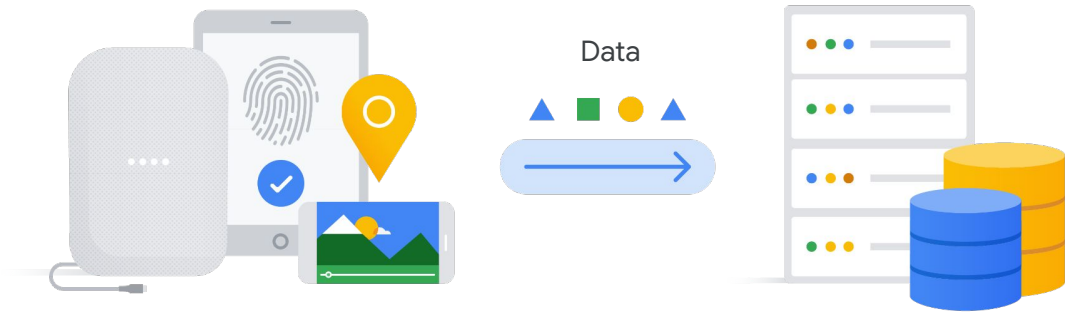
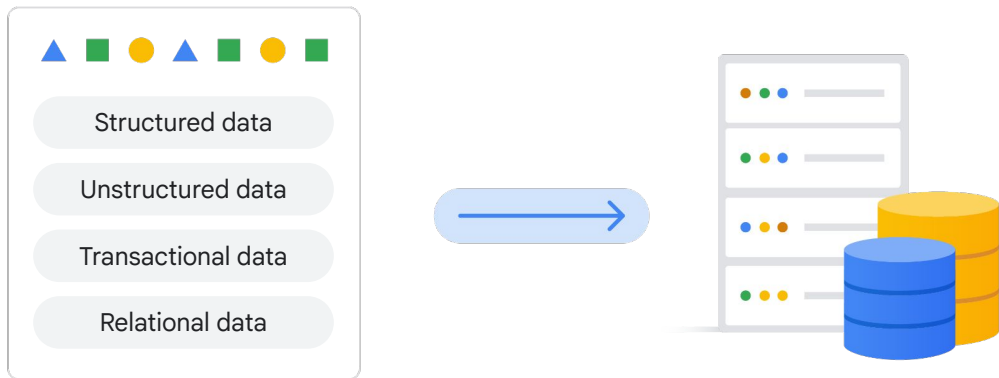


Google Cloud Core Infrastructure Module 4

On-demand course
March 2022



Every application needs to store data, like media to be streamed or perhaps even sensor data from devices, and different applications and workloads require different storage database solutions.



Google Cloud has storage options for structured, unstructured, transactional, and relational data.

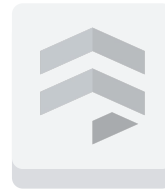
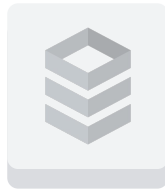
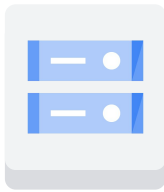


In this section of the course, we'll explore Google Cloud's five core storage products:

- Cloud Storage,
- Cloud SQL,
- Cloud Spanner,
- Firestore, and
- Cloud Bigtable.

Depending on your application, you might use one or several of these services to do the job.

Cloud Storage



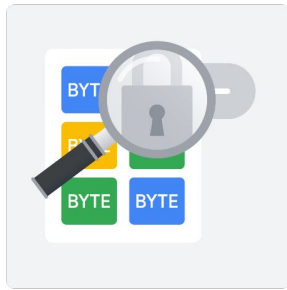
Let's begin with **Cloud Storage**, which is a service that offers developers and IT organizations durable and highly available **object storage**.

But what is object storage?

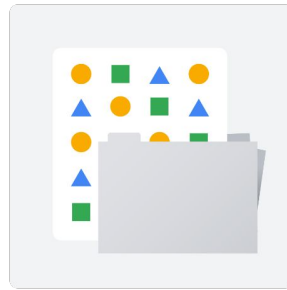


Object storage

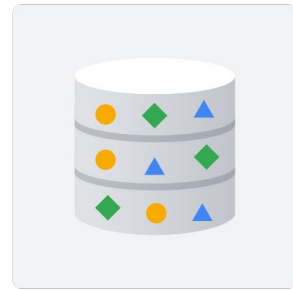
Object storage is a computer data storage architecture that manages data as "objects"



Object storage

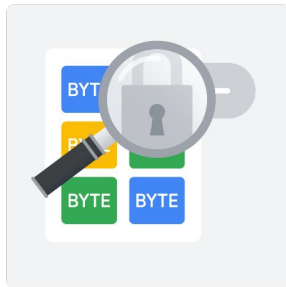


File storage



Block storage

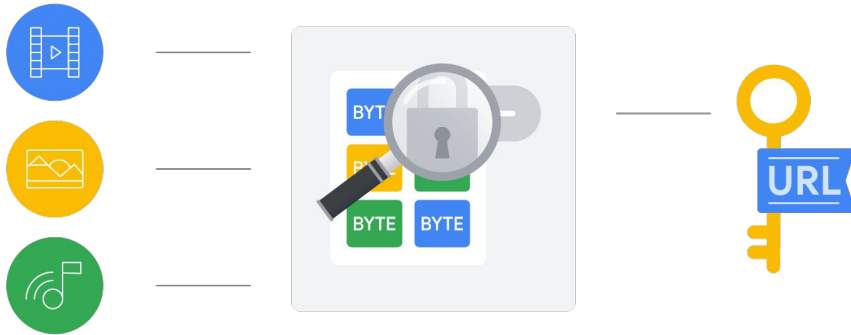
and not as a file and folder hierarchy (file storage), or as chunks of a disk (block storage).



Object packaged format contains:

- ✓ Binary form of the actual data itself
- ✓ Relevant associated meta-data
- ✓ Globally unique identifier

These objects are stored in a packaged format which contains the binary form of the actual data itself, as well as relevant associated meta-data (such as date created, author, resource type, and permissions), and a globally unique identifier.



These unique keys are in the form of URLs, which means object storage interacts well with web technologies. Data commonly stored as objects include video, pictures, and audio recordings.



Cloud Storage



Google's object storage product



Allows customers to store any amount of data, and to retrieve it as often as needed

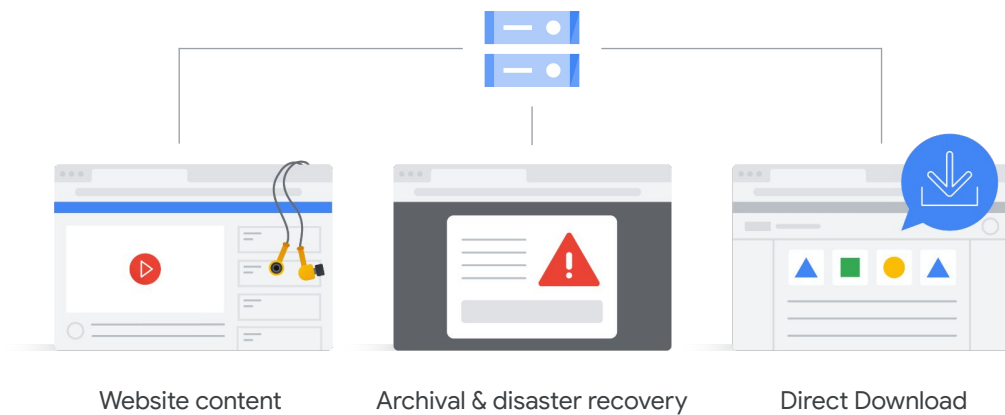


Fully managed scalable service that has a wide variety of uses

Cloud Storage is Google's object storage product.

