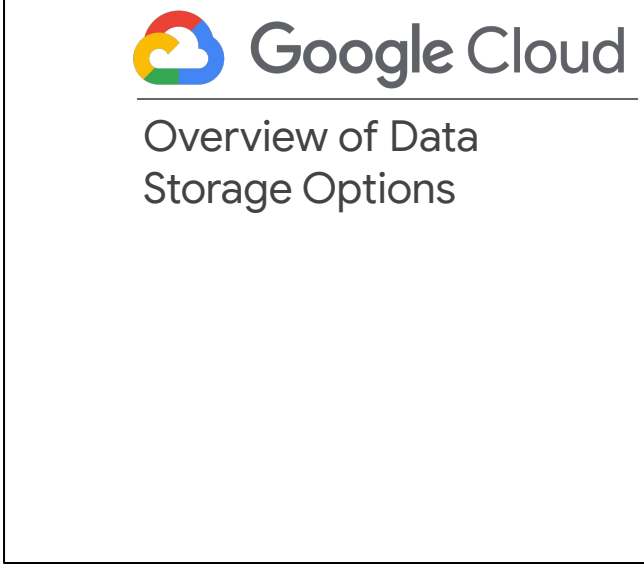


Applications can have various types of data. For example, if your application is a social image sharing site, you have to store image files, high volumes of user messages, and transactional data. You'll likely need to cache frequently accessed data as well. And of course, you'll want to collect, query, and analyze all the data to gather business intelligence about your users, as well as product usage patterns.

Google Cloud offers a number of managed services that you can use for each type of data. In this module, Overview of Data Storage Options, you'll learn about services such as Cloud Storage, Datastore, Cloud Bigtable, Cloud SQL, Cloud Spanner, and BigQuery. You'll learn the ideal use cases for each data storage option, as well as use cases for which the option might not be suitable. Armed with this knowledge, you can choose a data storage option that meets the specific use case in your application.

A common problem that application developers face is handling database connections in a secure manner. You'll learn how to connect your Cloud SQL second generation instance without having to allowlist IP addresses or configure SSL.

Google Cloud provides a full suite of storage service options

- Cost-effective
- Varied choices based on your:
  - Application
  - Workload

Google Cloud

You have a full suite of cost-effective storage services to choose from when developing with Google Cloud. No one size fits all, and your choice of storage and database solutions will depend on your application and workload.

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

Cloud Storage

Overview	Ideal for
• Fully managed, highly reliable • Cost-efficient, scalable object/blob store • Objects access via HTTP requests • Object name is the only key	• Images and videos • Objects and blobs • Unstructured data • Static website hosting

Google Cloud

Cloud Storage is a unified object storage for developers and enterprises that allows you to serve, analyze, and archive data. Objects are accessed via HTTP requests, including ranged GETS to retrieve portions of the data. The only key is the object name. There is object metadata but the object itself is treated as just bytes with no structure. The scale of the system allows for serving static content or accepting user uploaded content like photos and videos.

Cloud Storage is built for availability, durability, scalability, and consistency. It is an ideal solution for storing images and videos, objects and blobs, and any unstructured data.

For more information, see the Cloud Storage Documentation:
https://cloud.google.com/storage/docs/

Firestore in native mode

Overview	Ideal for
• Fully managed, serverless, NoSQL • Scalable • Native mobile and web client libraries • Real-time updates and offline features	• Native mobile and web clients • Document-oriented data • Large collections of small documents • Durable key-value data • Hierarchical data • Managing multiple indexes • Transactions

Google Cloud

Firestore is a fast, fully managed, serverless, NoSQL document database built for automatic scaling, high performance, and ease of application development.

Firestore in native mode provides features such as:

- A strongly consistent storage layer
- A collection and document-based data model
- Real-time updates and offline features
- Mobile and Web client libraries

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

For more information, see the Firestore Documentation:
https://cloud.google.com/firestore/docs/

Datastore (Firestore in Datastore mode)

Overview	Ideal for
• Fully managed NoSQL	• Server applications
• Scalable	• Semi-structured application data
• No mobile and web client libraries	• Durable key-value data
• No real-time and offline features	• Hierarchical data
	• Managing multiple indexes
	• Transactions

Google Cloud

Firestore in Datastore mode, which is often referred to as Datastore, is a highly-scalable NoSQL database for your applications. Datastore automatically handles sharding and replication, providing you with a highly available and durable database that scales automatically to handle your applications' load. Datastore provides a myriad of capabilities such as ACID transactions, SQL-like queries, indexes and much more. Datastore scales seamlessly and automatically with your data allowing applications to maintain high performance as they receive more traffic.

