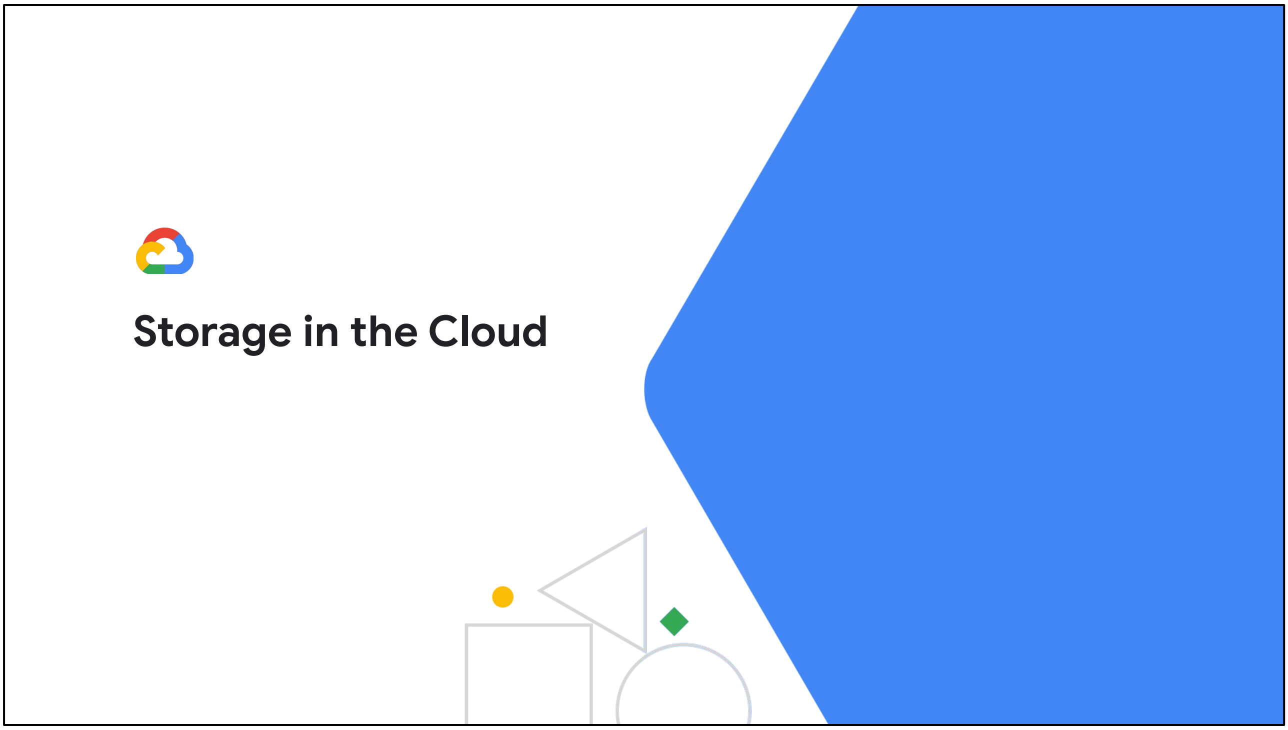
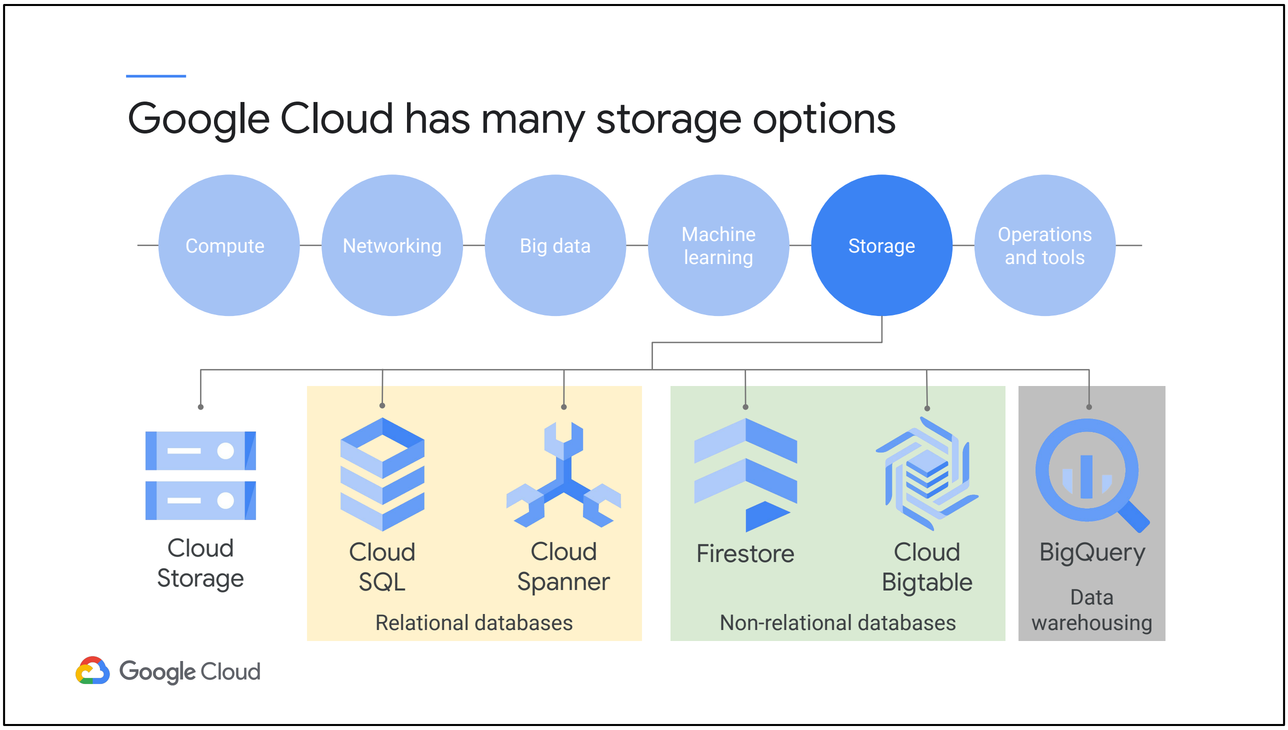


Storage in the Cloud

Every application needs to store data. Maybe... media to be streamed. Maybe... sensor data from devices. Maybe... customer account balances. Maybe the fact that my Dragonite has more than 2600 C.P.! Different applications and workloads require different storage and database solutions.



Google Cloud has many storage options

Compute

Networking

Big data

Machine learning

Storage

Operations and tools

Cloud Storage

Cloud SQL

Cloud Spanner

Relational databases

Firestore

Cloud Bigtable

Non-relational databases

BigQuery

Data warehousing

Google Cloud

Google Cloud has many storage options that satisfy nearly every customer use case. In this module, we turn our attention to the core storage options: Cloud Storage, Cloud SQL, Cloud Spanner, Firestore, and Cloud Bigtable.

Agenda

Cloud Storage

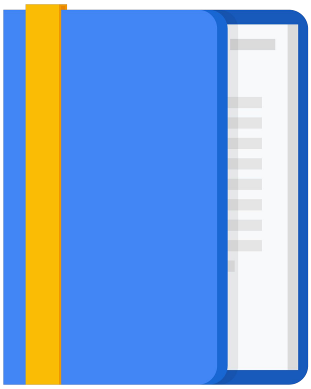
Cloud Bigtable

Cloud SQL, Cloud Spanner, and Firestore

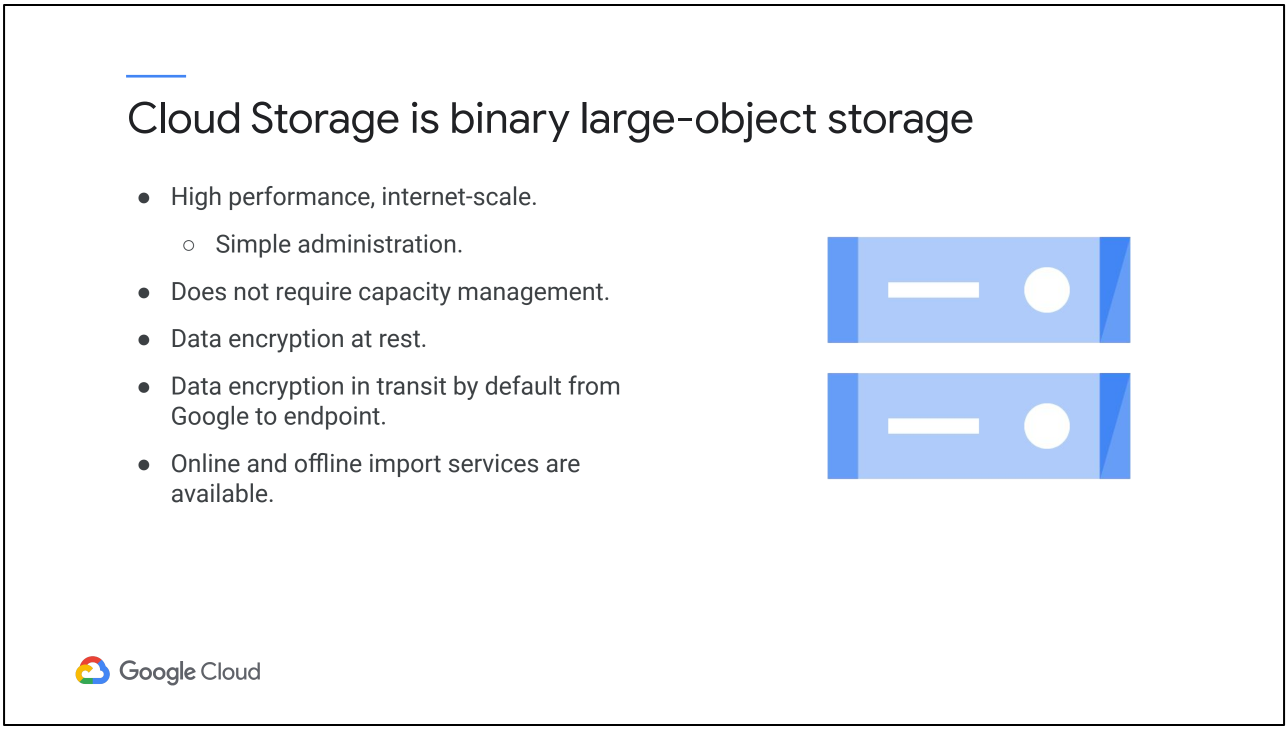
Comparing Storage Options

Quiz and Lab

Resources



Google Cloud



Cloud Storage is binary large-object storage

- High performance, internet-scale.
  - Simple administration.
- Does not require capacity management.
- Data encryption at rest.
- Data encryption in transit by default from Google to endpoint.
- Online and offline import services are available.

Cloud Storage offers developers and IT organizations durable and highly available object storage. It assesses no minimum fee; you pay only for what you use. Prior provisioning of capacity isn’t necessary.

What’s object storage? It’s not the same as file storage, in which you manage your data as a hierarchy of folders. It’s not the same as block storage, in which your operating system manages your data as chunks of disk. Instead, object storage means this: you say to your storage, “Here, keep this arbitrary sequence of bytes,,” and the storage lets you address it with a unique key. In Cloud Storage and in other systems, these unique keys are in the form of URLs, which means object storage interacts well with web technologies.

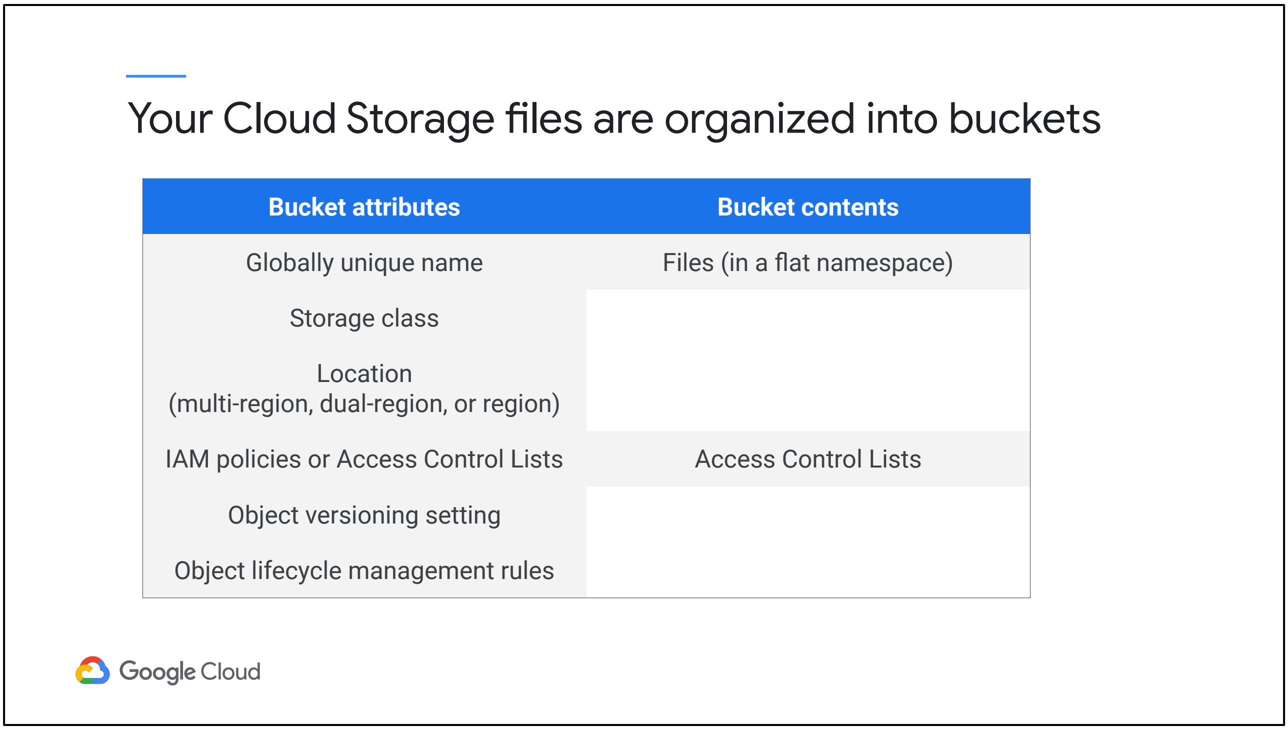
Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge. Data traveling between a customer’s device and Google is encrypted by default using HTTPS/TLS (Transport Layer Security). In fact, Google was the first major cloud provider to enable HTTPS/TLS by default.