It allows customers to store any amount of data, and to retrieve it as often as needed.

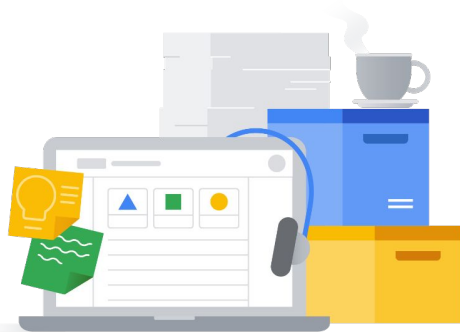
It's a fully managed scalable service that has a wide variety of uses.



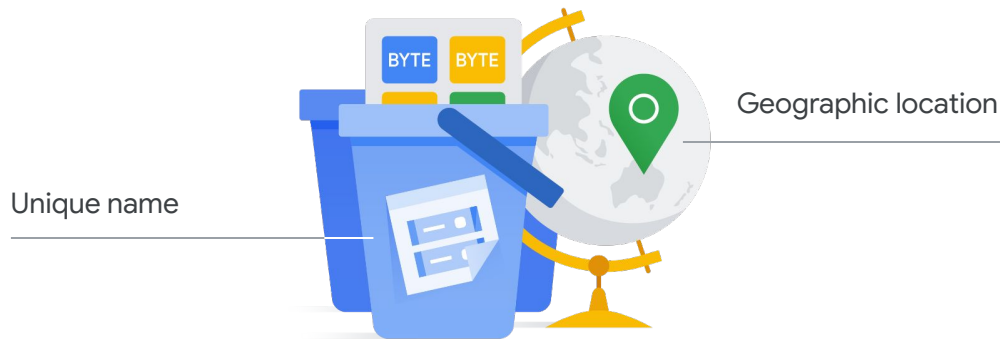
A few examples include serving website content, storing data for archival and disaster recovery, and distributing large data objects to end users via Direct Download.

Binary large-object (BLOB) storage

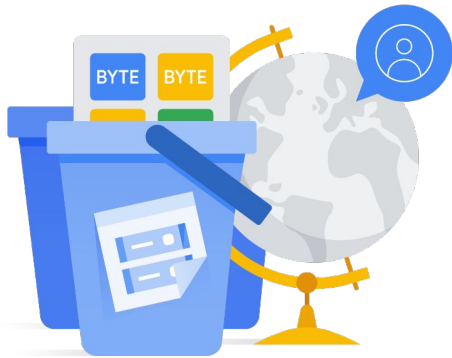
- ✓ Online content
- ✓ Backup and archiving
- ✓ Storage of intermediate results



Cloud Storage's primary use is whenever binary large-object storage (also known as a "BLOB") is needed for online content such as videos and photos, for backup and archived data and for storage of intermediate results in processing workflows.



Cloud Storage files are organized into buckets. A bucket needs a globally unique name and a specific geographic location for where it should be stored, and an ideal location for a bucket is where latency is minimized.

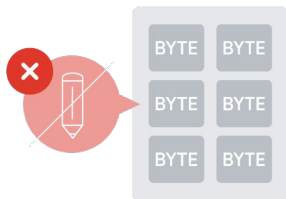


European location

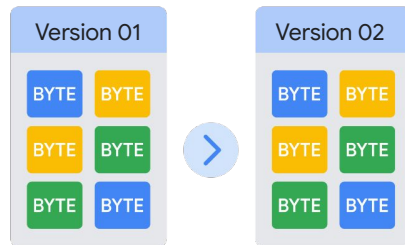
- ✓ Google Cloud region in Europe
- ✓ The EU multi-region

For example, if most of your users are in Europe, you probably want to pick a European location, so either a specific Google Cloud region in Europe, or else the EU multi-region.

Immutable objects

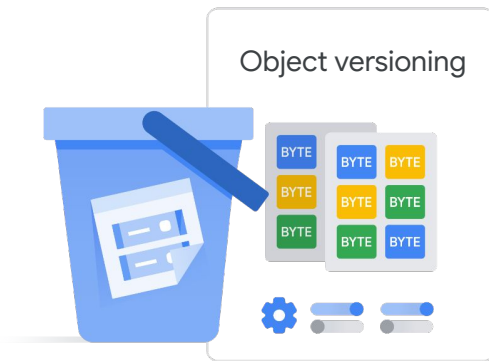


Objects are not edited




A new version is created

The storage objects offered by Cloud Storage are **immutable**, which means that you do not edit them, but instead a new version is created with every change made.



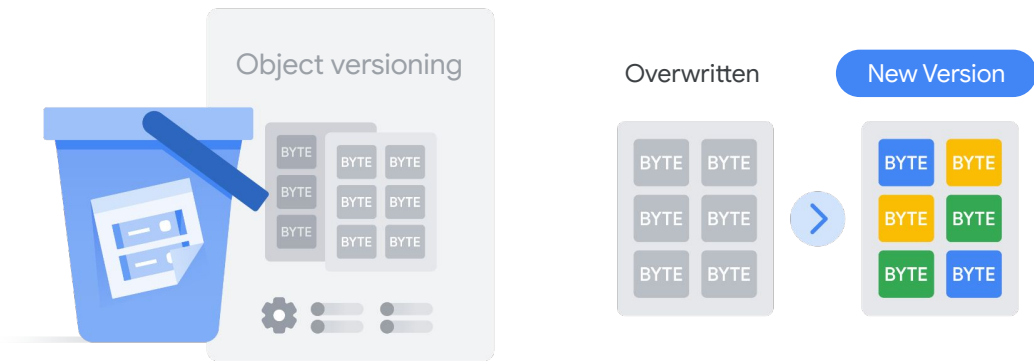
History	V1	V2	V3
Object 01	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Object 02	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Object 03	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Administrators have the option to either allow each new version to completely overwrite the older one, or to keep track of each change made to a particular object by enabling “versioning” within a bucket.

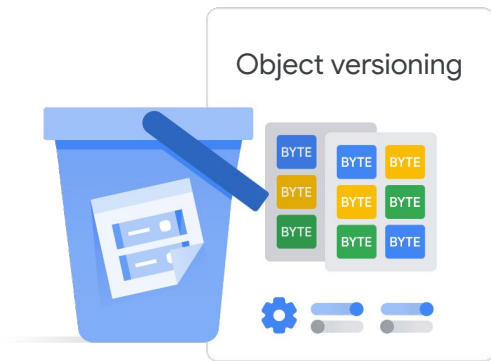


History of modifications				
Version 4	BYTE	BYTE	BYTE	BYTE
Version 3	BYTE	BYTE	BYTE	BYTE
Version 2	BYTE	BYTE	BYTE	BYTE
Version 1	BYTE	BYTE	BYTE	BYTE

If you choose to use versioning, Cloud Storage will keep a detailed history of modifications -- that is, overwrites or deletes -- of all objects contained in that bucket.

