and read/write capabilities.

Refer to activity 7 in your design workbook to fill out a similar table for your application’s services.

Review Activity 7: Choosing Google Cloud storage and data services

- Choose the storage services for each of your case-study services.

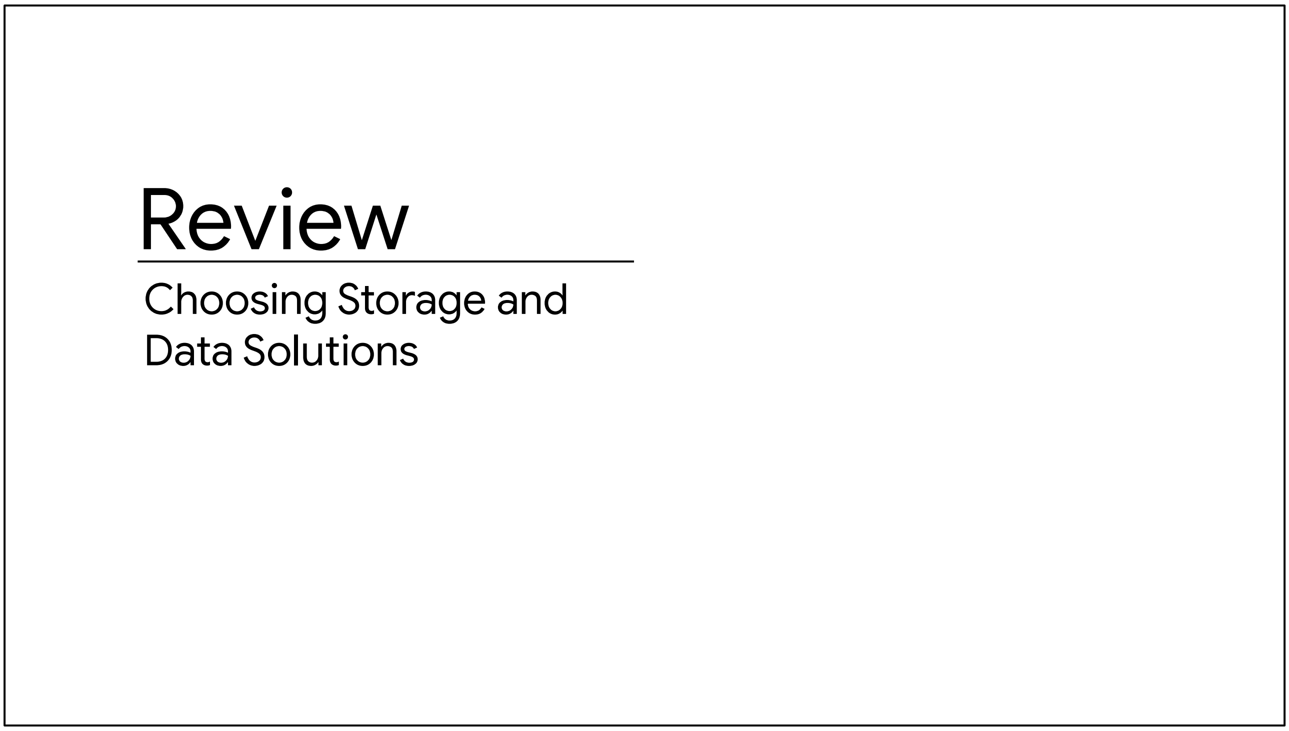
In this activity you were asked to select the appropriate Google Cloud storage services for your case study.



Service	Persistent Disk	Cloud Storage	Cloud SQL	Firestore	Bigtable	Spanner	BigQuery
Inventory				X			
Inventory Uploads		X					
Orders			X				
Analytics							X

Here’s an example for our online travel portal, ClickTravel:

- For the inventory service we will use Cloud Storage for the raw inventory uploads. Suppliers will upload their inventory as JSON data stored in text files. That inventory will then be imported into a Firestore database.
- The orders service will store its data in a relational database running in Cloud SQL.
- The analytics service will aggregate data from various sources into a data warehouse, for which we’ll use BigQuery.



Review

Choosing Storage and Data Solutions

In this module, we covered the various storage services available in Google Cloud. These include Cloud Storage for binary data, Cloud SQL and Spanner for relational databases, Bigtable and Firestore for NoSQL databases, and BigQuery for data warehouses. We also talked about different storage characteristics and how they can be used to help you choose where to store your data.