Cloud Storage is not a file system, although it can be accessed as one via third-party tools such as Cloud Storage FUSE. The storage objects offered by Cloud Storage are “immutable,” which means that you do not edit them in place, but instead create a new version. Cloud Storage’s primary use is whenever binary large-object storage is needed: online content, backup and archiving, storage of intermediate results in processing workflows, and more.

Offline Media Import/Export is a third-party solution that allows you to load data into Cloud Storage by sending your physical media, such as hard disk drives (HDDs), tapes, and USB flash drives, to a third-party service provider who uploads data on your behalf. Offline Media Import/Export is helpful if you're limited to a slow, unreliable, or expensive internet connection.

Offline import is available through third-party providers:
https://cloud.google.com/storage/docs/offline-media-import-export
Cloud Storage Transfer Service enables you to import large amounts of online data into Cloud Storage quickly and cost-effectively. To use Cloud Storage Transfer Service, you set up a transfer from a data source to data sink. Data sources can be an Amazon Simple Storage Service (Amazon S3) bucket, an HTTP/HTTPS location, or another Cloud Storage bucket. Data sinks are always a Cloud Storage bucket.

Example uses of Cloud Storage Transfer Service include:
Backing up data to a Cloud Storage bucket from other storage providers.
Moving data from a Standard Storage bucket to a Nearline Storage bucket to lower your storage costs.



Your Cloud Storage files are organized into buckets

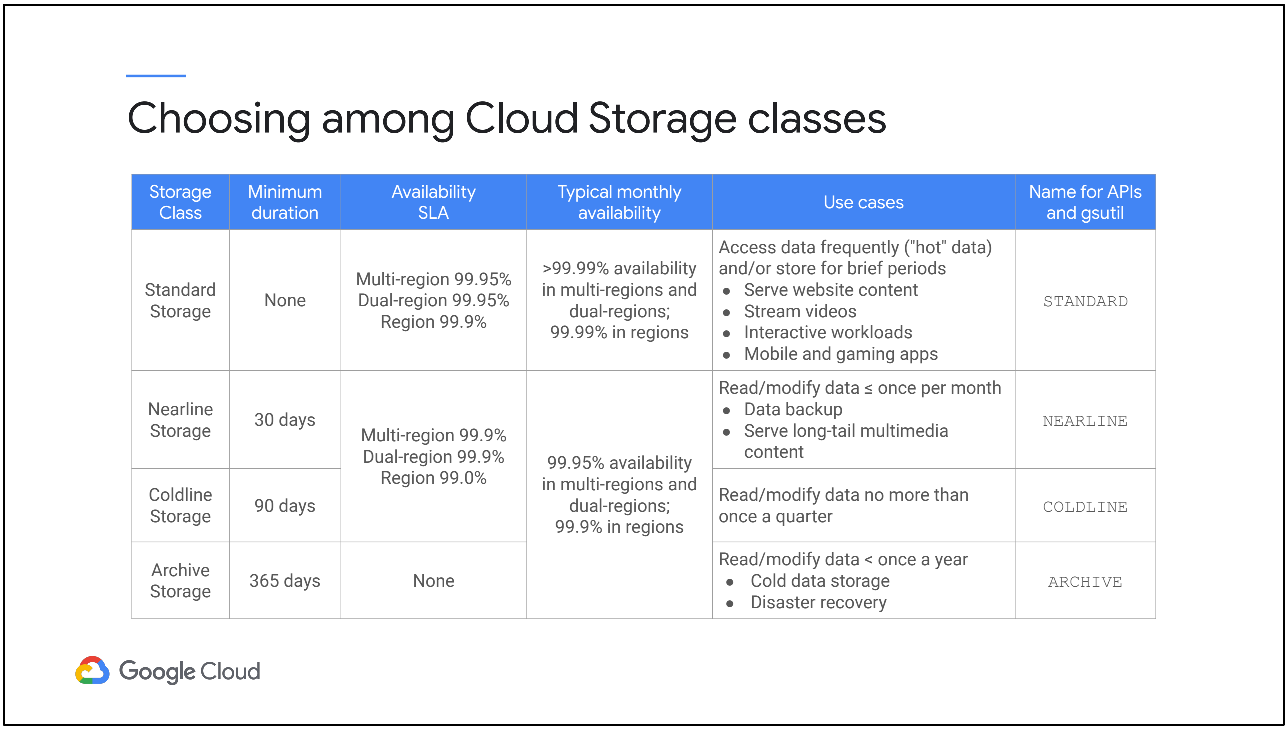
Bucket attributes	Bucket contents
Globally unique name	Files (in a flat namespace)
Storage class	
Location (multi-region, dual-region, or region)	
IAM policies or Access Control Lists	Access Control Lists
Object versioning setting	
Object lifecycle management rules	

Your Cloud Storage files are organized into buckets. When you create a bucket: you give it a globally-unique name; you specify a geographic location where the bucket and its contents are stored; and you choose a default storage class. Pick a location that minimizes latency for your users. For example, if most of your users are in Europe, you probably want to pick a European location: a Google Cloud region in Europe, or else the EU multi-region.

There are several ways to control users' access to your objects and buckets. For most purposes, Cloud IAM is sufficient. Roles are inherited from project to bucket to object. If you need finer control, you can create access control lists ("ACLs") that offer finer control, ACLs define who has access to your buckets and objects, as well as what level of access they have. Each ACL consists of two pieces of information: A scope, which defines who can perform the specified actions (for example, a specific user or group of users). And a permission, which defines what actions can be performed (for example, read or write).

Remember that Cloud Storage objects are immutable. You can turn on object versioning on your buckets if you want. If you do, Cloud Storage keeps a history of modifications--that is, overwrites or deletes--of all objects in the bucket. You can list the archived versions of an object, restore an object to an older state, or permanently delete a version, as needed. If you don't turn on object versioning, new always overwrites old.

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.



Choosing among Cloud Storage classes

Storage Class	Minimum duration	Availability SLA	Typical monthly availability	Use cases	Name for APIs and gsutil
Standard Storage	None	Multi-region 99.95% Dual-region 99.95% Region 99.9%	>99.99% availability in multi-regions and dual-regions; 99.99% in regions	Access data frequently ("hot" data) and/or store for brief periods • Serve website content • Stream videos • Interactive workloads • Mobile and gaming apps	`STANDARD`
Nearline Storage	30 days	Multi-region 99.9% Dual-region 99.9% Region 99.0%	99.95% availability in multi-regions and dual-regions; 99.9% in regions	Read/modify data ≤ once per month • Data backup • Serve long-tail multimedia content	`NEARLINE`
Coldline Storage	90 days			Read/modify data no more than once a quarter	`COLDLINE`
Archive Storage	365 days	None		Read/modify data < once a year • Cold data storage • Disaster recovery	`ARCHIVE`

Google Cloud

Google Cloud Storage has four primary storage classes, with different characteristics, use cases, and prices for your needs.

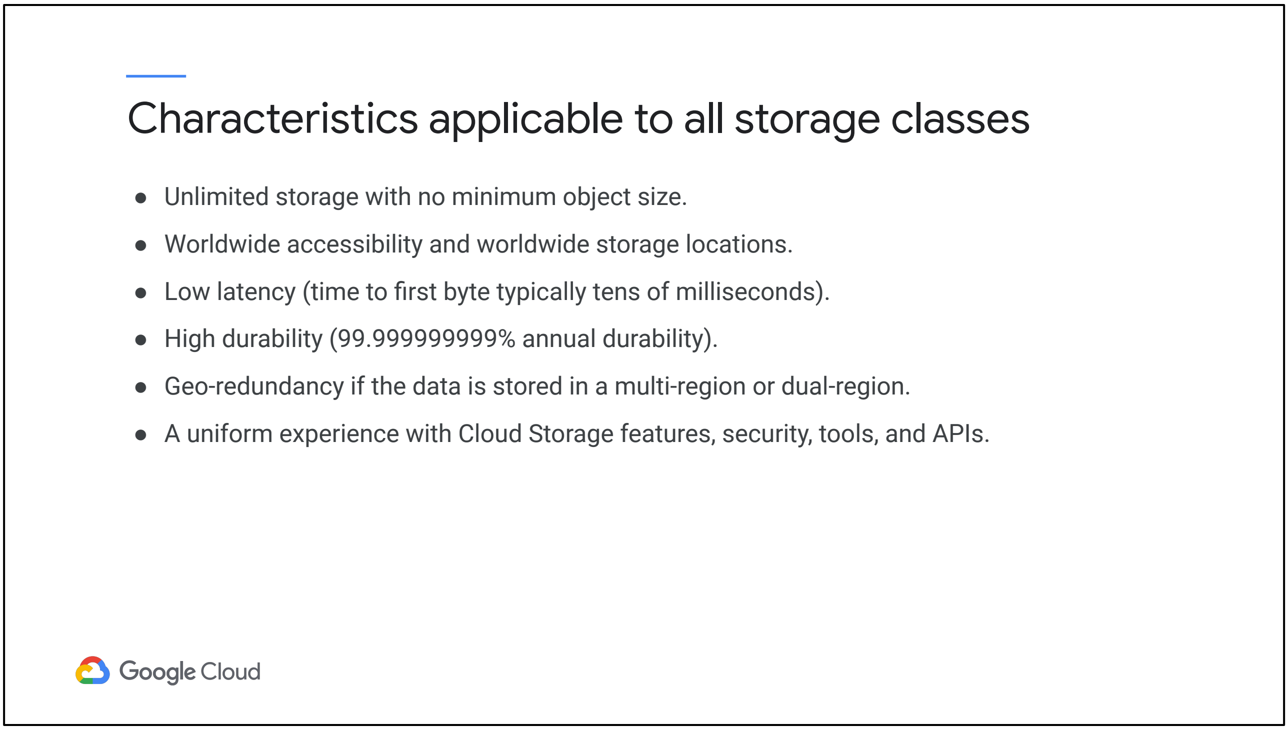
Standard Storage is best for data that is frequently accessed ("hot" data) and/or stored for only brief periods of time. When used in a region, co-locating your resources maximizes the performance for data-intensive computations and can reduce network charges. When used in a dual-region, you still get optimized performance when accessing Google Cloud products that are located in one of the associated regions, but you also get the improved availability that comes from storing data in geographically separate locations. When used in a multi-region, Standard Storage is appropriate for storing data that is accessed around the world, such as serving website content, streaming videos, executing interactive workloads, or serving data supporting mobile and gaming applications.

Nearline Storage is a low-cost, highly durable storage service for storing infrequently accessed data. Nearline Storage is a better choice than Standard Storage in scenarios where slightly lower availability, a 30-day minimum storage duration, and costs for data access are acceptable trade-offs for lowered at-rest storage costs. Nearline Storage is ideal for data you plan to read or modify on average once per month or less. Nearline Storage is appropriate for data backup, long-tail multimedia content, and data archiving.

Coldline Storage is a very-low-cost, highly durable storage service for storing infrequently accessed data. Coldline Storage is a better choice than Standard Storage or Nearline Storage in scenarios where slightly lower availability, a 90-day minimum storage duration, and higher costs for data access are acceptable trade-offs for lowered at-rest storage costs. Coldline Storage is ideal for data you plan to read or modify at most once a quarter.

Archive Storage is the lowest-cost, highly durable storage service for data archiving, online backup, and disaster recovery. Archive Storage has higher costs for data access and operations, as well as a 365-day minimum storage duration. Archive Storage is the best choice for data that you plan to access less than once a year. For example, cold data storage, such as data stored for legal or regulatory reasons, and disaster recovery.

For more information, see: https://cloud.google.com/storage/docs/storage-classes



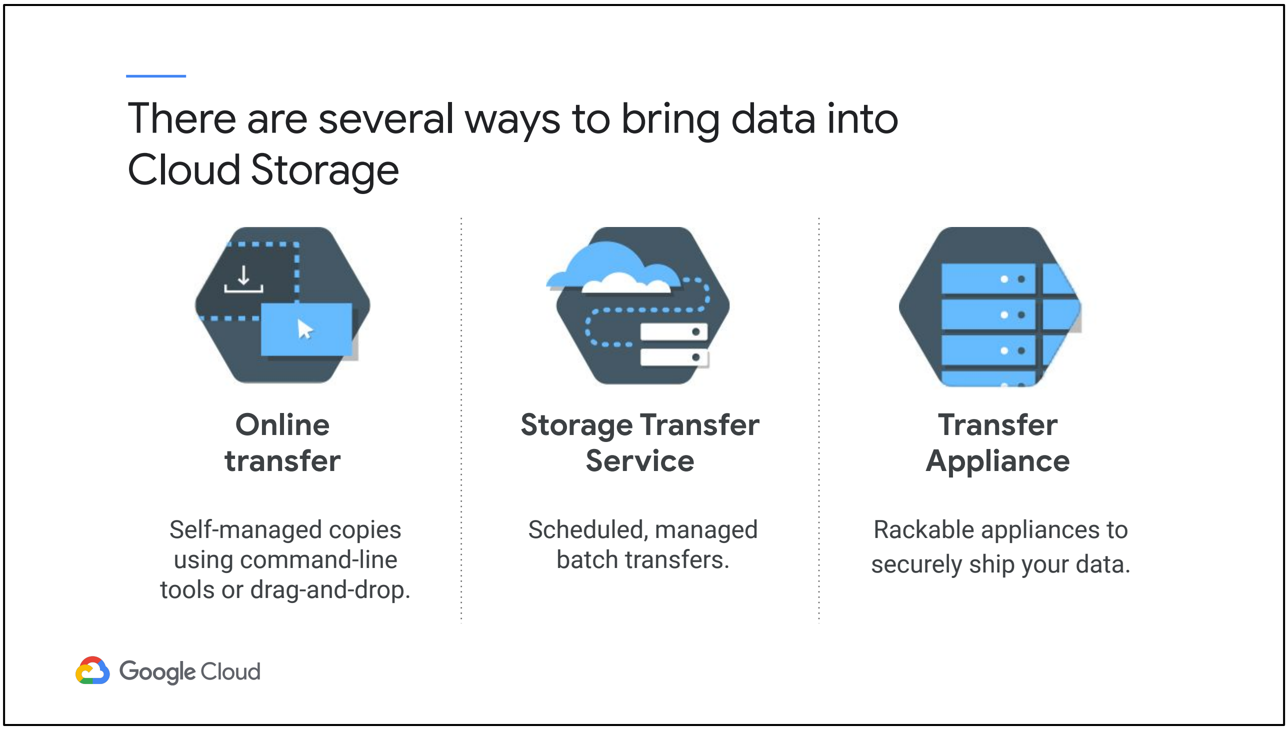
Characteristics applicable to all storage classes

- Unlimited storage with no minimum object size.
- Worldwide accessibility and worldwide storage locations.
- Low latency (time to first byte typically tens of milliseconds).
- High durability (99.999999999% annual durability).
- Geo-redundancy if the data is stored in a multi-region or dual-region.
- A uniform experience with Cloud Storage features, security, tools, and APIs.