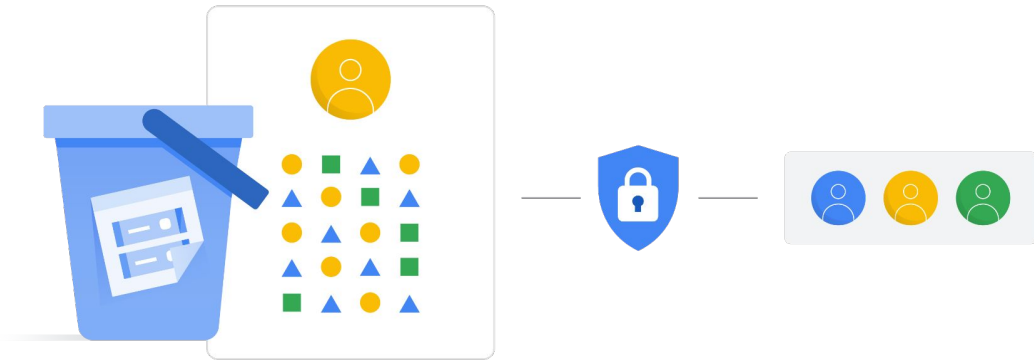


If you don't turn on object versioning, by default new versions will always overwrite older versions.

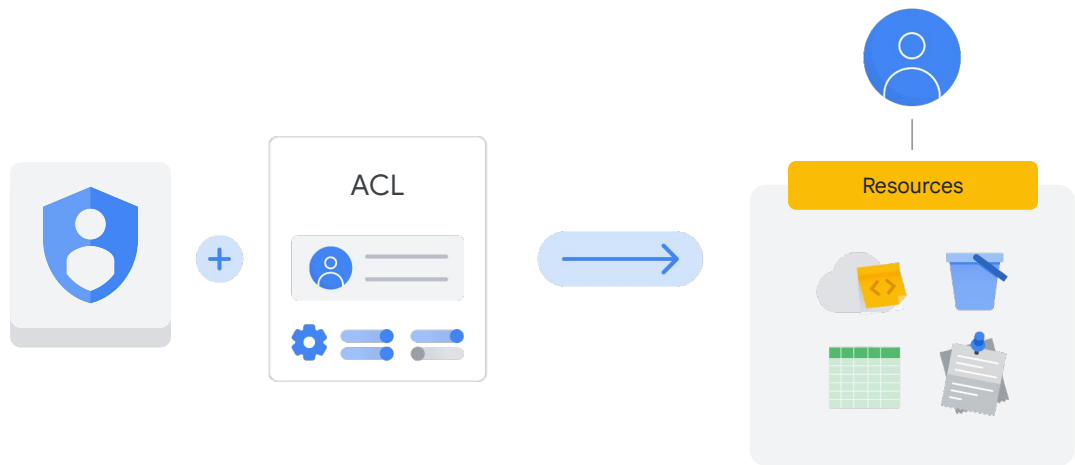


History	V1	V2	V3
Object 01	✓		
Object 02	✓	✓	✓
Object 03	✗	✓	

With object versioning enabled, you can list the archived versions of an object, restore an object to an older state, or permanently delete a version of an object, as needed.



In many cases, personally identifiable information may be contained in data objects, so controlling access to stored data is essential to ensuring security and privacy are maintained.



Using IAM roles and, where needed, access control lists (ACLs), organizations can conform to security best practices, which require each user to have access and permissions to **only the resources they need to do their jobs**, and no more than that.

01

For most purposes, Cloud IAM is sufficient

Inherited from project to bucket to object

02

If you need finer control, you can create access control lists (ACLs)

Scope



Permission



There are a couple of options to control user access to objects and buckets. For most purposes, IAM is sufficient. Roles are inherited from project to bucket to object.

If you need finer control, you can create access control lists. Each access control list consists of two pieces of information. The first is a scope, which defines who can access and perform an action. This can be a specific user or group of users. The second is a permission, which defines what actions can be performed, like read or write.

Lifecycle policies



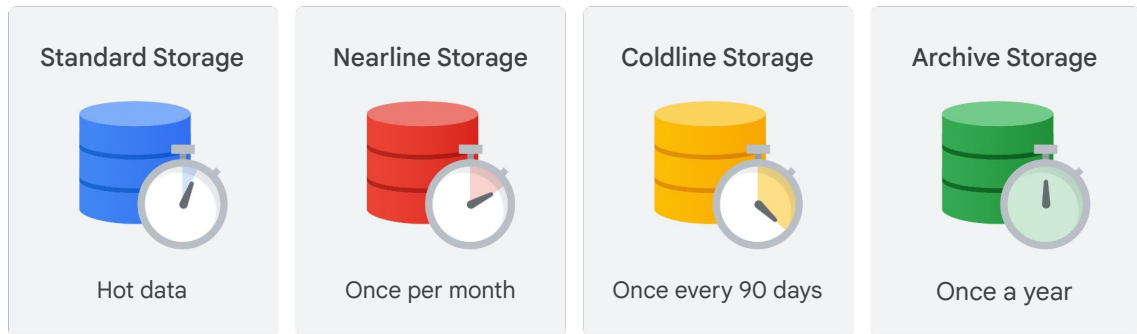
Examples

- ✓ Delete objects older than 365 days
- ✓ Delete objects created before MM/DD/YY
- ✓ Keep only the 3 most recent versions

Because storing and retrieving large amounts of object data can quickly become expensive, Cloud Storage also offers lifecycle management policies.