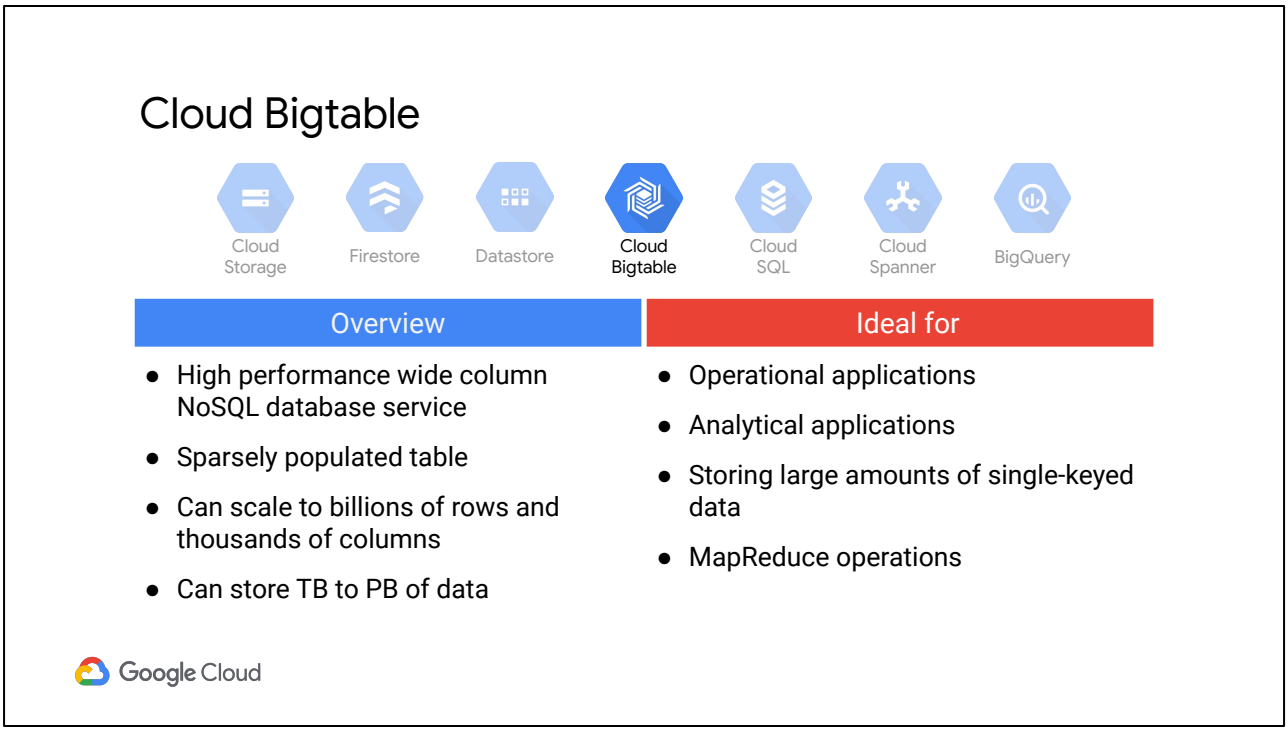
Datastore is an ideal solution when your application data is semi-structured or hierarchical and when you need to store durable key-value data at scale. Typical use cases for Datastore include product catalogs, user profiles, and transactions based on ACID properties.

Datastore is not ideal for every use case. For example, Datastore is not a relational database, and it is not an effective solution for analytic data.

Datastore was formerly implemented as an extension of Cloud Bigtable but now is implemented using Firestore. Firestore in Datastore mode is backward compatible with Datastore, but the native mode's data model, real-time updates, and mobile and web client library features are not available when using Datastore mode. Firestore in Datastore mode provides improvements to the original Datastore, including strongly consistent queries, transaction improvements, and the removal of the one per second limit on writes to an entity group.

Eventually, existing Datastore instances will be automatically migrated to Firestore in Datastore Mode. No functional changes are required to existing Datastore code in applications.