Google Cloud

We've discussed the four primary storage classes and differentiated between them in terms of characteristics, availability and use cases. It is worth noting that there are a number of characteristics that apply across all storage classes. These include:

- Unlimited storage with no minimum object size requirement,
- Worldwide accessibility and locations,
- Low latency and high durability,
- Geo-redundancy if data is stored in a multi-region or dual-region, and
- A uniform experience, which extends to security, tools, and APIs.



There are several ways to bring data into Cloud Storage

Online transfer

Self-managed copies using command-line tools or drag-and-drop.

Storage Transfer Service

Scheduled, managed batch transfers.

Transfer Appliance

Rackable appliances to securely ship your data.

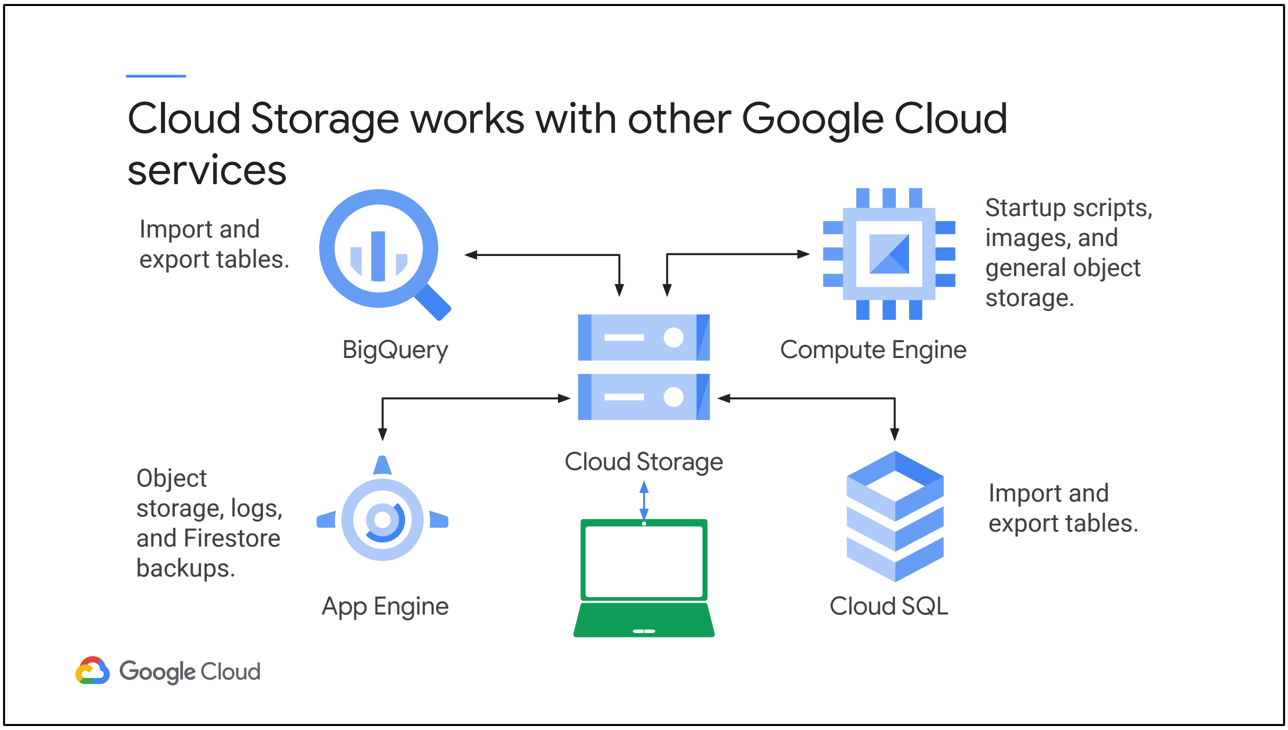
Google Cloud

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

Many customers simply use gsutil, which is the Cloud Storage command from the Cloud SDK. You can also move data in with a drag and drop in the Cloud Console, if you use the Google Chrome browser. But what if you have to upload terabytes or even petabytes of data? Google Cloud offers the online Storage Transfer Service and the offline Transfer Appliance to help.

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.

The Transfer Appliance is a rackable, high-capacity storage server that you lease from Google Cloud. You simply 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. The service enables you to securely transfer up to a petabyte of data on a single appliance. As of this recording, it's still beta, and it's not available everywhere, so check the website for details.



Cloud Storage works with other Google Cloud services

Import and export tables.

BigQuery

Compute Engine

Startup scripts, images, and general object storage.

Cloud Storage

Object storage, logs, and Firestore backups.

App Engine

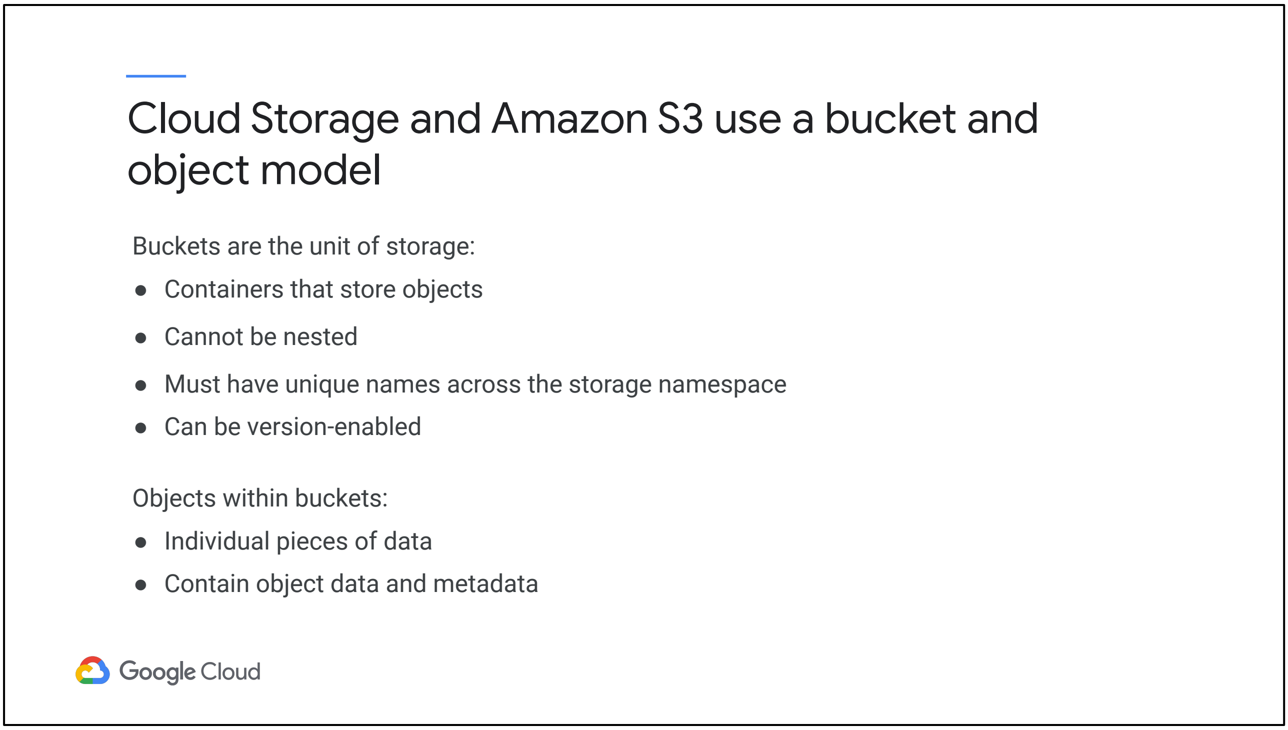
Cloud SQL

Import and export tables.

There are other ways of getting your data into Cloud Storage, as this storage option is tightly integrated with many of the Google Cloud products and services.

For example, you can import and export tables from and to BigQuery, as well as 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.

In short, Cloud Storage is often the ingestion point for data being moved into the cloud, and is frequently the long-term storage location for data.



Cloud Storage and Amazon S3 use a bucket and object model

Buckets are the unit of storage:

- Containers that store objects
- Cannot be nested
- Must have unique names across the storage namespace
- Can be version-enabled

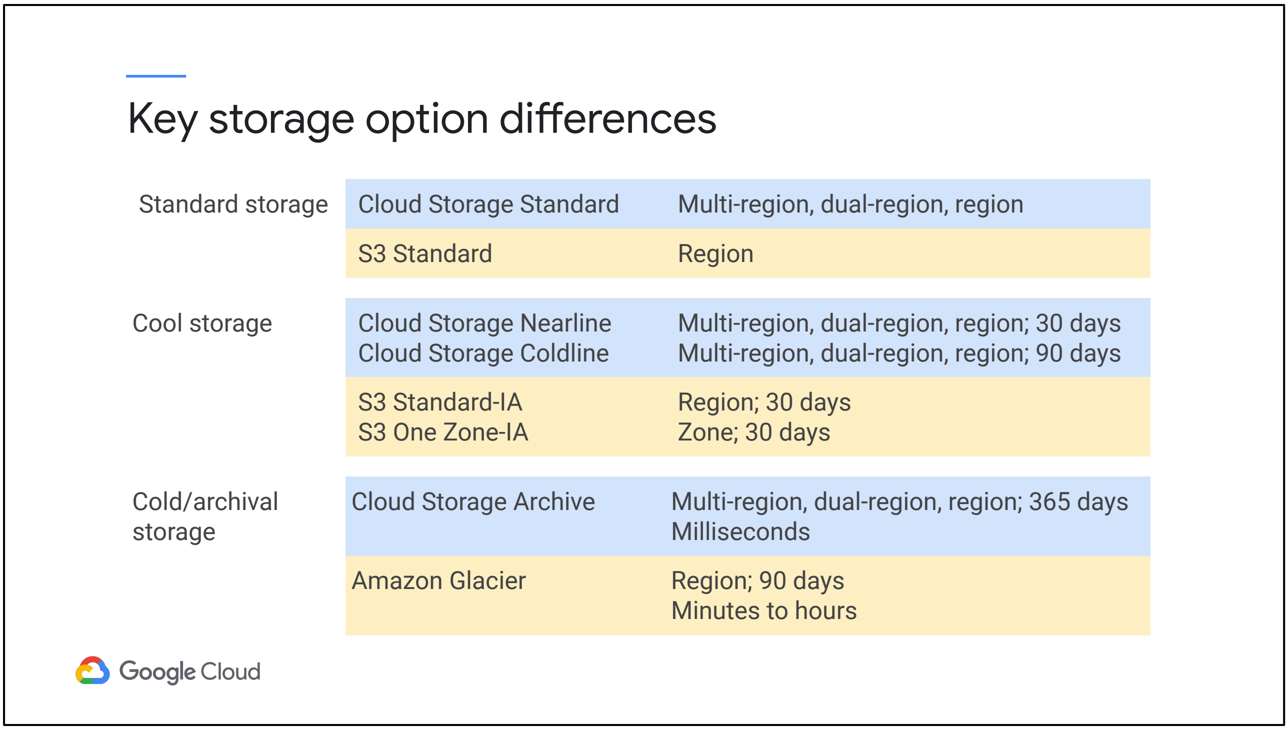
Objects within buckets:

- Individual pieces of data
- Contain object data and metadata

When you compare their respective storage services, Cloud Storage and Amazon Simple Storage Service (Amazon S3) have a lot of similarities.

In both services, you store objects in a bucket. Buckets cannot be nested, must be given globally unique names, and can be version-protected.

Objects are individual files inside a bucket and can be any file type, but they cannot be larger than 5 terabytes. However, storage capacity is virtually unlimited. Objects can also have metadata for information such as object size, date of last modification, and media type.



Key storage option differences

Standard storage	Cloud Storage Standard	Multi-region, dual-region, region
	S3 Standard	Region
Cool storage	Cloud Storage Nearline Cloud Storage Coldline	Multi-region, dual-region, region; 30 days Multi-region, dual-region, region; 90 days
	S3 Standard-IA S3 One Zone-IA	Region; 30 days Zone; 30 days
Cold/archival storage	Cloud Storage Archive	Multi-region, dual-region, region; 365 days Milliseconds
	Amazon Glacier	Region; 90 days Minutes to hours

Google Cloud

Let’s highlight some of the key differences between Cloud Storage and the storage options provided by AWS.

For standard distributed object storage requirements, the main difference between Cloud Storage Standard and Amazon S3 Standard lies in deployment locality. Where S3 Standard is regional, Cloud Storage Standard is multi-regional, dual-regional, and regional.

Cloud Storage and Amazon S3 each offer a reduced-cost storage class for data that does not require the availability of the standard storage tier. Cloud Storage offers Cloud Storage Nearline and Cloud Storage Coldline, and Amazon S3 offers Standard - Infrequent Access and One Zone - Infrequent Access. Again, the key difference lies in deployment locality. While Nearline and Coldline are multi-regional, dual-regional, and regional, S3 Standard IA is regional, and S3 One Zone IA is zonal. Coldline provides a lower storage cost option subject to a minimum duration of 90 days instead of 30 days.

Google and Amazon each offer cold storage options for data that does not need to be accessed regularly or retrieved quickly. Cloud Storage offers an additional class called Cloud Storage Archive, and Amazon offers Amazon Glacier. In addition to the deployment locality difference again, a key difference is that first-byte latency for Cloud Storage Archive is milliseconds – the same as it is for Cloud Storage Standard. The latency for Amazon Glacier can range from minutes to hours. The lower storage cost for Archive storage is 365 days as opposed to the 90 days for Amazon Glacier.