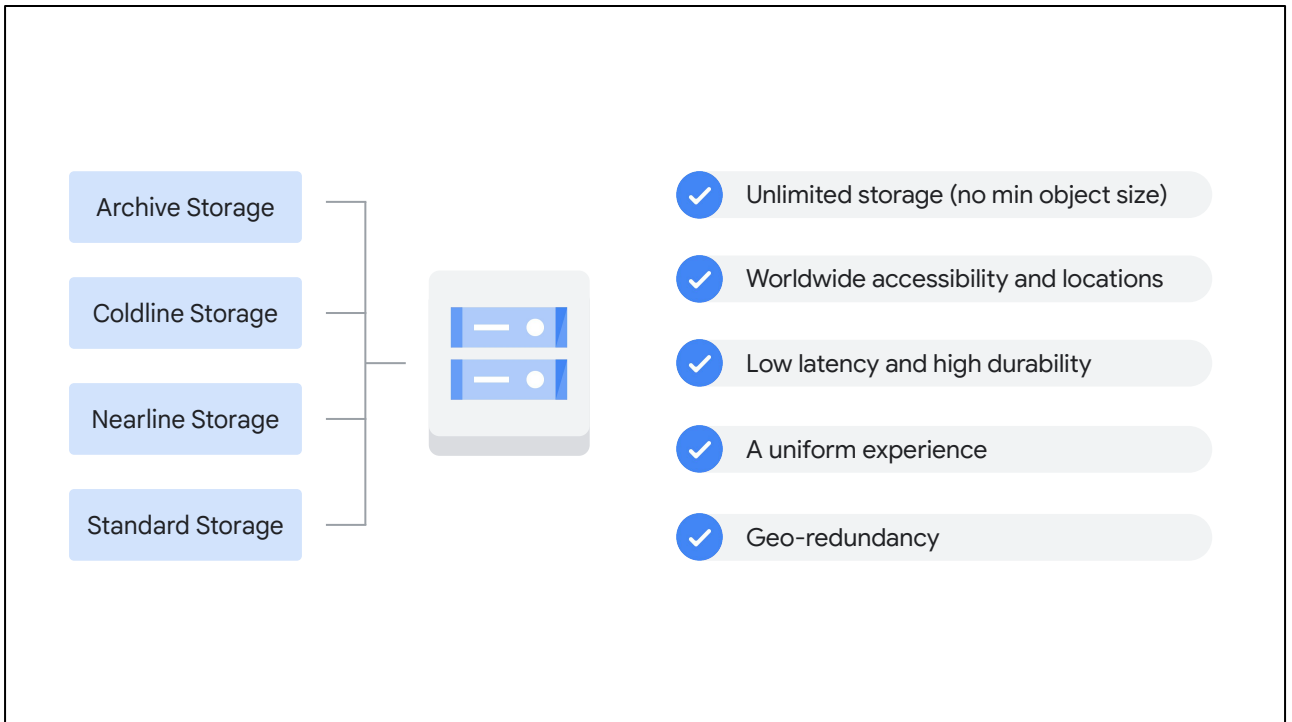
For example, you could tell Cloud Storage to delete objects older than 365 days; or to delete objects created before January 1, 2013; or to keep only the 3 most recent versions of each object in a bucket that has versioning enabled.

Having this control ensures that you're not paying for more than you actually need.



There are four primary storage classes in Cloud Storage:

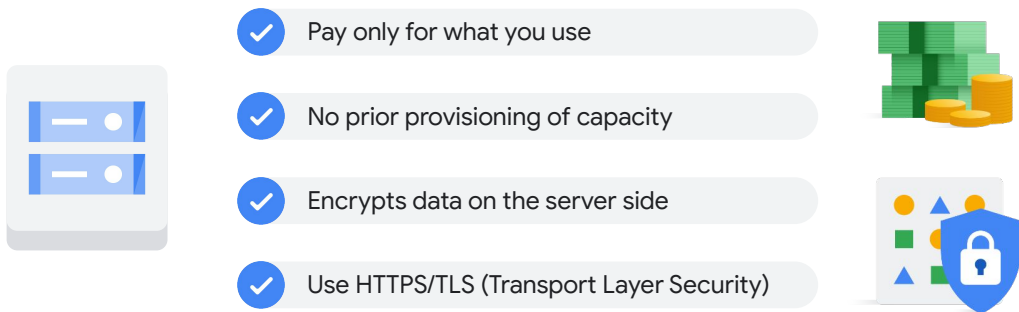
- The first is **Standard Storage**. Standard Storage is considered best for frequently accessed, or “hot,” data. It’s also great for data that’s stored for only brief periods of time.
- The second storage class is **Nearline Storage**. This is best for storing infrequently accessed data, like reading or modifying data on average once a month or less. Examples might include data backups, long-tail multimedia content, or data archiving.
- The third storage class is **Coldline Storage**. This is also a low-cost option for storing infrequently accessed data. However, as compared to Nearline Storage, Coldline Storage is meant for reading or modifying data, at most, once every 90 days.
- The fourth storage class is **Archive Storage**. This is the lowest-cost option, used ideally for data archiving, online backup, and disaster recovery. It’s the best choice for data that you plan to access less than once a year, because it has higher costs for data access and operations and a 365-day minimum storage duration.



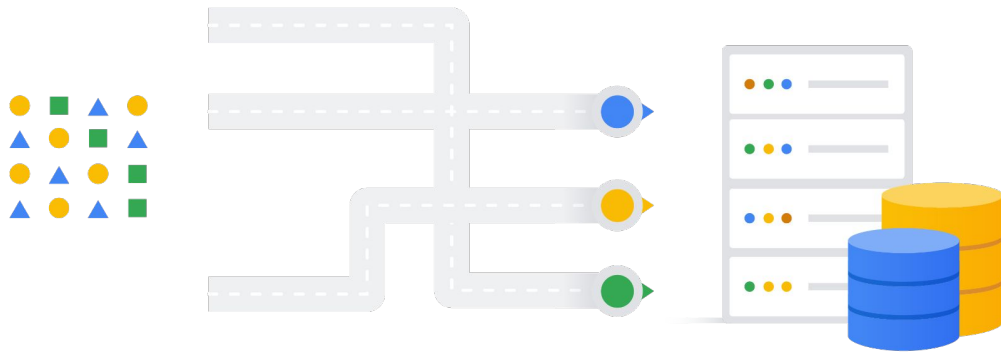
Although each of these four classes has differences, it's worth noting there are several characteristics that apply across *all* of these storage classes.

These include:

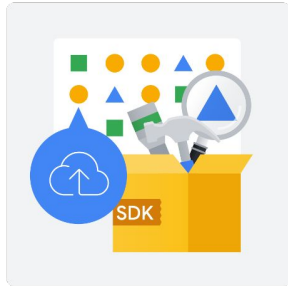
- Unlimited storage with no minimum object size requirement,
- Worldwide accessibility and locations,
- Low latency and high durability,
- A uniform experience, which extends to security, tools, and APIs, and,
- Geo-redundancy if data is stored in a multi-region or dual-region. This means placing physical servers in geographically diverse data centers to protect against catastrophic events and natural disasters, and load-balancing traffic for optimal performance.



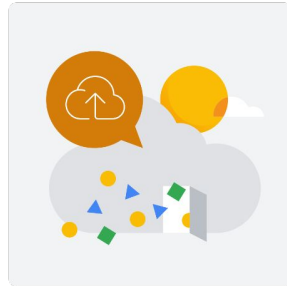
Cloud Storage has no minimum fee because you pay only for what you use, and prior provisioning of capacity isn't necessary. And from a security perspective, Cloud Storage always encrypts data on the server side, before it's written to disk, at no additional charge. Data traveling between a customer's device and Google is encrypted by default using HTTPS/TLS (Transport Layer Security).



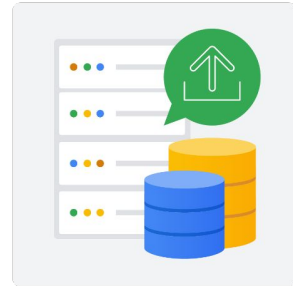
Regardless of which storage class you choose, there are several ways to bring data into Cloud Storage.