For more information, see the Datastore Documentation:
https://cloud.google.com/datastore/docs/
Automatic migration from Datastore to Firestore in Datastore mode:
https://cloud.google.com/datastore/docs/upgrade-to-firestore

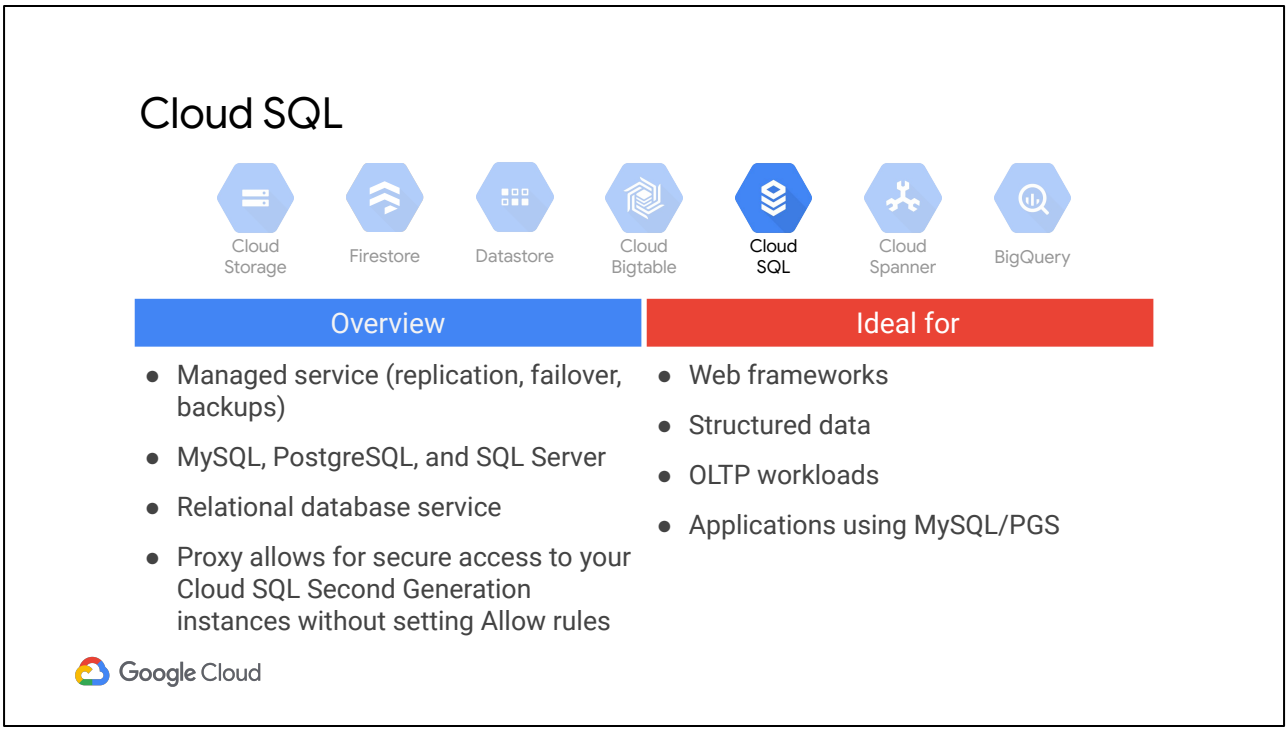


Bigtable is a high performance NoSQL database service. It is a sparsely populated table that can scale to billions of rows and thousands of columns. Bigtable can store terabytes to petabytes of data. Bigtable is built for fast key-value lookup and scanning over a defined key range. It is similar to a spreadsheet that gives you access to any set of columns from contiguous rows by searching only the value in the first column (the key). Updates to individual rows are atomic. Due to the fast lookup and write speed of Bigtable, it's great for user behavior.

Bigtable supports operational and analytical applications and is ideal for storing large amounts of single-keyed data and MapReduce operations. Bigtable offers consistent sub-10ms latency. Bigtable offers seamless scaling: Changes to the deployment configuration are immediate, so there’s no downtime during reconfiguration.

Cloud Bigtable supports the open source industry standard HBase API.