Agenda

Cloud Storage

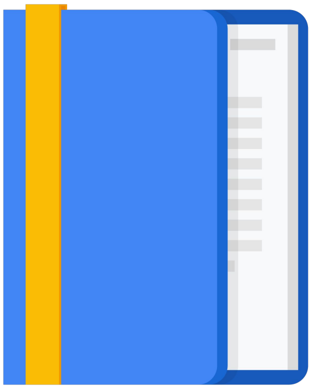
Cloud Bigtable

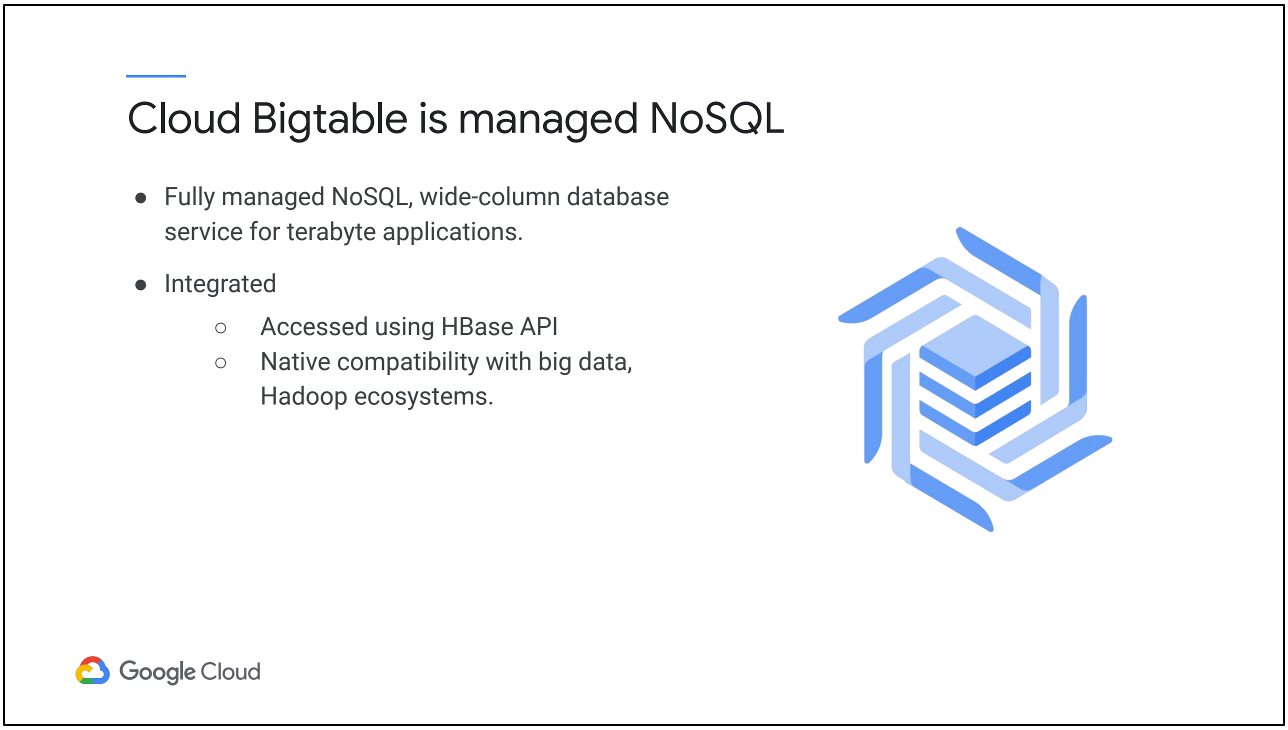
Cloud SQL, Cloud Spanner, and Firestore

Comparing Storage Options

Quiz and Lab

Resources





Cloud Bigtable is managed NoSQL

- Fully managed NoSQL, wide-column database service for terabyte applications.
- Integrated
  - Accessed using HBase API
  - Native compatibility with big data, Hadoop ecosystems.

Google Cloud

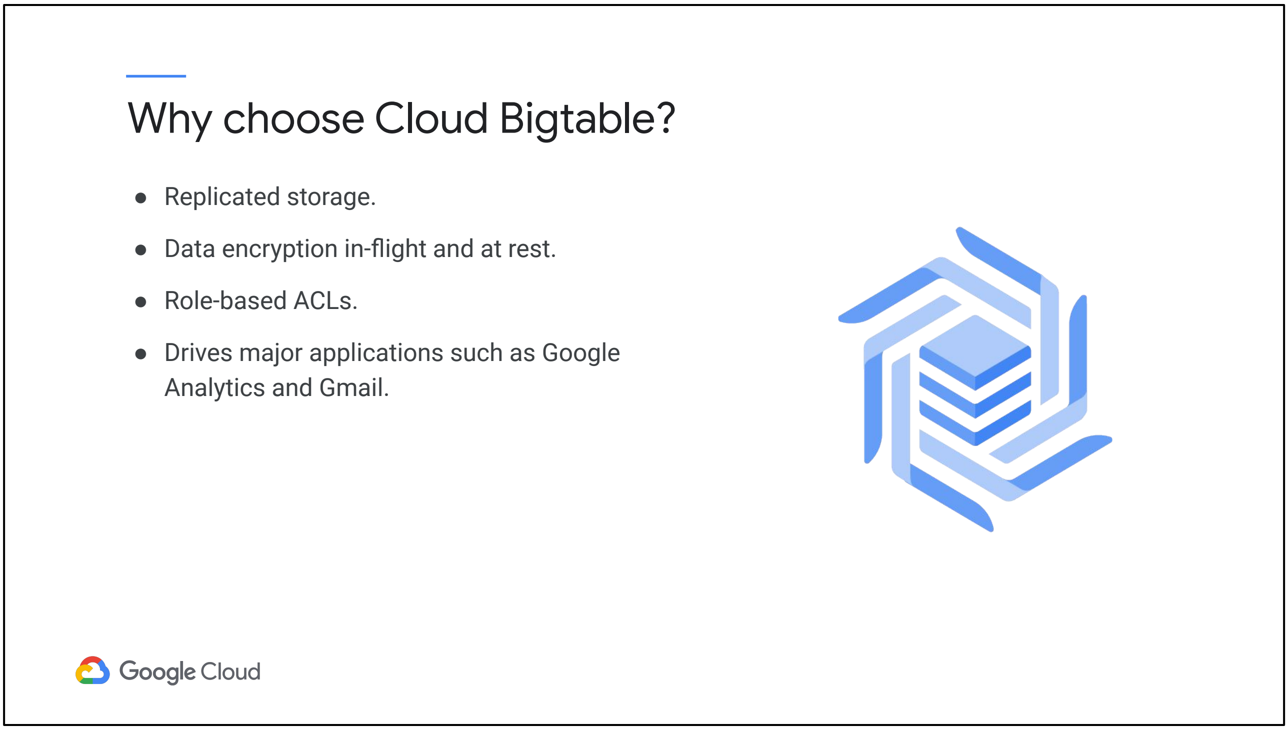
Cloud Bigtable is Google's NoSQL big data database service. What does NoSQL mean? Well, this isn't a database course, so I'll give you a very informal picture. Think of a relational database as offering you tables in which every row has the same set of columns, and the database engine enforces that rule, and other rules you specify for each table. That's called the database schema. A rigorously enforced, infrequently changing schema is a big help for some applications-- and a huge pain for others!

Some applications call for a much more flexible approach, for example, a NoSQL schema. In other words, for these applications, not all rows might need to have the same columns, and in fact the database might be designed to take advantage of that by sparsely populating the rows. That's part of what makes a NoSQL database what it is.

Which brings us to Bigtable. Your databases in Bigtable are sparsely populated tables that can scale to billions of rows and thousands of columns, allowing you to store petabytes of data. Google Cloud fully manages the service, so you don't have to configure and tune it. It's ideal for data that has a single lookup key. Some application developers think of Bigtable as a persistent hashtable.

Bigtable is ideal for storing very large amounts of data with very low latency. It supports high throughput, both read and write, so it's a great choice for both operational and analytical applications, including Internet of Things, user analytics, and financial data analysis.

Bigtable is offered through the same open source API as HBase, which is the native database for the Apache Hadoop project. I’ll talk more about Hadoop later in the course. Anyway, having the same API enables portability of applications between HBase and Bigtable.



Why choose Cloud Bigtable?

- Replicated storage.
- Data encryption in-flight and at rest.
- Role-based ACLs.
- Drives major applications such as Google Analytics and Gmail.

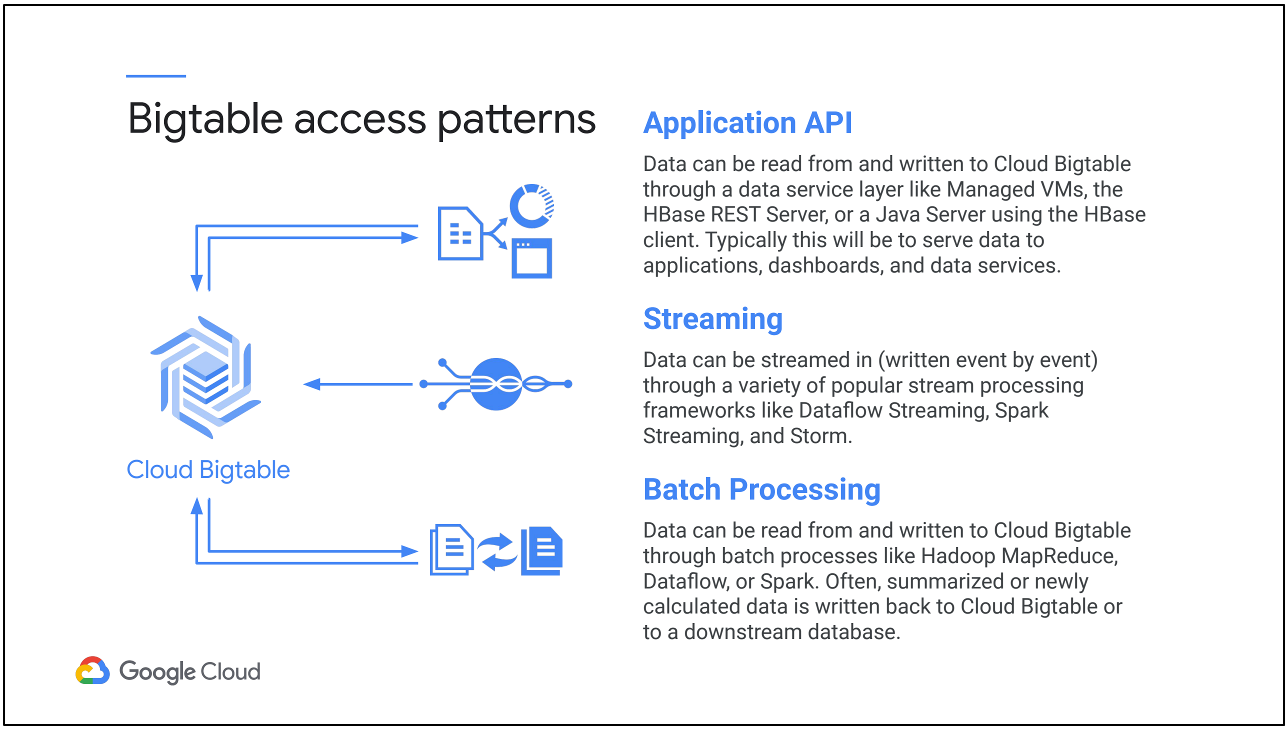
Google Cloud

Given that you could manage your own Apache HBase installation, you might ask yourself: Why should I choose Cloud Bigtable? Here are a few reasons why you might. First, scalability. If you manage your own HBase installation, scaling past a certain rate of queries per second is going to be tough. But with Bigtable, you can just increase your machine count, which doesn't even require downtime. Also, Bigtable handles administration tasks like upgrades and restarts transparently.

All data in Bigtable is encrypted both in-flight and at rest.

You can even use IAM permissions to control who has access to Bigtable data.

One last reference point: Bigtable is actually the same database that powers many of Google's core services, including Search, Analytics, Maps, and Gmail.



Bigtable access patterns

Application API

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 will be to serve data to applications, dashboards, and data services.

Streaming

Data can be streamed in (written event by event) through a variety of popular stream processing frameworks like Dataflow Streaming, Spark Streaming, and Storm.

Batch Processing

Data can 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.

As Cloud Bigtable is part of the Google Cloud ecosystem, it can interact with other Google Cloud services and third-party clients.

From an application API perspective, 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 will be 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.

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.