Online transfer



Storage Transfer Service



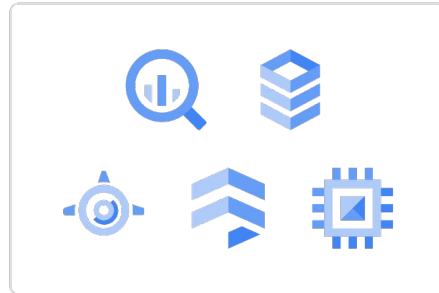
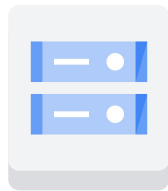
Transfer Appliance

Many customers simply carry out their own **online transfer** using `gcloud storage`, which is the Cloud Storage command from the Cloud SDK. Data can also be moved in by using a drag and drop option in the Google Cloud console, if accessed through the Google Chrome web browser. But what if you have to upload terabytes or even petabytes of data?

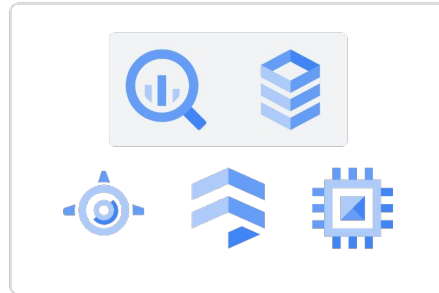
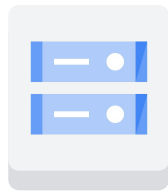
Storage Transfer Service enables you to import large amounts of online data into Cloud Storage quickly and cost-effectively.

The Storage Transfer Service lets you schedule and manage batch transfers to Cloud Storage from another cloud provider, from a different Cloud Storage region, or from an HTTP(S) endpoint.

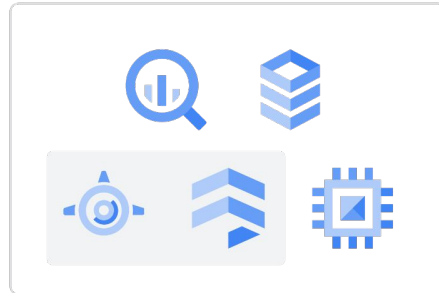
And then there is the **Transfer Appliance**, which is a rackable, high-capacity storage server that you lease from Google Cloud. You connect it to your network, load it with data, and then ship it to an upload facility where the data is uploaded to Cloud Storage. You can transfer up to a petabyte of data on a single appliance.



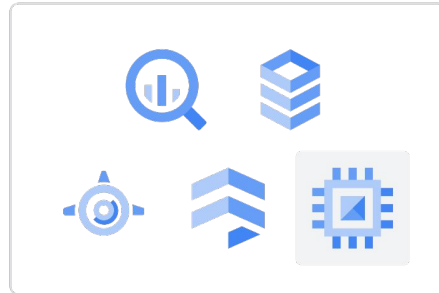
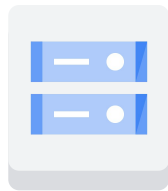
Cloud Storage's tight integration with other Google Cloud products and services means that there are many additional ways to move data into the service.



For example, you can import and export tables to and from both BigQuery and Cloud SQL. You can also store

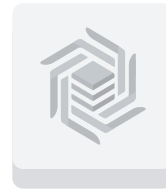
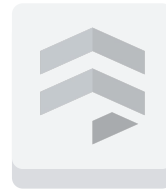
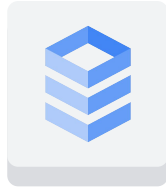
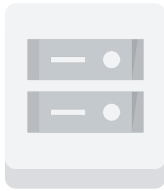


App Engine logs, Firestore backups, and objects used by App Engine applications, like images.

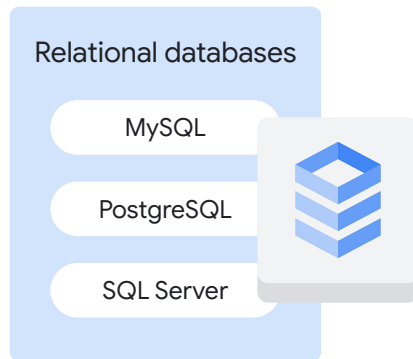


Cloud Storage can also store instance startup scripts, Compute Engine images, and objects used by Compute Engine applications.

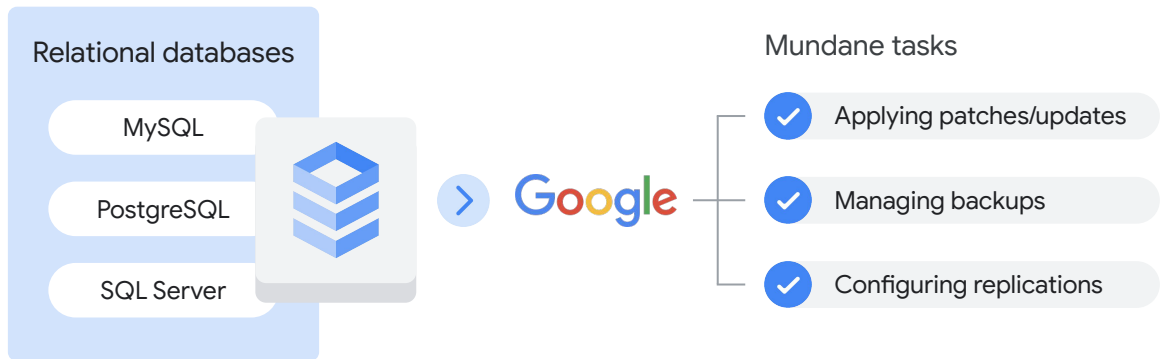
Cloud SQL



Google Cloud's second core storage option is **Cloud SQL**.



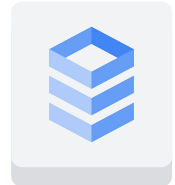
Cloud SQL offers fully managed relational databases, including MySQL, PostgreSQL, and SQL Server as a service.



It's designed to hand off mundane, but necessary and often time-consuming, tasks to Google—like applying patches and updates, managing backups, and configuring replications—so your focus can be on building great applications.

Cloud SQL:

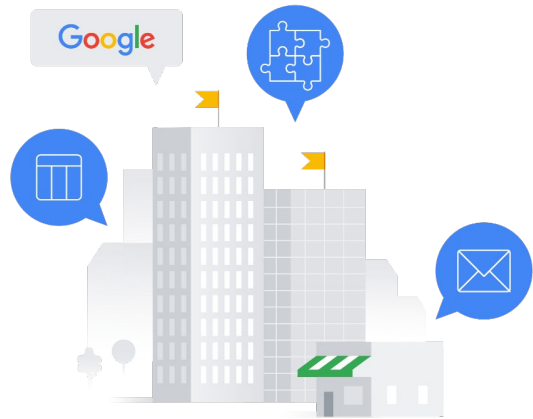
- Doesn't require any software installation or maintenance
- Can scale up to 64 processor cores, 400+ GB of RAM, and 30 TB of storage
- Supports automatic replication scenarios
- Supports managed backups – The cost of an instance covers 7 backups
- Encrypts customer data when on Google's internal networks and when stored in database tables, temporary files, and backups
- Includes a network firewall



Cloud SQL:

- Doesn't require any software installation or maintenance.
- Can scale up to 64 processor cores, 400+ GB of RAM, and 30 TB of storage.
- Supports *automatic replication* scenarios, such as from a Cloud SQL primary instance, an external primary instance, and external MySQL instances.
- Supports *managed backups*, so backed-up data is securely stored and accessible if a restore is required. The cost of an instance covers seven backups.
- Encrypts customer data when on Google's internal networks and when stored in database tables, temporary files, and backups.
- Includes a network firewall, which controls network access to each database instance.