For more information, see the Bigtable Documentation:
https://cloud.google.com/bigtable/docs/

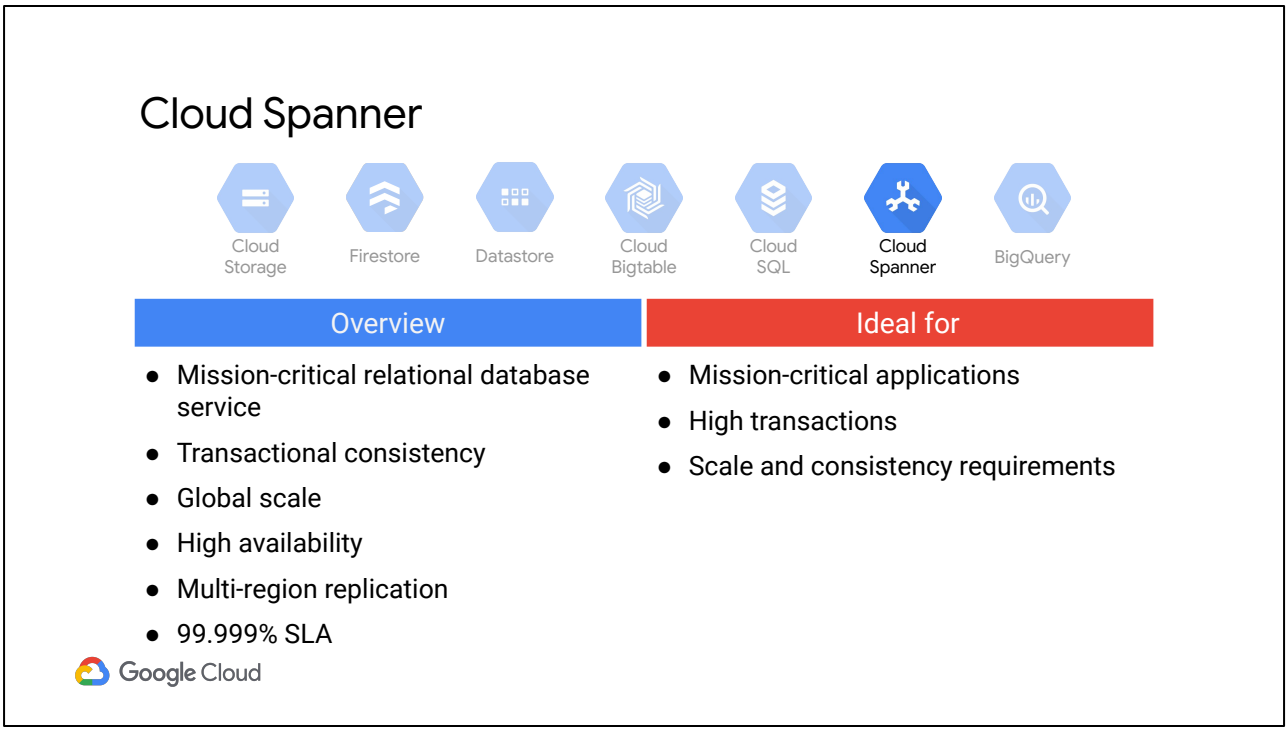


Cloud SQL is Google Cloud's relational database service. It is a managed service that lets Google manage replication, failover, and backups of your databases so you can focus on your MySQL, PostgreSQL, or SQL Server-compatible applications. Cloud SQL lets you easily configure replication and backups to protect your data. You can replicate a primary instance to one or more read replicas. A read replica is a copy of the primary that reflects changes to the primary instance in almost-real time. You can enable automatic failover to make your database highly available. Backups allow you to restore your Cloud SQL instance to recover lost data or recover from a problem with your instance. You can enable automated backups for any instance that contains data that you need to protect from loss or damage.

The Cloud SQL Proxy works by having a local client, called the proxy, running in the local environment. Your application communicates with the proxy with the standard database protocol used by your database. The proxy uses a secure tunnel to communicate with its companion process running on the server. Cloud SQL Proxy provides secure access to your Cloud SQL Second Generation instances without your having to allow IP addresses or configure SSL. The proxy uses the Cloud SQL API to authenticate with Google Cloud. You must enable the API before using the proxy, and you must provide the proxy with a valid user account.

Cloud SQL is ideal for web frameworks, applications requiring structured data, and online transaction processing (OLTP) workloads. It is ideal for applications using MySQL/PostgreSQL/SQL Server, with minimal refactoring required for migration to Google Cloud.