Agenda

Cloud Storage

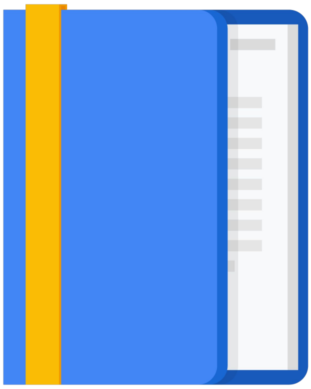
Cloud Bigtable

Cloud SQL, Cloud Spanner, and Firestore

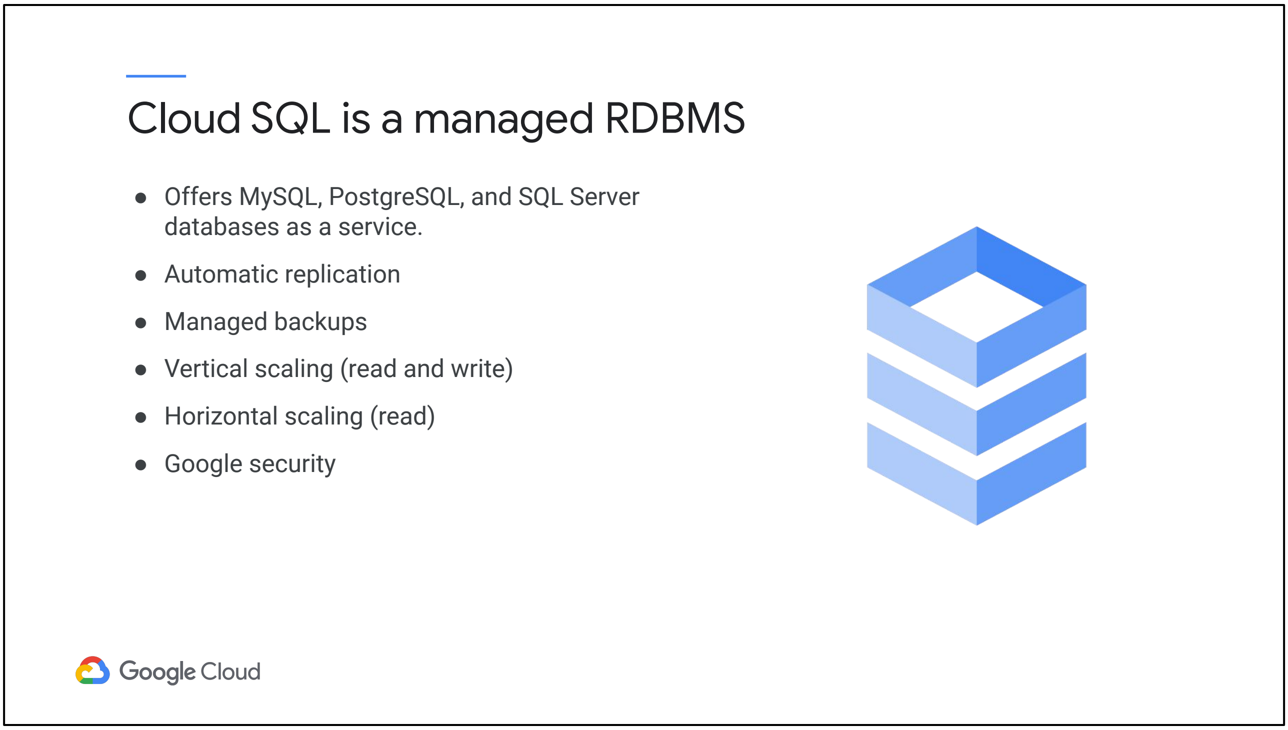
Comparing Storage Options

Quiz and Lab

Resources



Google Cloud



Cloud SQL is a managed RDBMS

- Offers MySQL, PostgreSQL, and SQL Server databases as a service.
- Automatic replication
- Managed backups
- Vertical scaling (read and write)
- Horizontal scaling (read)
- Google security

Google Cloud

A moment ago, we talked about NoSQL databases. Now, let's turn our attention to relational database services. Remember, these services use a database schema to help your application keep your data consistent and correct. Another feature of relational database services that helps with the same goal - transactions. Your application can designate a group of database changes as all or nothing. Either they all get made, or none do. Without database transactions, your online bank wouldn't be able to offer you the ability to move money from one account to another. What if, after subtracting $10,000 from one of your accounts, some glitch prevented it from adding that 10,000 to the destination account? Your bank would have just misplaced $10,000.

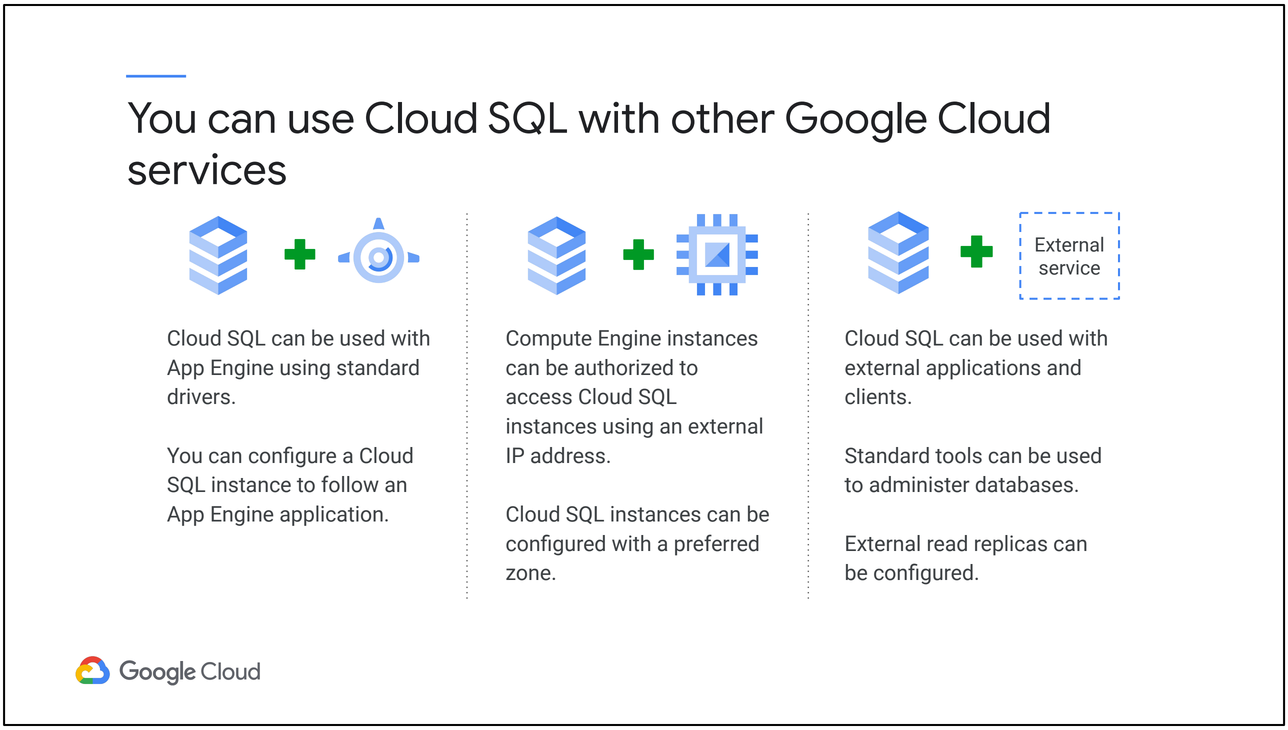
Classically, relational databases are a lot of work to set up, maintain, manage, and administer. If that doesn't sound like a good use of your time but you still want the protections of a relational database, consider Cloud SQL. It offers you your choice of the MySQL, PostgreSQL or SQL Server database engines as a fully managed service. Cloud SQL offers databases that are capable of handling terabytes of storage across all 3 options. Of course, you could always run your own database server inside a Compute Engine virtual machine, which a lot of Google Cloud customers do. But there are some benefits of using the Cloud SQL managed service instead.

Cloud SQL provides several replica services, like read, failover, and external replicas. This means that if an outage occurs, Cloud SQL can replicate data between multiple zones with automatic failover.

Cloud SQL also helps you backup your data with either on-demand or scheduled backups.

It can also scale both vertically, by changing the machine type and horizontally, via read replicas.

From a security perspective, Cloud SQL instances include network firewalls, and customer data is encrypted when on Google's internal networks and when stored in; database tables, temporary files, and backups.



You can use Cloud SQL with other Google Cloud services

Cloud SQL can be used with App Engine using standard drivers.

You can configure a Cloud SQL instance to follow an App Engine application.

Compute Engine instances can be authorized to access Cloud SQL instances using an external IP address.

Cloud SQL instances can be configured with a preferred zone.

External service

Cloud SQL can be used with external applications and clients.

Standard tools can be used to administer databases.

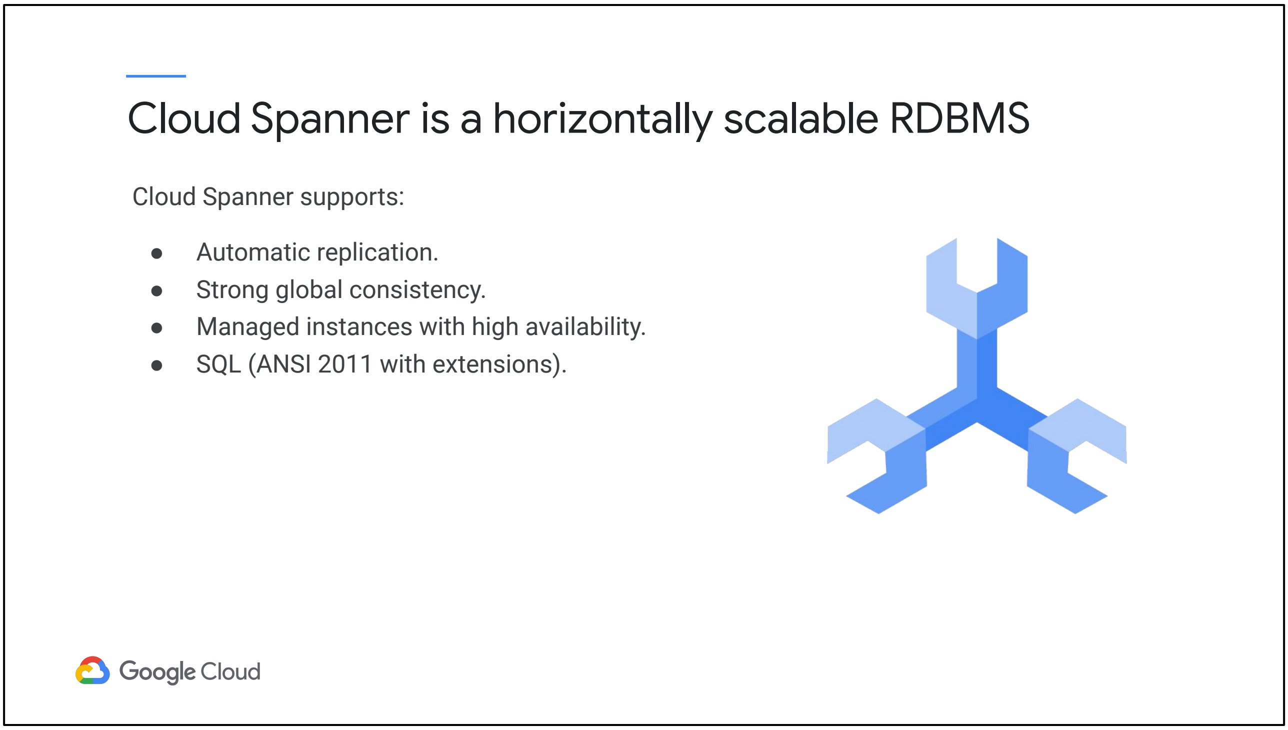
External read replicas can be configured.

Google Cloud

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

You can authorize Compute Engine instances 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 be used to, like SQL Workbench, Toad and other external applications using standard MySQL drivers.



Cloud Spanner is a horizontally scalable RDBMS

Cloud Spanner supports:

- Automatic replication.
- Strong global consistency.
- Managed instances with high availability.
- SQL (ANSI 2011 with extensions).

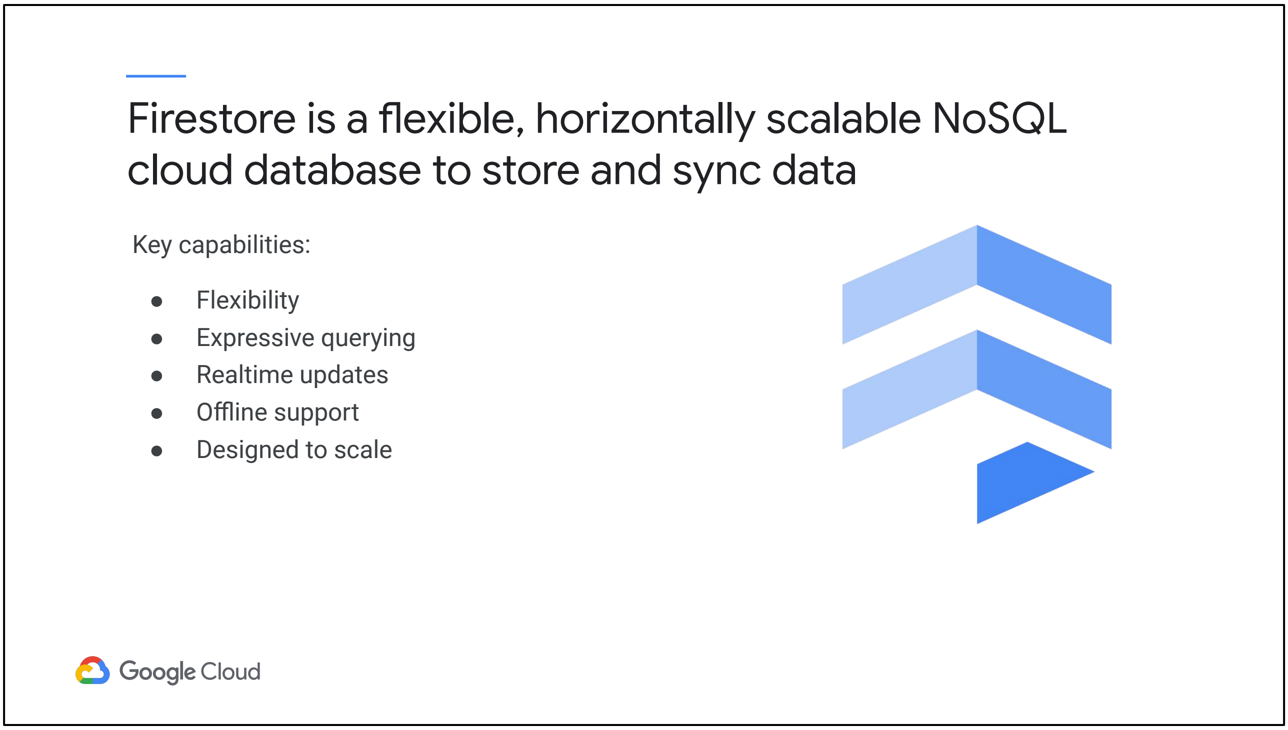
Google Cloud

Cloud Spanner supports strong consistency, including strongly consistent secondary indexes, SQL, and managed instances with high availability through synchronous and built-in data replication. Battle tested by Google's own mission-critical applications and services, Spanner powers Google's $80 billion business.

Cloud Spanner is especially suited for applications requiring:

- A SQL RDBMS, with joins and secondary indexes
- Built-in high availability
- Strong global consistency
- Database sizes exceeding ~2 TB
- Many IOPS (Tens of thousands of reads/writes per second or more)

For a technical overview of Cloud Spanner, see https://cloudplatform.googleblog.com/2017/02/inside-Cloud-Spanner-and-the-CAP-Theorem.html.