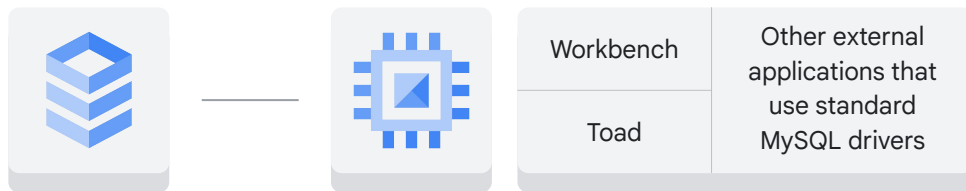
Cloud SQL instances are accessible by other Google Cloud and external services



A benefit of Cloud SQL instances is that they are accessible by other Google Cloud services, and even external services. Cloud SQL can be used with App Engine using standard drivers like Connector/J for Java or MySQLdb for Python.



Compute Engine instances can be authorized to access Cloud SQL instances and configure the Cloud SQL instance to be in the same zone as your virtual machine.



Cloud SQL also supports other applications and tools that you might use, like SQL Workbench, Toad, and other external applications using standard MySQL drivers.

Cloud Spanner



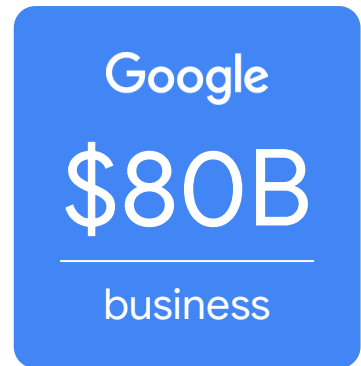
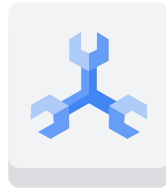
The third core storage option offered by Google Cloud is **Cloud Spanner**.

- ✓ Scales horizontally
- ✓ Strongly consistent
- ✓ Speaks SQL



Cloud Spanner is a fully managed relational database service that scales horizontally, is strongly consistent, and speaks SQL.

- ✓ Scales horizontally
- ✓ Strongly consistent
- ✓ Speaks SQL



Battle tested by Google's own mission-critical applications and services, Spanner is the service that powers Google's \$80 billion business.

Cloud Spanner is especially suited for applications that require:

SQL relational database management system with joins and secondary indexes

Built-in high availability

Strong global consistency

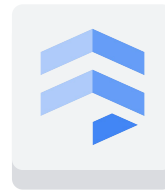
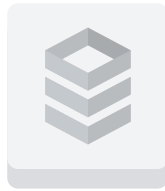
High numbers of input/output operations per second



Cloud Spanner is especially suited for applications that require:

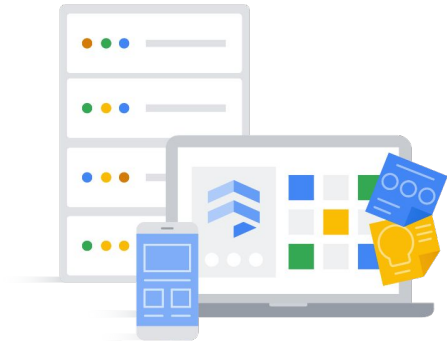
- A SQL relational database management system with joins and secondary indexes
- Built-in high availability
- Strong global consistency
- And high numbers of input and output operations per second. We're talking tens of thousands of reads and writes per second or more.

Firestore



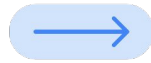
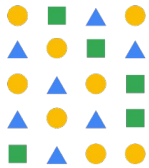
Google Cloud's fourth core storage option is **Firestore**.

- ✓ Flexible
- ✓ Horizontally scalable
- ✓ NoSQL cloud database

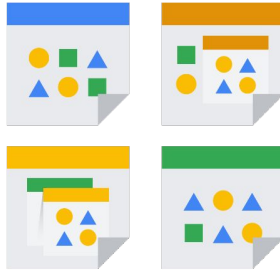


Firestore is a flexible, horizontally scalable, NoSQL cloud database for mobile, web, and server development.

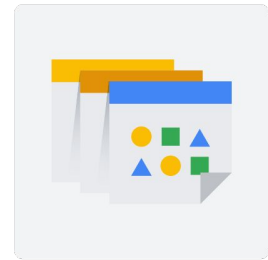
Incoming data



Documents

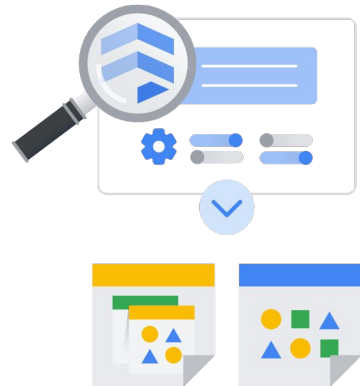


Collection



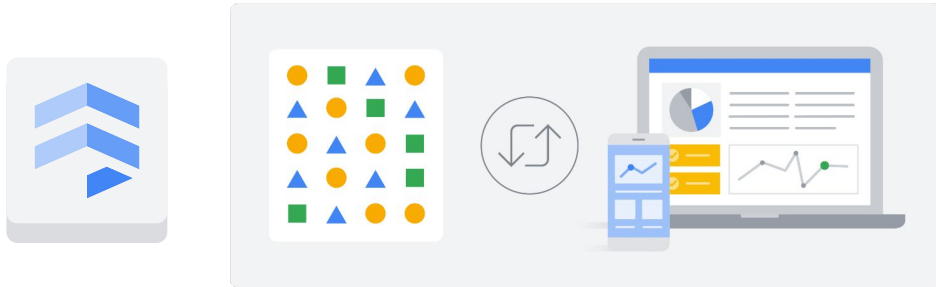
With Firestore, data is stored in *documents* and then organized into *collections*. Documents can contain complex nested objects in addition to subcollections.

- ✓ Retrieve individual documents
- ✓ Retrieve all the documents in a collection
- ✓ Can include multiple, chained filters
- ✓ Can combine filtering and sorting options
- ✓ Indexed by default

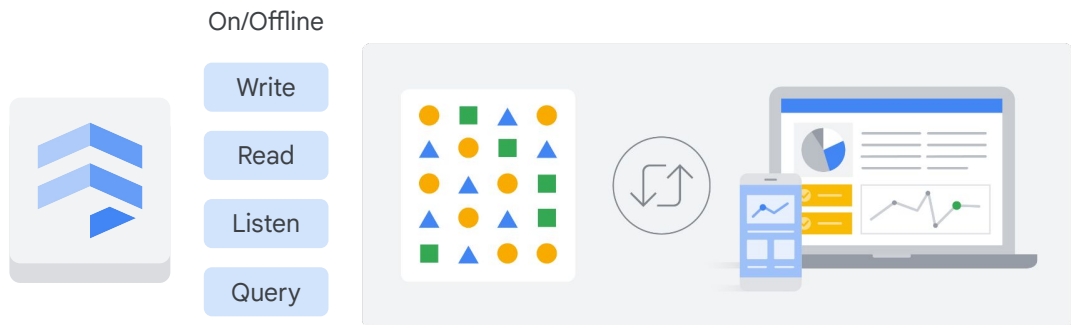


Firestore's NoSQL queries can then be used to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters.