For more information, see:
Google Cloud SQL Documentation: https://cloud.google.com/sql/docs/
Replication Options: https://cloud.google.com/sql/docs/mysql/replication/
Google Cloud SQL Backups:
https://cloud.google.com/sql/docs/mysql/backup-recovery/backups
Cloud SQL Proxy: https://cloud.google.com/sql/docs/mysql/sql-proxy

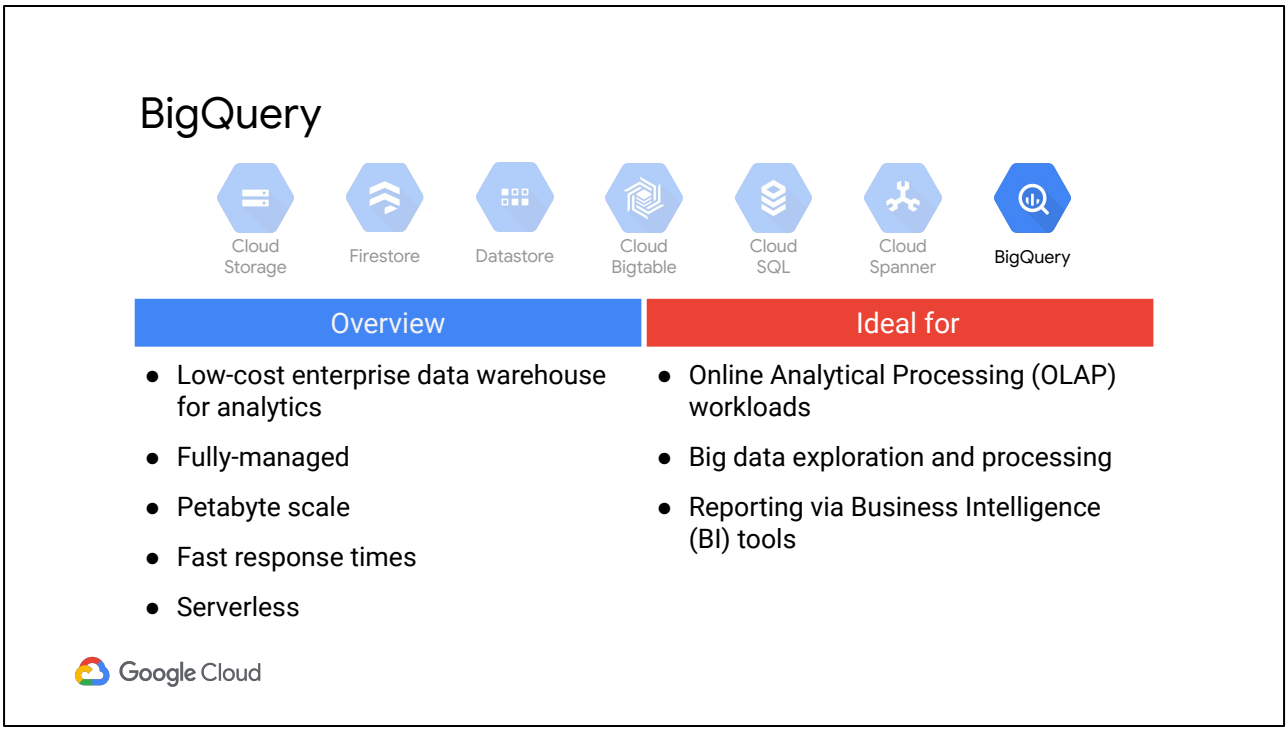


Google Cloud Spanner is Google Cloud's fully managed relational database service offering both strong consistency and horizontal scalability. It is designed for mission-critical OLTP applications. Cloud Spanner provides automatic, synchronous replication for high availability. Spanner is built for multi-region replication and offers one of the highest SLAs in the industry: 99.999%.

Spanner is ideal for applications with relational, structured, and semi-structured data that require high availability, strong consistency, and transactional reads and writes.

Unlike Cloud SQL, Spanner requires every table to have a primary key. Another difference is that Spanner also supports interleaved tables, where child rows are inserted into the table adjacent to the parent row. This improves the query performance when joins are done between parent and child.

For more information, see the Cloud Spanner Documentation:
https://cloud.google.com/spanner/docs/
For information on schema design best practices for Spanner, see
https://cloud.google.com/spanner/docs/schema-design



BigQuery is a low-cost enterprise data warehouse for analytics. It is a fully managed service, meaning you don't need to worry about administration of your data warehouse. BigQuery can scan TB in seconds and PB in minutes.