Firestore is a flexible, horizontally scalable NoSQL cloud database to store and sync data

Key capabilities:

- Flexibility
- Expressive querying
- Realtime updates
- Offline support
- Designed to scale

Google Cloud

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

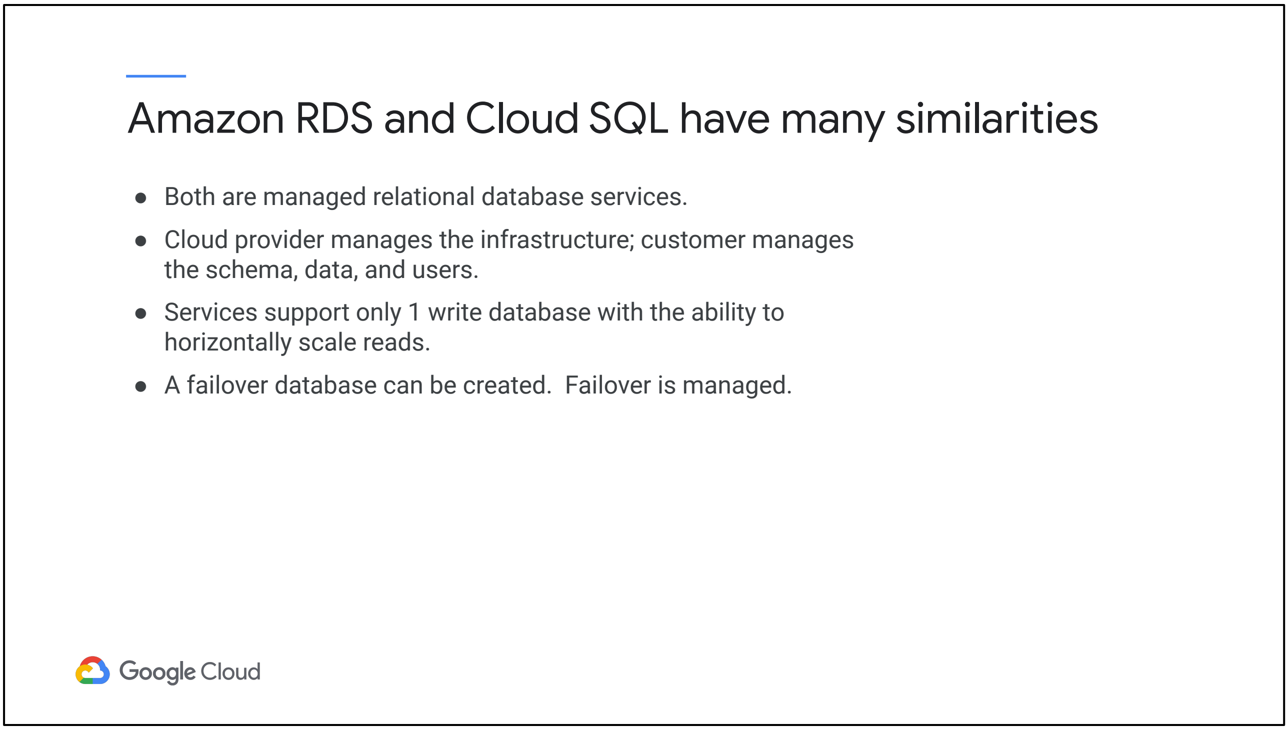
The Firestore data model supports flexible, hierarchical data structures. You store data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

You can use queries 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. 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.

Firestore 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.

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

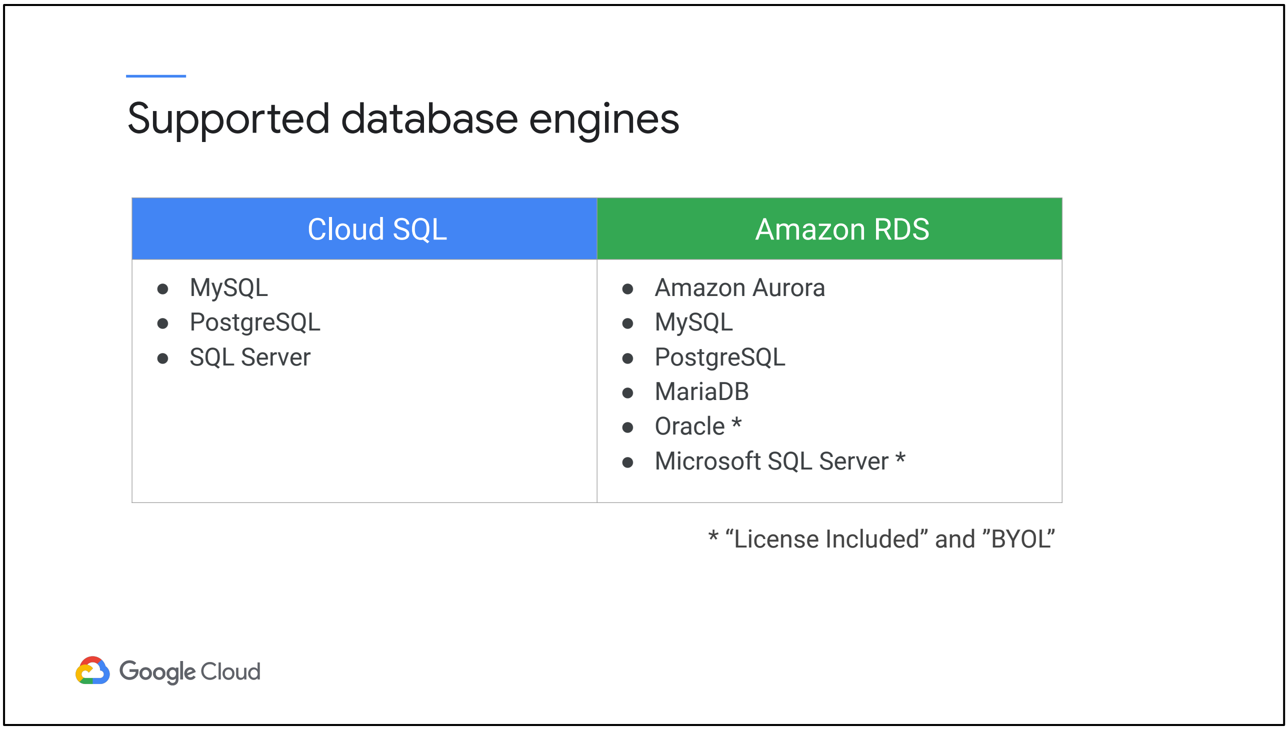


Amazon RDS and Cloud SQL have many similarities

- Both are managed relational database services.
- Cloud provider manages the infrastructure; customer manages the schema, data, and users.
- Services support only 1 write database with the ability to horizontally scale reads.
- A failover database can be created. Failover is managed.

Amazon Relational Data Service (Amazon RDS) and Cloud SQL are similar in many ways.

- Both are managed relational database services that use commercial relational database management systems
- AWS and Google Cloud will manage the infrastructure of the databases to include the instances, backups, updates, and failover. The user is still responsible for data management, schema construction, and user management.
- Reads and writes can be scaled differently. For the write database, the instance can only be vertically scaled. You can make a single write database larger. You can horizontally scale reads with read replicas and put them in different zones or availability zones.
- A failover replica can be built in another zone or availability zone. The cloud provider is responsible for the failover process.



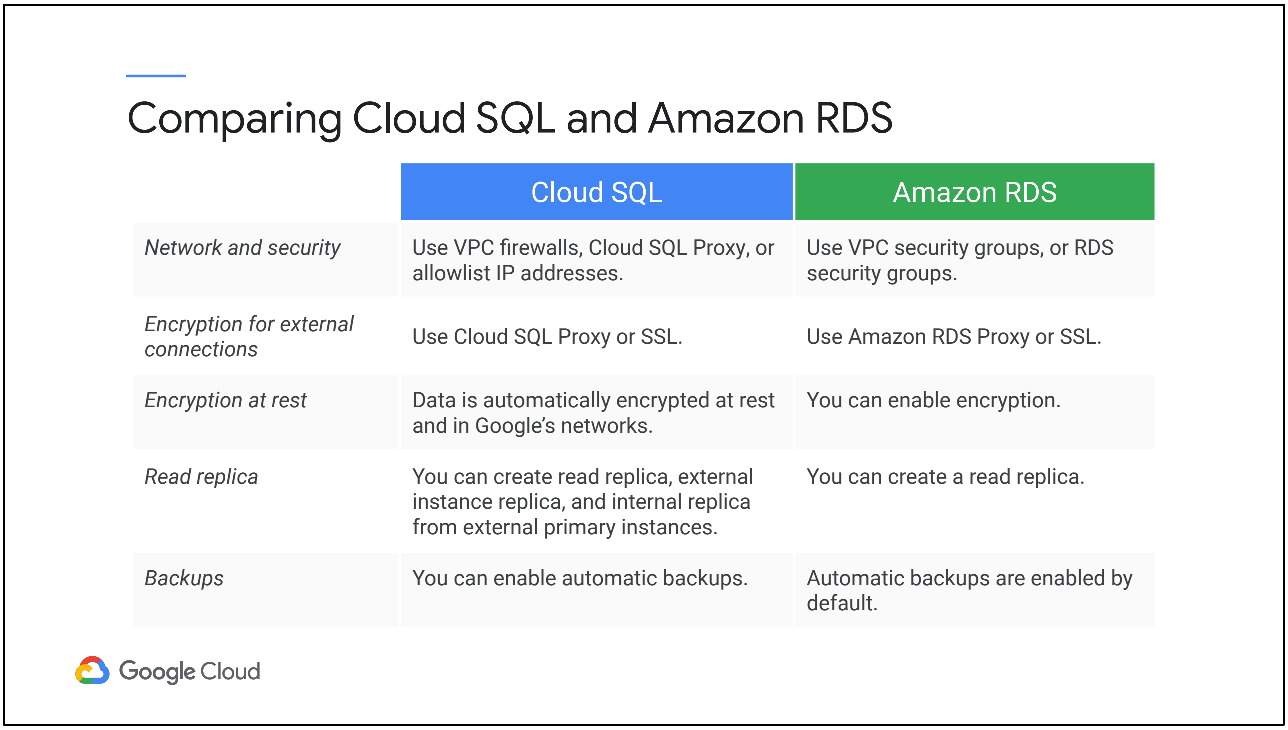
Supported database engines

Cloud SQL	Amazon RDS
• MySQL • PostgreSQL • SQL Server	• Amazon Aurora • MySQL • PostgreSQL • MariaDB • Oracle * • Microsoft SQL Server *

* “License Included” and ”BYOL”

Cloud SQL is a fully managed relational database service for MySQL, PostgreSQL, and SQL Server.

Amazon RDS supports Amazon Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server.



Comparing Cloud SQL and Amazon RDS

	Cloud SQL	Amazon RDS
Network and security	Use VPC firewalls, Cloud SQL Proxy, or allowlist IP addresses.	Use VPC security groups, or RDS security groups.
Encryption for external connections	Use Cloud SQL Proxy or SSL.	Use Amazon RDS Proxy or SSL.
Encryption at rest	Data is automatically encrypted at rest and in Google's networks.	You can enable encryption.
Read replica	You can create read replica, external instance replica, and internal replica from external primary instances.	You can create a read replica.
Backups	You can enable automatic backups.	Automatic backups are enabled by default.

Google Cloud

Let's compare Cloud SQL and Amazon RDS.

- In Amazon RDS, you use VPC security groups to authorize connections or use RDS security groups to give DB users and groups access to tables, views, and queries. In Cloud SQL, use VPC firewalls or Cloud SQL Proxy or authorize networks by adding IP addresses to an allowlist.
- Both Amazon RDS and Cloud SQL support the use of SSL for encrypted external connections. You can also use Amazon RDS Proxy and Cloud SQL Proxy to encrypt connections to their respective databases.
- In Amazon RDS, you can enable encryption of data at rest. Cloud SQL customer data is encrypted by default when on Google's internal networks and when stored in database tables, temporary files, and backups.
- You can create a read replica in Amazon RDS. In Cloud SQL for MySQL, you can create a read replica of a Cloud SQL instance. You can also create an external instance that replicates from a Cloud SQL primary and create a replica in Cloud SQL from external primary instances.
- In Amazon RDS, backups are enabled by default. In Cloud SQL, you can optionally enable automatic backups.