Queries can include multiple, chained filters and combine filtering and sorting options. They're also indexed by default, so query performance is proportional to the size of the result set, not the dataset.



Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.



It caches data that an app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Firestore synchronizes any local changes back to Firestore.

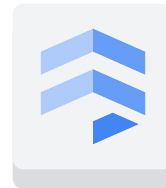
Google Cloud

✓ Automatic multi-region data replication

✓ Strong consistency guarantees

✓ Atomic batch operations

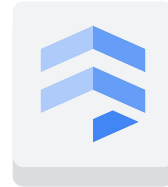
✓ Real transaction support



Firestore leverages Google Cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations and real transaction support.

When using Firestore, you're charged for:

- ✓ Reading, writing, and deleting documents
- ✓ Queries, which are counted as one document read
- ✓ The amount of database storage used
- ✓ The amount of network bandwidth used*



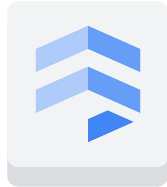
*Minus up to 10 GiB of free egress per month between regions in the US

From a pricing perspective, you're charged for each document read, write, and delete that you perform with Firestore.

Queries are also charged at the rate of one "document read" per query, whether the query returns data or not.

You're also charged for the amount of storage your data consumes and for certain kinds of network bandwidth used to access your data. Ingress is currently free, and in many cases so is egress.

Consult the pricing page for Firestore for details, or you can use Google's Billing Calculator to estimate prices for your particular use case.



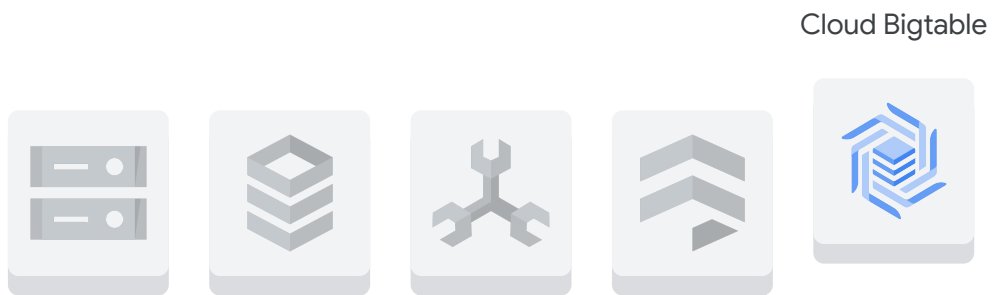
Firestore has a free quota per day of

- ✓ 50,000 document reads
- ✓ 20,000 document writes
- ✓ 20,000 document deletes
- ✓ 1 GB of stored data

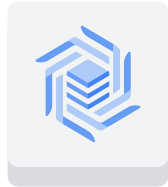
In addition to the 10GiB of free network egress per month between US regions, Firestore has a free quota per day of:

- 50,000 document reads
- 20,000 document writes
- 20,000 document deletes, and,
- 1 GB of stored data

Charges only begin once the free daily quota has been exceeded. This allows you to get started developing with Firestore for very little, or even for free.



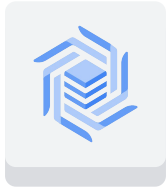
The last of Google Cloud's core storage options we're going to explore is **Cloud Bigtable**.



Cloud Bigtable

Google's NoSQL big
data database service

Cloud Bigtable is Google's NoSQL big data database service.

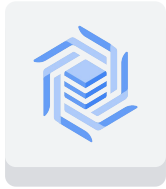


Cloud Bigtable

Google's NoSQL big data database service



It's the same database that powers many core Google services, including Search, Analytics, Maps, and Gmail.



Cloud Bigtable

Google's NoSQL big data database service

✓ Handle massive workloads

✓ Consistent low latency

✓ High throughput

✓ Great choice for:

✓ Operational applications

✓ Analytical applications

Bigtable is designed to handle massive workloads at consistent low latency and high throughput, so it's a great choice for both operational and analytical applications, including Internet of Things, user analytics, and financial data analysis.

Customers often choose **Bigtable** if:

They work with more than 1TB of semi-structured or structured data

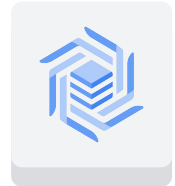
Data is fast with high throughput, or it's rapidly changing

They work with NoSQL data

Data is a time-series or has natural semantic ordering

They work with big data, running asynchronous batch or synchronous real-time processing on the data

They run machine learning algorithms on the data

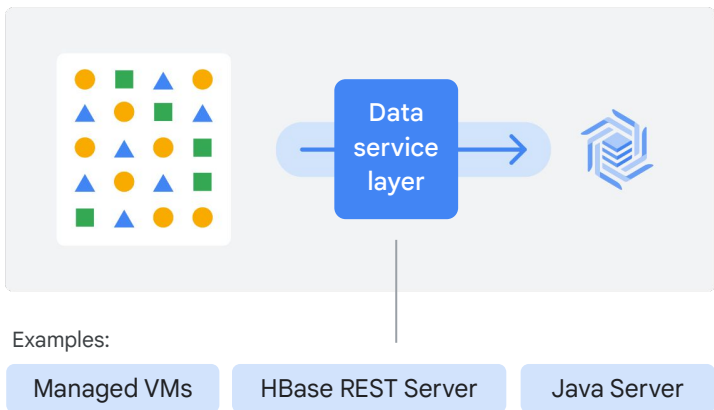
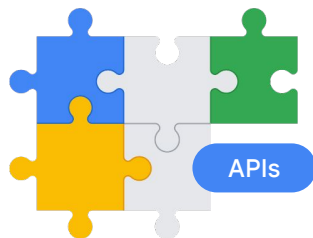


When deciding which storage option is best, customers often choose Bigtable if:

- They're working with more than 1TB of semi-structured or structured data.
- Data is fast with high throughput, or it's rapidly changing.
- They're working with NoSQL data. This usually means transactions where strong relational semantics are not required.
- Data is a time-series or has natural semantic ordering.
- They're working with big data, running asynchronous batch or synchronous real-time processing on the data.
- Or they're running machine learning algorithms on the data.