It is a great solution for Online Analytical Processing (OLAP) workloads, for big data exploration and processing, and to report via Business Intelligence (BI) tools.

For more information, see the BigQuery Documentation:
https://cloud.google.com/bigquery/docs/

Run Microsoft SQL Server on Google Cloud

- You can SQL Server images on Google Compute Engine
- Compute Engine VMs can be preloaded with SQL Server
- Licensing from Microsoft is included automatically
- Supported versions include:
  - SQL Server Standard
  - SQL Server Web
  - SQL Server Enterprise

You can run your Microsoft SQL Server deployment on Google Cloud. Compute Engine VMs come preloaded with SQL Server and licensing from Microsoft is included automatically. Supported versions include SQL Server Standard, SQL Server Web, and SQL Server Enterprise. Microsoft SQL Server on Google Cloud is not a managed service like Cloud SQL or Spanner. For more information on running Microsoft SQL Server on Google Cloud, please see the references available in the downloads pane.

Storage options for mobile

	Cloud Storage for Firebase	Firebase Realtime Database	Firebase Hosting
Overview	• Mobile and web access to Google Cloud Storage • Serverless third-party authentication and authorization	• Realtime • NoSQL JSON database	• Web and mobile content hosting • Production-grade
Ideal for	• Images, pictures, and videos • Objects and blobs • Unstructured data	• Mobile and web applications • Realtime	• Atomic release management • JS app support • Firebase integration

Google Cloud

Firebase is a mobile and web application development platform with a few storage options for web and mobile development with Google Cloud.

Cloud Storage for Firebase stores user-generated data and files to Google Cloud Storage. The Firebase SDKs for Cloud Storage provide simple and intuitive authentication for developers. Ideal use cases include saving user-generated images, pictures, videos, objects, and blogs from your mobile or web applications.

Firebase Realtime Database allows you to store and sync data with Firebase's NoSQL cloud database. Data is synced across all clients in real time so that it remains available when your app goes offline. It is an ideal storage solution for mobile and web applications that require responsiveness when offline. Firebase Realtime Database is Firebase's original database, Firestore is now recommended instead of Firebase Realtime Database for most developers starting a new project.

Firebase Hosting is a fast and secure way to host static resources for your web app. It is ideal for URL rewriting, atomic release management, and firebase integration of your web or mobile app.

Cloud Storage for Firebase Documentation: https://firebase.google.com/docs/storage/
Firebase Realtime Database Documentation: https://firebase.google.com/docs/database/
Firebase Hosting Documentation: https://firebase.google.com/docs/hosting/

Cache your application data

- Memorystore automates complex tasks for Redis and Memcached caching engines
- Fully protocol compatible with each engine
- Ideal for high-performance, scalable web applications, gaming, and stream processing
- Fully managed service
- Google-grade security

Memorystore

Google Cloud

Applications that you run on Google Cloud can achieve high levels of performance by leveraging either Redis or Memcached without the burden of managing complex deployments. Memorystore supports both of these highly scalable, available, and secure open source caching engines and is fully protocol compatible.

Memorystore is ideal for scalable web applications, gaming, and stream processing, where a distributed in-memory data store allows for the fast, real-time processing of data.

As a fully managed service, provisioning, replication, failover, and patching are all automated. You can also monitor instances and set up alerts with Cloud Monitoring.

Memorystore is protected from the internet through the use of VPC networks and private IP. It also integrates with Cloud Identity and Access Management.

For more information, see:
https://cloud.google.com/memorystore
https://cloud.google.com/appengine/docs/flexible/python/using-redislabs-memcache
https://cloud.google.com/memorystore/docs/memcached/memcached-overview

Storage at a glance

Product	Simple Description	Ideal for	Not Ideal for
Cloud Storage	Binary/object store	Large or rarely accessed unstructured data	Structured data, building fast apps
Firestore	Real-time NoSQL database to store and sync data	Mobile, web, multi-user, IoT & real-time applications	Analytic data, heavy writes
Datastore	Scalable store for structured data	App Engine and server apps, heavy read/write	Relational or analytic data
Bigtable	High-volume, low-latency database	"Flat," heavy read/write, or analytical data	High structure or transactional data
Cloud SQL	Well-understood VM-based RDBMS	Web frameworks, existing applications	Scaling, analytics, heavy writes
Spanner	Relational database service	Low-latency transactional systems	Analytic data
BigQuery	Auto-scaling analytic data warehouse	Interactive analysis of static datasets	Building fast apps

Google Cloud

Here are the Google Cloud storage options at a glance. When you are choosing the right storage option for your application, it's important to understand what a product is and isn't ideal for by design.