Agenda

Cloud Storage

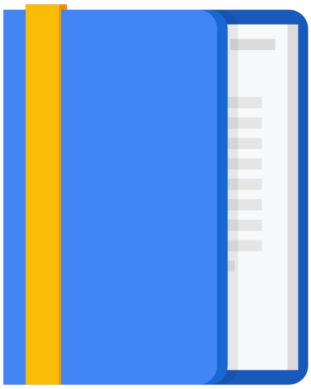
Cloud Bigtable

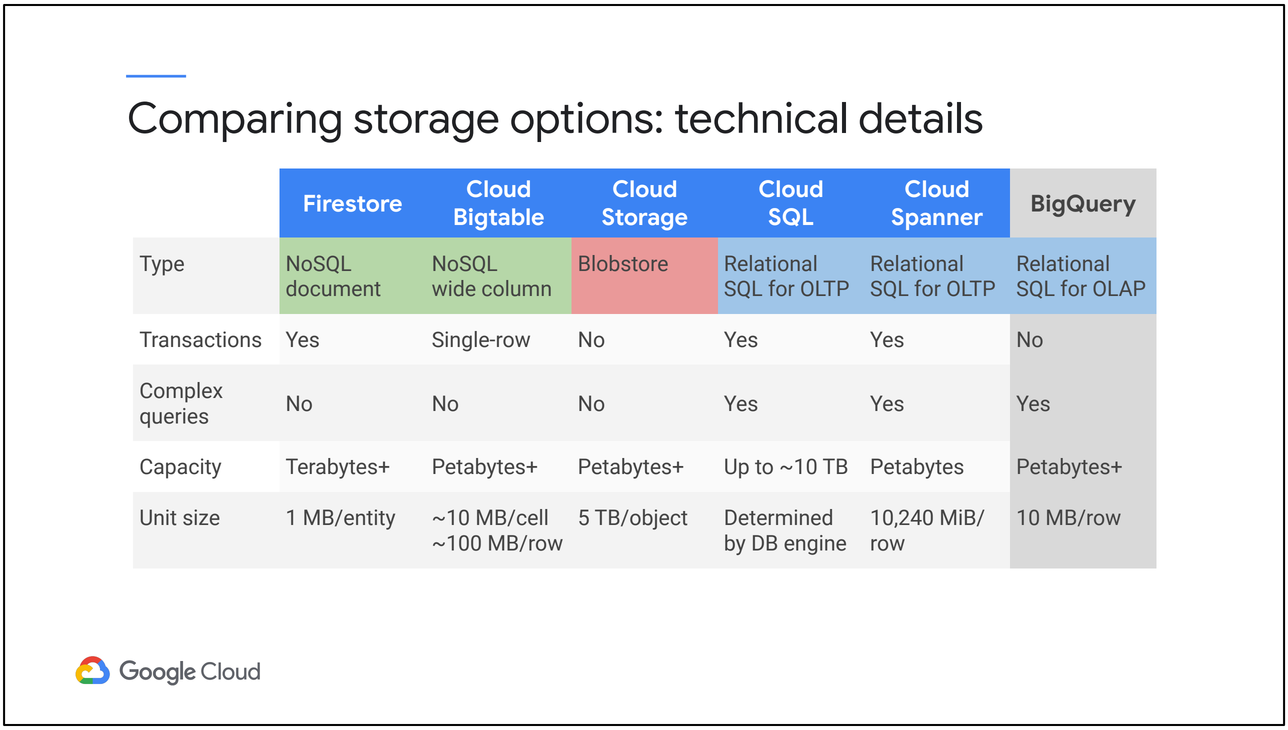
Cloud SQL, Cloud Spanner, and Firestore

Comparing Storage Options

Quiz and Lab

Resources





Comparing storage options: technical details

	Firestore	Cloud Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Transactions	Yes	Single-row	No	Yes	Yes	No
Complex queries	No	No	No	Yes	Yes	Yes
Capacity	Terabytes+	Petabytes+	Petabytes+	Up to ~10 TB	Petabytes	Petabytes+
Unit size	1 MB/entity	~10 MB/cell ~100 MB/row	5 TB/object	Determined by DB engine	10,240 MiB/ row	10 MB/row

Google Cloud

Now that we covered Google Cloud’s core storage options, let’s compare them to help you choose the right service for your application or workflow.

This table focuses on the technical differentiators of the storage services. Each row is a technical specification and each column is a service. Let me cover each service from left to right.

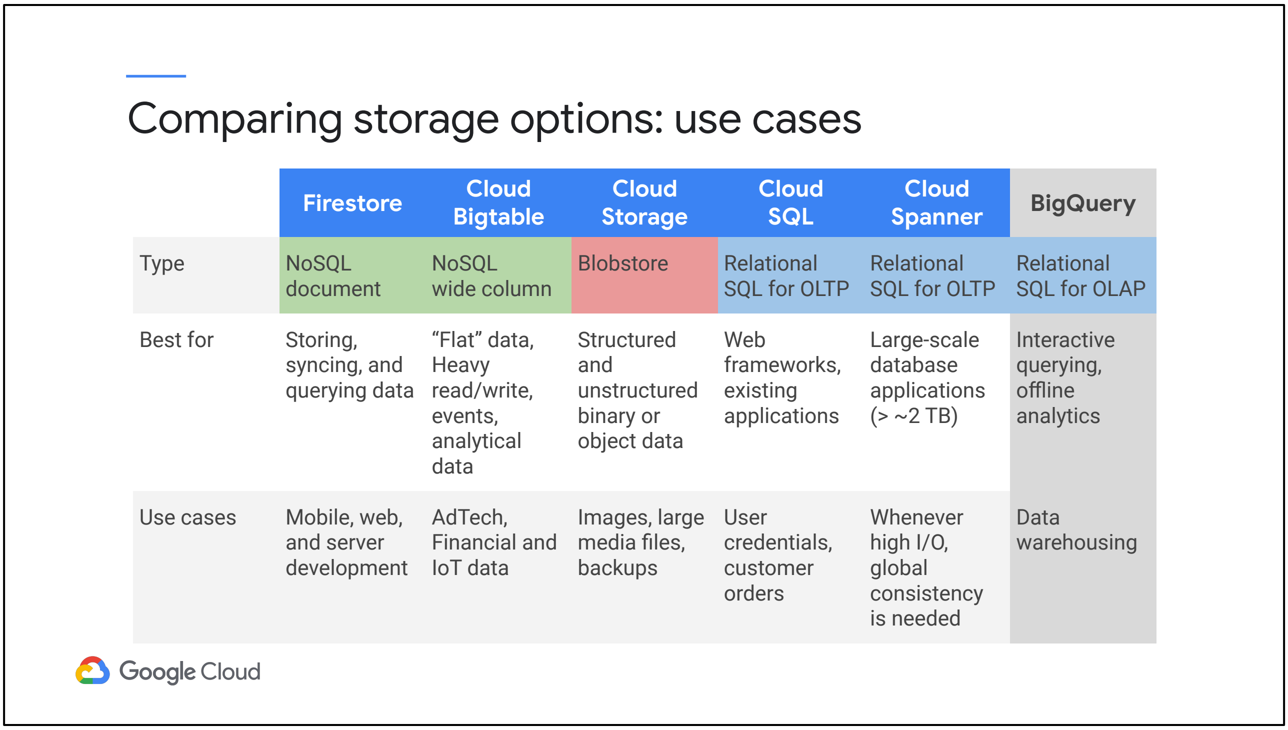
Consider Firestore if you need massive scaling and predictability together with real time query results and offline query support. This storage services provides terabytes of capacity with a maximum unit size of 1 MB per entity.

Consider using Cloud Bigtable if you need to store a large amount of structured objects. Cloud Bigtable does not support SQL queries, nor does it support multi-row transactions. This storage service provides petabytes of capacity with a maximum unit size of 10 MB per cell and 100 MB per row.

Consider using Cloud Storage if you need to store immutable blobs larger than 10 MB, such as large images or movies. This storage service provides petabytes of capacity with a maximum unit size of 5 TB 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 Up to 10,230 GB, depending on machine type, while Cloud Spanner provides petabytes. If Cloud SQL does not fit your requirements because you need horizontal scalability, not just through read replicas, consider using Cloud Spanner.

We didn’t cover BigQuery in this module as it sits on the edge between data storage and data processing, but you will learn more about it in the “Big Data and Machine Learning in the Cloud” module. The usual reason to store data in BigQuery is to use its big data analysis and interactive querying capabilities. You would not want to use BigQuery, for example, as the backing store for an online application.



Comparing storage options: use cases

	Firestore	Cloud Bigtable	Cloud Storage	Cloud SQL	Cloud Spanner	BigQuery
Type	NoSQL document	NoSQL wide column	Blobstore	Relational SQL for OLTP	Relational SQL for OLTP	Relational SQL for OLAP
Best for	Storing, syncing, and querying data	"Flat" data, Heavy read/write, events, analytical data	Structured and unstructured binary or object data	Web frameworks, existing applications	Large-scale database applications (> ~2 TB)	Interactive querying, offline analytics
Use cases	Mobile, web, and server development	AdTech, Financial and IoT data	Images, large media files, backups	User credentials, customer orders	Whenever high I/O, global consistency is needed	Data warehousing

Google Cloud

Considering the technical differentiators of the different storage services helps some people decide which storage service to choose, others like to consider use cases. Let me go through each service one more time.

- Firestore is best for storing, syncing, and querying data for mobile and web apps.
- Bigtable is best for analytical data with heavy read and write events, like AdTech, financial or IoT data.
- Cloud Storage is best for structured and unstructured binary or object data, like images, large media files and backups.
- Cloud SQL is best for web frameworks and existing applications, like storing user credentials and customer orders.
- Cloud Spanner is best for large-scale database applications that are larger than 2 TB. For example, for financial trading and e-commerce use cases.

As I mentioned at the beginning of the module, depending on your application you might use one or several of these services to get the job done.

Agenda

Cloud Storage

Cloud Bigtable

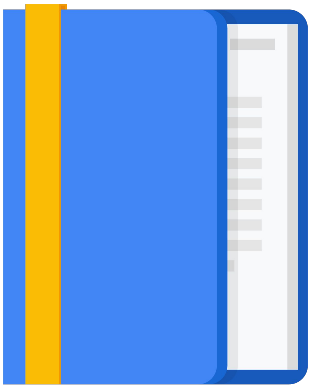
Cloud SQL and Cloud Spanner

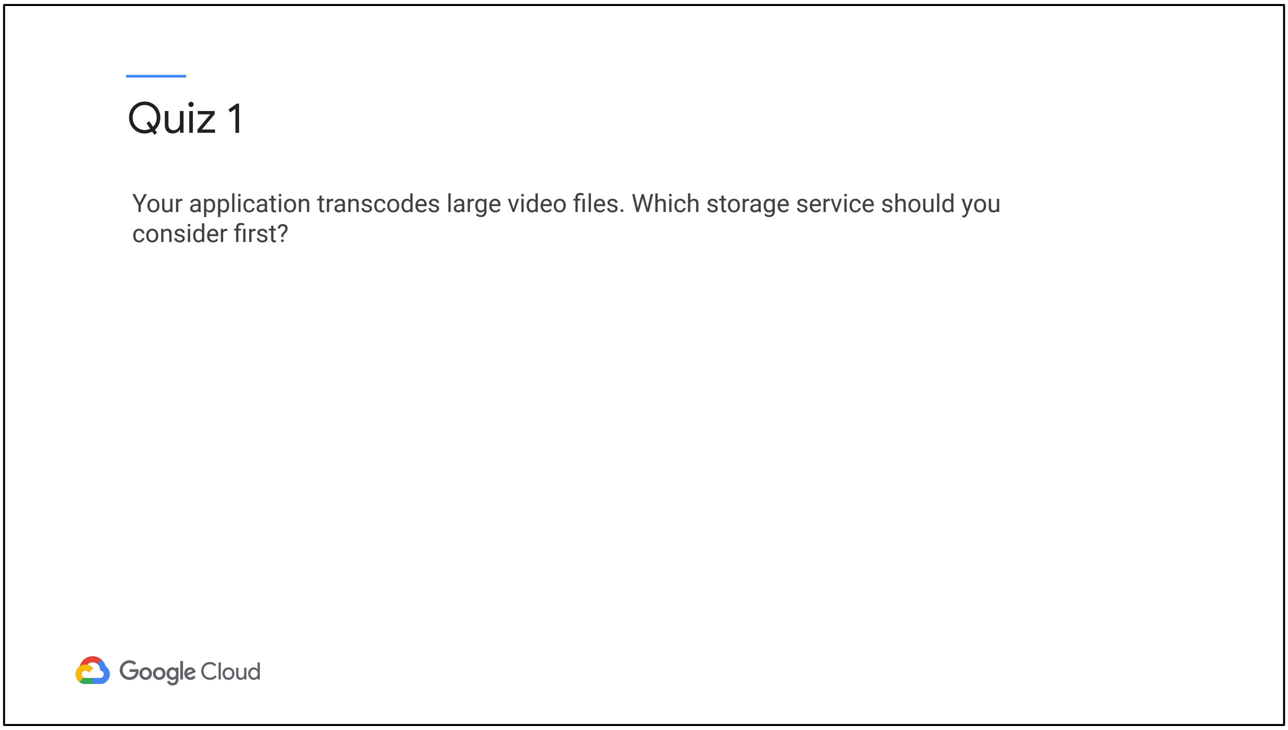
Firestore

Comparing Storage Options

Quiz and Lab

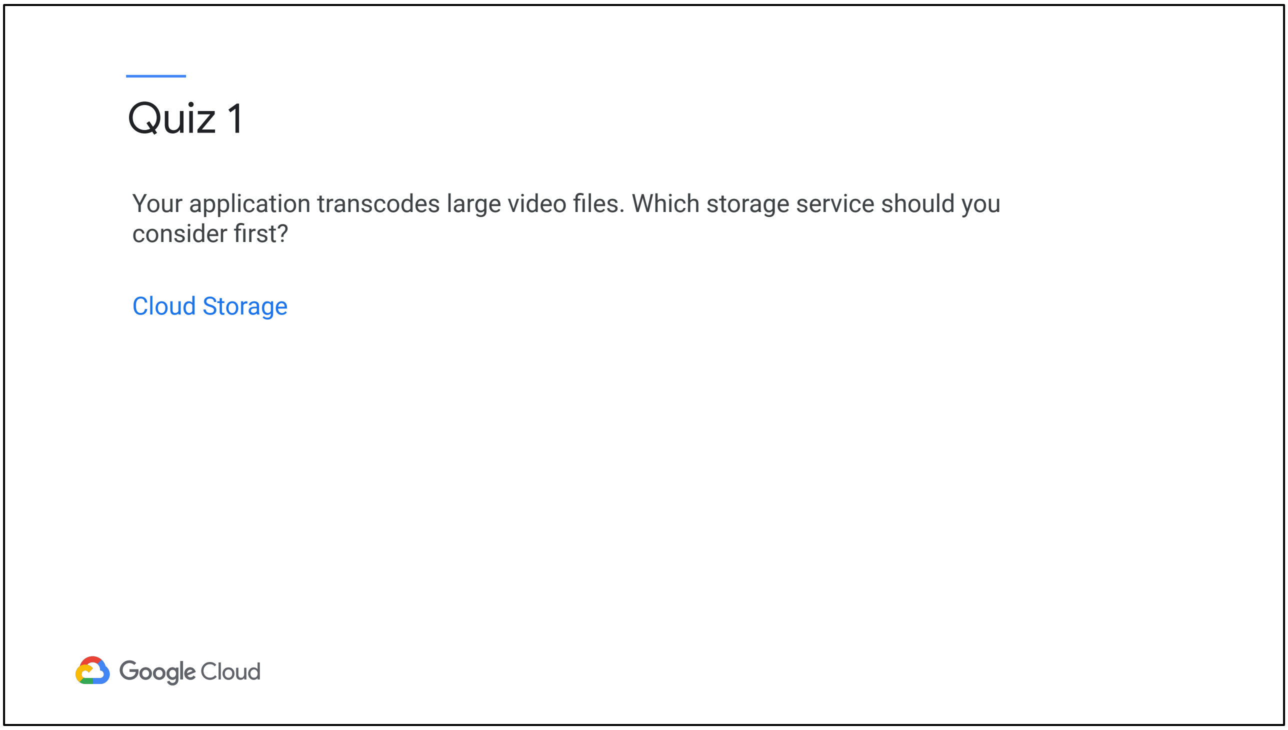
Resources





Quiz 1

Your application transcodes large video files. Which storage service should you consider first?