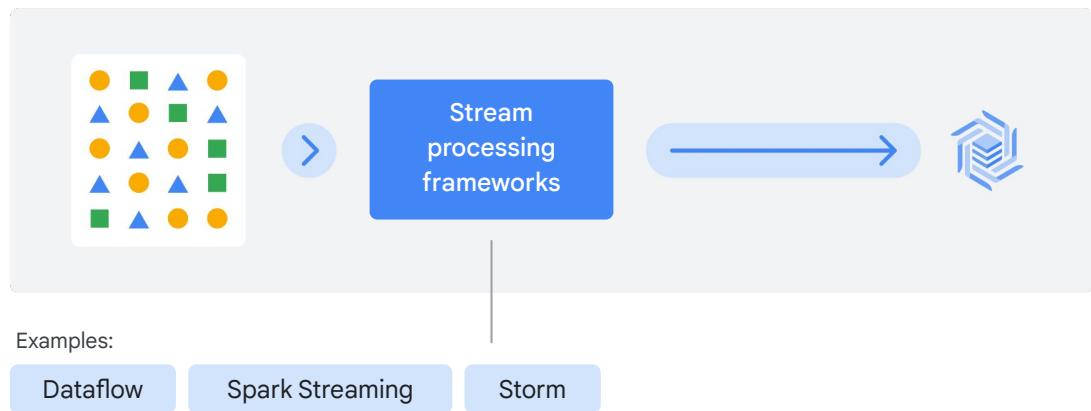
Cloud Bigtable **can interact** with other
Google Cloud services and third-party clients

Cloud Bigtable can interact with other Google Cloud services and third-party clients.

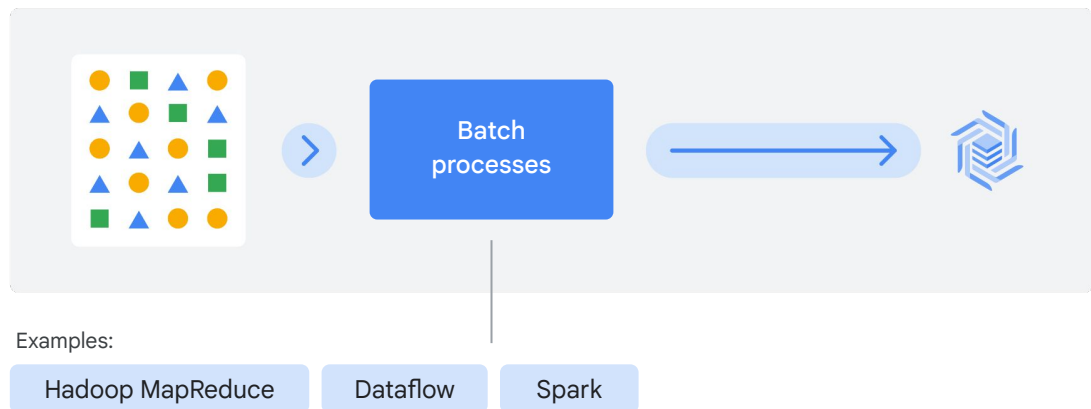


Using **APIs**, data can be read from and written to Cloud Bigtable through a data service layer like Managed VMs, the HBase REST Server, or a Java Server using the HBase client.

Typically this is used to serve data to applications, dashboards, and data services.



Data can also be **streamed** in through a variety of popular stream processing frameworks like Dataflow Streaming, Spark Streaming, and Storm.



And if streaming is not an option, data can also be read from and written to Cloud Bigtable through batch processes like Hadoop MapReduce, Dataflow, or Spark.

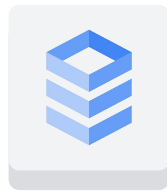
Often, summarized or newly calculated data is written back to Cloud Bigtable or to a downstream database.



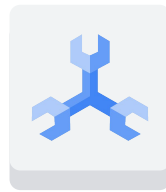
Cloud Storage



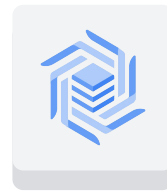
Firestore



Cloud SQL



Cloud Spanner



Cloud Bigtable

Now that we've covered Google Cloud's core storage options,

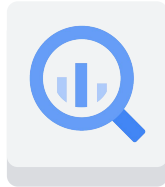
Option	Best for	Capacity
Cloud Storage	Storing immutable blobs larger than 10 MB	Petabytes Max unit size: 5 TB per object
Cloud SQL	<ul style="list-style-type: none"> Full SQL support for an online transaction processing system Web frameworks and existing applications 	Up to 64 TB
Spanner	<ul style="list-style-type: none"> Full SQL support for an online transaction processing system Horizontal scalability 	Petabytes
Firestore	Massive scaling and predictability together with real time query results and offline query support	Terabytes Max unit size: 1 MB per entity
Cloud Bigtable	<ul style="list-style-type: none"> Storing large amount of structured objects Does not support SQL queries and multi-row transactions Analytical data with heavy read and write events 	Petabytes Max unit size: 10 MB p/cell, 100 MB p/row

let's do a comparison to help highlight the most suitable service for a specific application or workflow. Consider using **Cloud Storage** if you need to store immutable blobs larger than 10 megabytes, such as large images or movies. This storage service provides petabytes of capacity with a maximum unit size of 5 terabytes per object.

Consider using **Cloud SQL** or **Cloud Spanner** if you need full SQL support for an online transaction processing system. Cloud SQL provides up to 64 terabytes, depending on machine type, and Cloud Spanner provides petabytes. Cloud SQL is best for web frameworks and existing applications, like storing user credentials and customer orders. If Cloud SQL doesn't fit your requirements because you need horizontal scalability, not just through read replicas, consider using **Cloud Spanner**.

Consider **Firestore** if you need massive scaling and predictability together with real time query results and offline query support. This storage service provides terabytes of capacity with a maximum unit size of 1 megabyte per entity. Firestore is best for storing, syncing, and querying data for mobile and web apps.

Finally, consider using **Cloud Bigtable** if you need to store a large number of structured objects. Cloud Bigtable doesn't support SQL queries, nor does it support multi-row transactions. This storage service provides petabytes of capacity with a maximum unit size of 10 megabytes per cell and 100 megabytes per row. Bigtable is best for analytical data with heavy read and write events, like AdTech, financial, or IoT data. Depending on your application, it's possible that you might use one, or several, of these services to do the job.



BigQuery

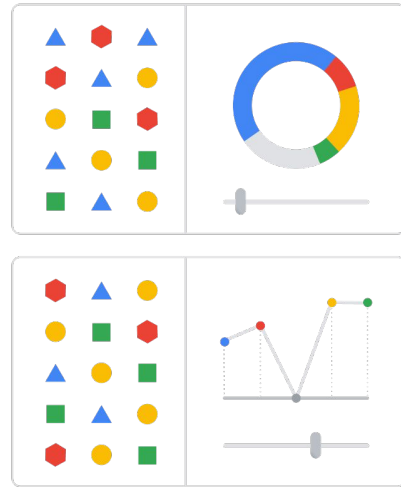
On the edge between data storage and data processing

You may have noticed that BigQuery hasn't been mentioned in this section of the course. This is because it sits on the edge between data storage and data processing, and is covered in more depth in other courses.



BigQuery

On the edge between data storage and data processing



The usual reason to store data in BigQuery is so you can use its big data analysis and interactive querying capabilities, but it's not purely a data storage product.