This slide includes a simple description of the products, as well as use cases that are ideal for each product. Use cases that are not ideal for each product are also listed.

Storage at a glance

Product	Read/Write latency	Typical size	Storage type
Cloud Storage	Medium (100s of ms)	Any	Object
Firestore	Medium (10s of ms)	<200 TB	Document
Datastore	Medium (10s of ms)	< 200 TB	Document
Bigtable	Low (ms)	2 TB - 10 PB	Key-Value
Cloud SQL	Low (ms)	< 30 TB	Relational
Spanner	Low (ms)	Any	Relational
BigQuery	High (s)	Any	Columnar

Google Cloud

Other considerations for choosing a storage option for your application include the interface, read/write latency, typical size of your data, and storage type. Refer to the table to identify the best storage option for your application.

Quiz

My application requires mobile SDKs. Which storage option should I consider?

A. Cloud Storage for Firebase

B. Cloud Spanner

C. Cloud SQL

D. BigQuery

Question: My application requires mobile SDKs. Which storage option should I consider?

a. Cloud Storage for Firebase
b. Cloud Spanner
c. Cloud SQL
d. BigQuery

Quiz

My application requires mobile SDKs. Which storage option should I consider?

A. Cloud Storage for Firebase

B. Cloud Spanner

C. Cloud SQL

D. BigQuery

Google Cloud

If your application requires mobile SDKs, you should consider the Cloud Storage for Firebase option. (A)

Quiz

My application has heavy read/write requirements and my workload is analytics. What storage option should I consider for low-latency updates?

A. Cloud SQL

B. BigQuery

C. Cloud Storage

D. Cloud Bigtable

Question: My application has flat data and my workload is analytics. What storage option should I consider for low-latency updates?

a. Cloud SQL
b. BigQuery
c. Cloud Storage
d. Cloud Bigtable

Quiz

My application has heavy read/write requirements and my workload is analytics. What storage option should I consider for low-latency updates?

A. Cloud SQL

B. BigQuery

C. Cloud Storage

D. Cloud Bigtable

Google Cloud

If your application has flat data and your workload is analytics, you should consider Cloud Bigtable for low-latency updates. (D)

Quiz

My application has structured non-relational data. What storage option should I consider if I don't require mobile SDKs?

A. Firebase Realtime DB

B. Cloud Bigtable

C. Datastore

D. Cloud Storage

Question: My application has structured non-relational data. What storage options should I consider if I don't require mobile SDKs?

a. Firebase Realtime DB
b. Cloud Bigtable
c. Datastore
d. Cloud Storage

Quiz

My application has structured non-relational data. What storage option should I consider if I don't require mobile SDKs?

A. Firebase Realtime DB

B. Cloud Bigtable

C. Datastore

D. Cloud Storage

If your application has structured non-relational data, you should consider Datastore for storage if you don't require mobile SDKs. (C)

Quiz

My application has structured relational data. What storage option should I consider if my application requires horizontal scalability?

A. Cloud SQL

B. Cloud Spanner

C. Cloud Storage

D. BigQuery

Google Cloud

Question: My application has structured relational data. What storage option should I consider if my application requires horizontal scalability?

a. Cloud SQL
b. Cloud Spanner
c. Cloud Storage
d. BigQuery

Quiz

My application has structured relational data. What storage option should I consider if my application requires horizontal scalability?

A. Cloud SQL

B. Cloud Spanner

C. Cloud Storage

D. BigQuery

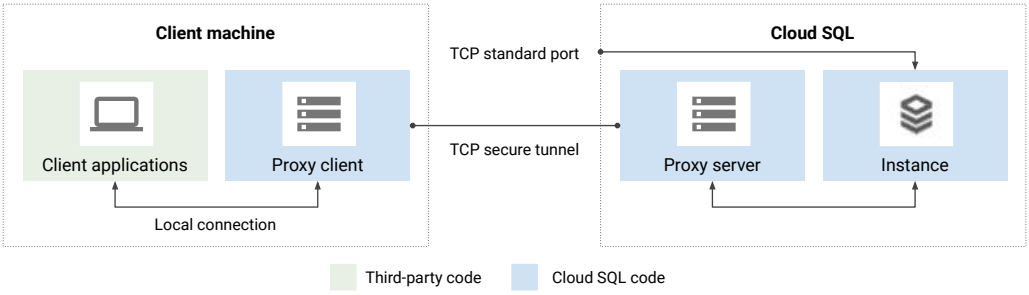
Google Cloud

If your application has structured relational data, you should consider Cloud Spanner for storage if your application requires horizontal scalability. (B)