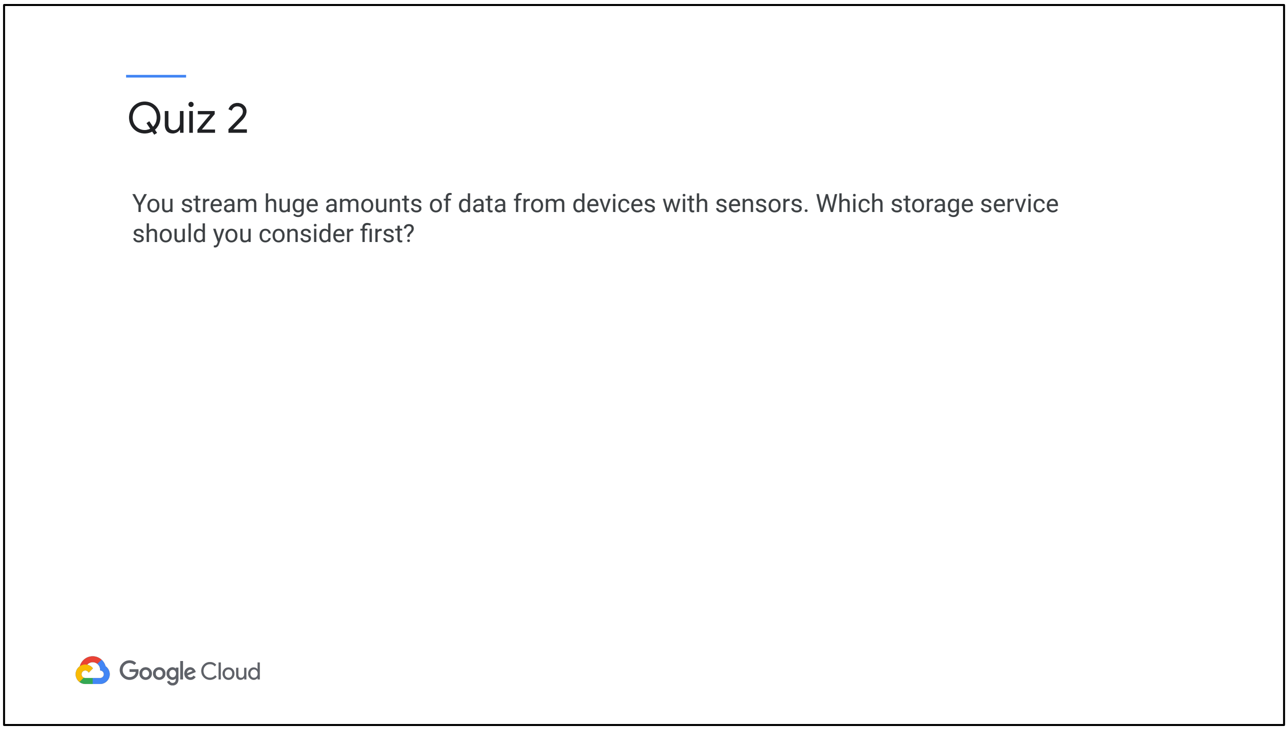


Quiz 1

Your application transcodes large video files. Which storage service should you consider first?

Cloud Storage

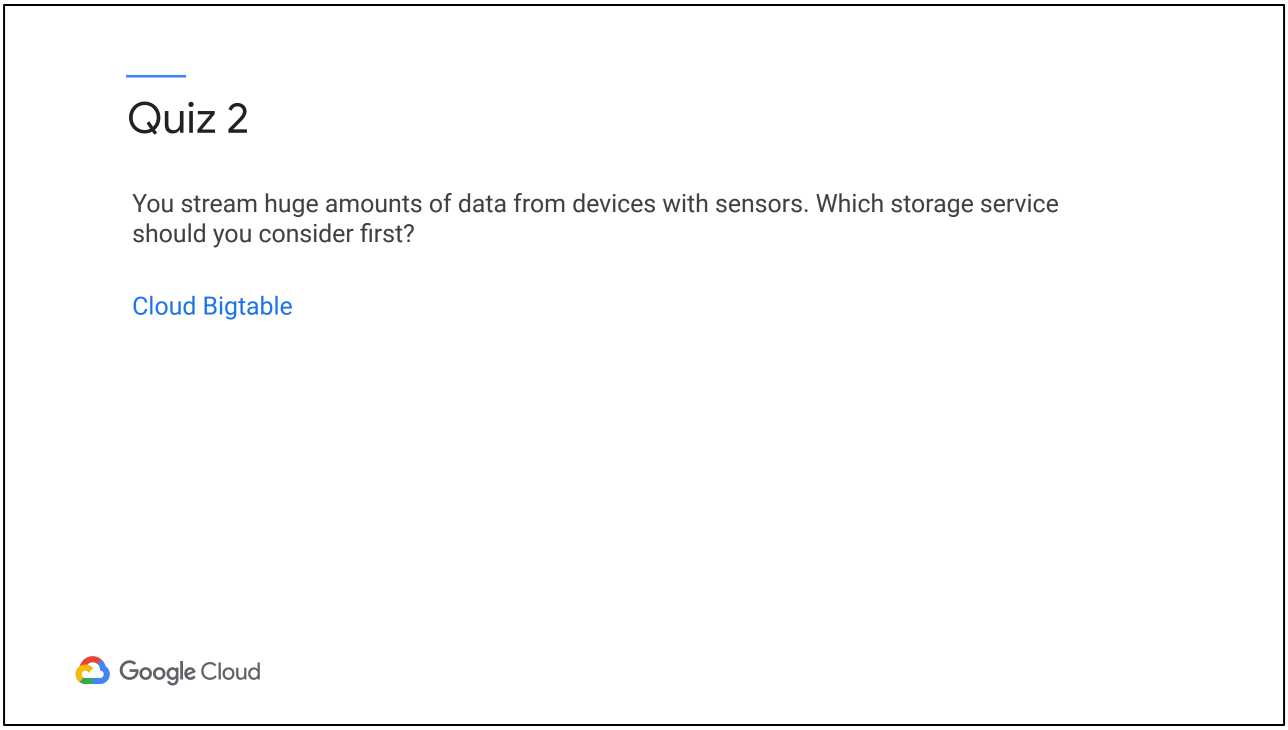
Google Cloud



Quiz 2

You stream huge amounts of data from devices with sensors. Which storage service should you consider first?

Google Cloud

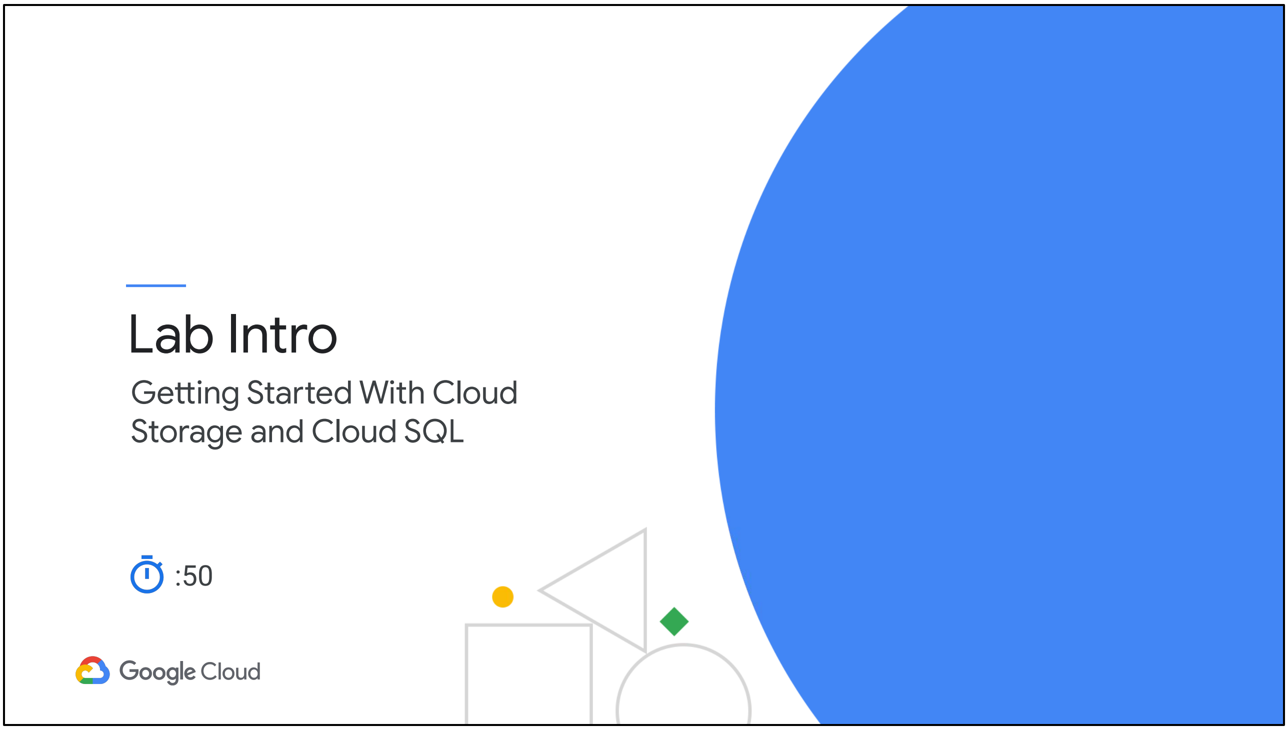


Quiz 2

You stream huge amounts of data from devices with sensors. Which storage service should you consider first?

Cloud Bigtable

Google Cloud



Lab Intro

Getting Started With Cloud Storage and Cloud SQL

:50

In this lab you will store an image in a Cloud Storage bucket and configure Compute Engine to use a Cloud SQL database to reference the image. The objectives of this lab are to:

- Create a Cloud Storage bucket and place an image into it.
- Create a Cloud SQL instance and configure it.
- Connect to a Cloud SQL instance from a web server.
- Use an image stored in a Cloud Storage bucket in a web page.

Agenda

Cloud Storage

Cloud Bigtable

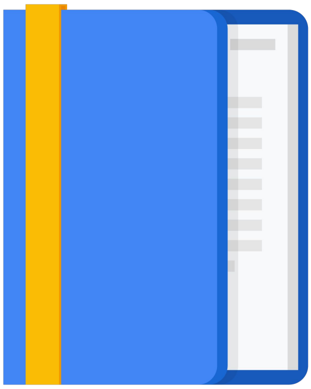
Cloud SQL and Cloud Spanner

Firestore

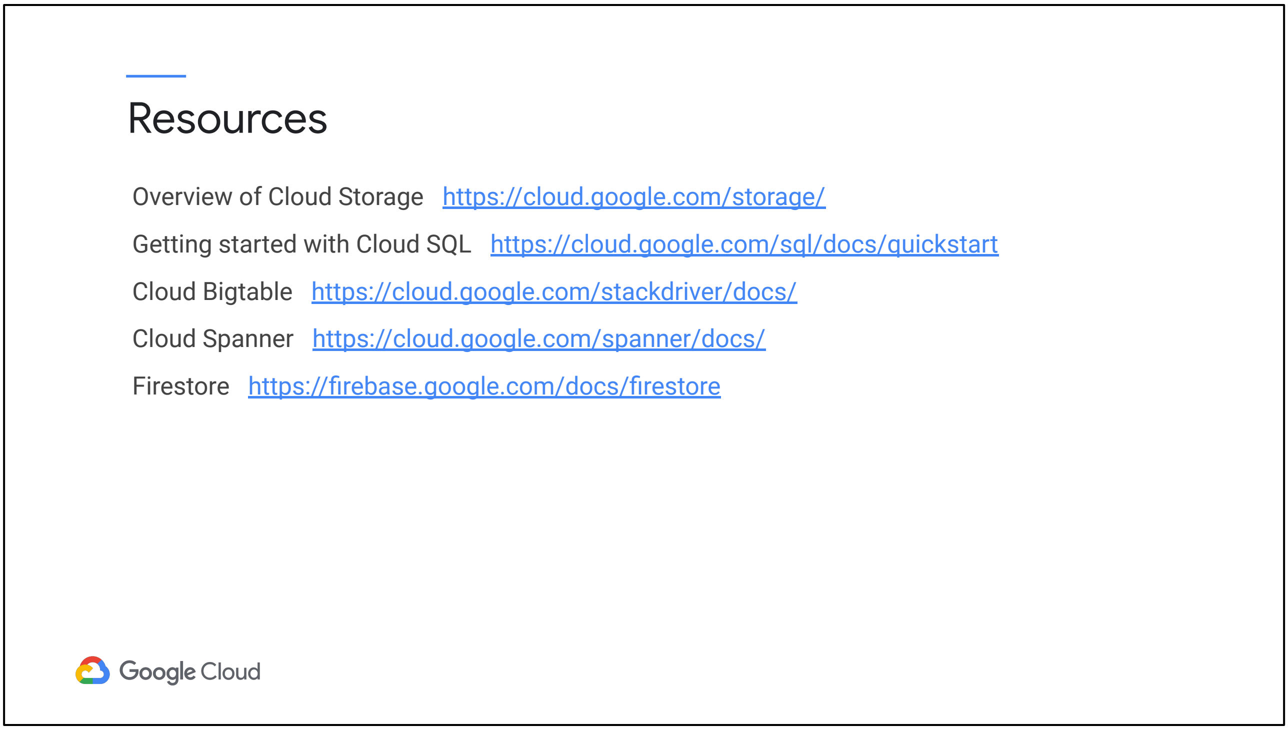
Comparing Storage Options

Quiz and Lab

Resources



Google Cloud



Resources

Overview of Cloud Storage https://cloud.google.com/storage/

Getting started with Cloud SQL https://cloud.google.com/sql/docs/quickstart

Cloud Bigtable https://cloud.google.com/stackdriver/docs/

Cloud Spanner https://cloud.google.com/spanner/docs/

Firestore https://firebase.google.com/docs/firestore

Google Cloud