Connect securely to a Cloud SQL database by using Cloud SQL Proxy

Access your Cloud SQL Second Generation instance without having to:

- Set an Allow rule for IP addresses
- Configure SSL



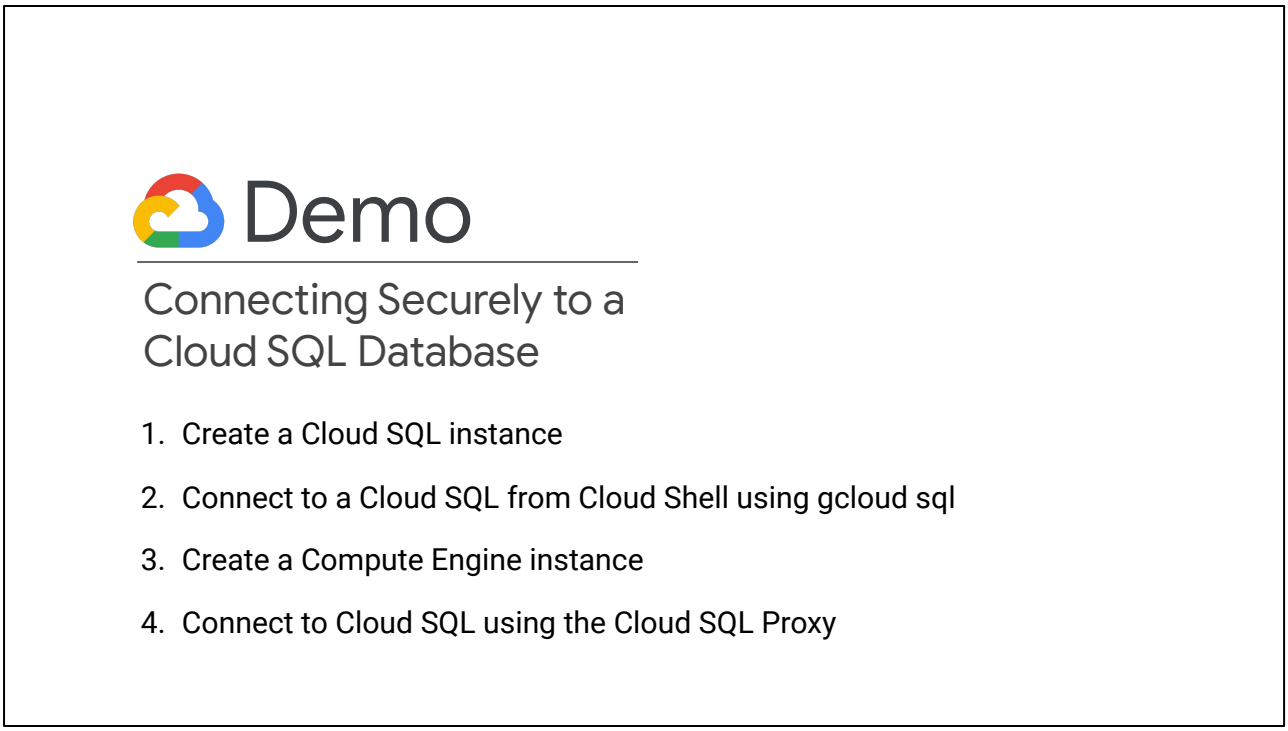
Google Cloud

Before we finish this module, we will explore how to connect securely to a Cloud SQL Database by using Cloud SQL Proxy.

The proxy automatically encrypts traffic to and from the database; SSL certificates are used to verify client and server identities. The proxy handles authentication with Cloud SQL so you don't have to provide static IP addresses. The proxy uses the Cloud SQL API to authenticate with Google Cloud - you must enable the API before using the proxy and provide the proxy with a valid user account.

The diagram shows how the proxy connects to Cloud SQL.

For more information, see: https://cloud.google.com/sql/docs/mysql/sql-proxy



Demo: Connecting Securely to a Cloud SQL Database


Google Cloud