In Google Kubernetes Engine, you can build your own storage solutions by attaching volumes to containers. However, these solutions require you to manage and protect your data.

Another option is to lean on Google to manage the availability and performance of your data by using Google Cloud managed storage services.

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

01 Managed and self-managed storage services

02 Cloud Storage use cases

03 Google Cloud managed database services

04 Cloud SQL and Cloud SQL Auth Proxy

05 Google Cloud storage options

In this section, titled “Using Google Cloud managed storage services with Google Kubernetes Engine,” you’ll:

- Contrast managed storage services with self-managed storage.
- Identify Cloud Storage use cases.
- Compare Google Cloud managed database services.
- Explore Cloud SQL Auth Proxy and learn how it connects to Cloud SQL from within GKE.
- And compare Google Cloud Storage options

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

01 Managed and self-managed storage services

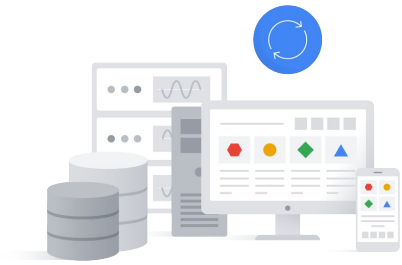
02 Cloud Storage use cases

03 Google Cloud managed database services

04 Cloud SQL and Cloud SQL Auth Proxy

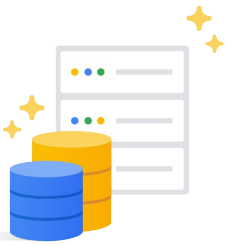
05 Google Cloud storage options

GKE supported storage options



Option 1: Store database files using a GKE Volume based on Persistent Disks.

You are responsible for managing the lifecycle and service.



Option 2: Use Google Cloud's fully managed database and storage services.

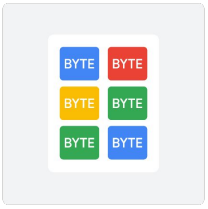
Google Kubernetes Engine offers several storage options for applications in your cluster, including storage abstractions like Volumes and PersistentVolumes.

- Let's say you're building and deploying a MySQL server as a container. To store the database files, you use a GKE Volume based on Persistent Disks. The disadvantage of this approach is that you are responsible for managing the application lifecycle and building a resilient and reliable service.
- Another option is to use Google Cloud's fully managed database and storage services.

Fully managed database and storage services

Relational databases

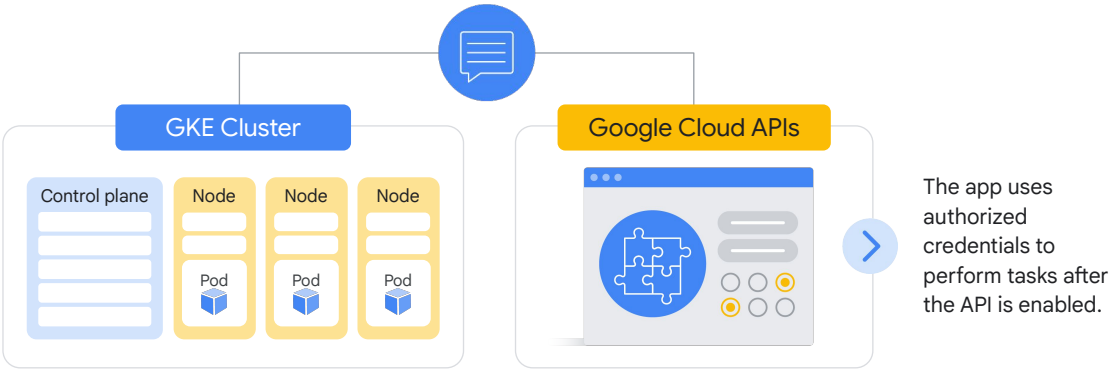
Non-relational databases



Object storage

Google Cloud offers relational database, non-relational database, and object storage services that help remove operational management burden.

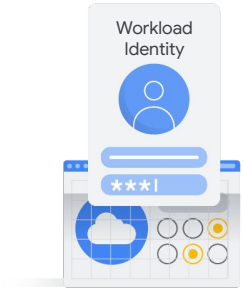
Access managed services in GKE

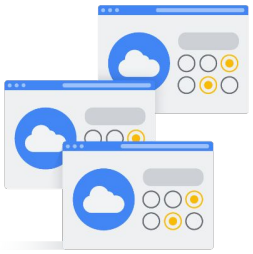


To access managed storage services, applications running in a GKE Kubernetes cluster must be capable of communicating with Google Cloud APIs.

After the relevant API is enabled, the application will use granted credentials for authentication and authorization to perform tasks.

Simplify process management with Workload Identity





Uses dedicated credentials to directly authenticate.

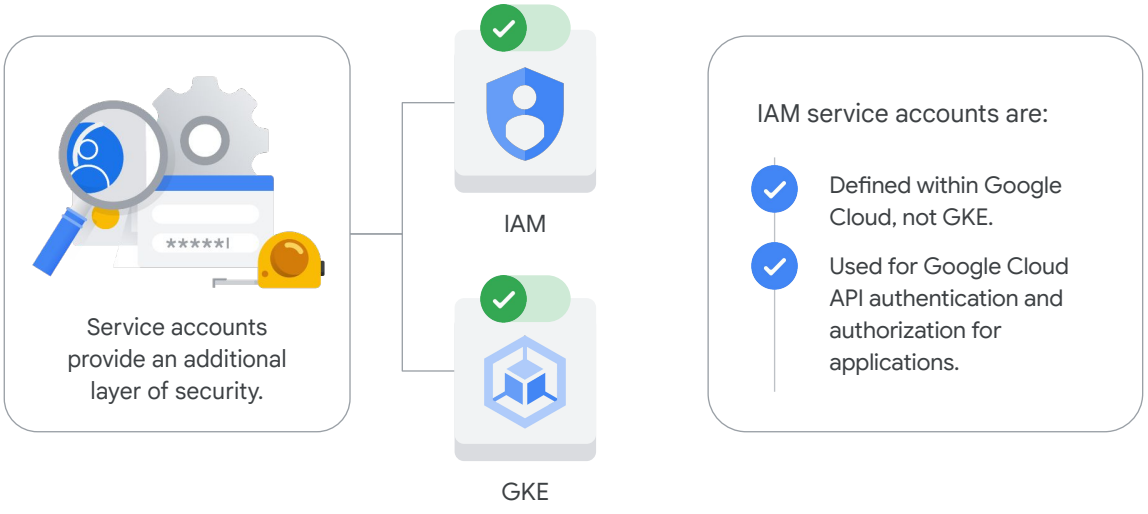
Eliminates the need to manage a collection of service account keys for each Google Cloud API you want to use.

To help, Workload Identity can be used to simplify and secure process management on your cluster.

Every action in Google Cloud must be authenticated and authorized.

Workload Identity uses dedicated credentials to directly authenticate, which eliminates the need to manage a collection of service account keys for each Google Cloud API you want to use.

Service accounts



And both GKE and IAM have service accounts to provide an additional layer of security.

IAM Service Accounts, which are defined within Google Cloud instead of GKE, are used for Google Cloud API authentication and authorization for applications.

IAM policies can bind services accounts

```
gcloud projects add-iam-policy-binding
projects/PROJECT_ID \

--role=roles/container.clusterViewer \

--member=principal://iam.googleapis.com/
projects/PROJECT_NUMBER/locations/global/
workloadIdentityPools/PROJECT_ID.svc.id.goog/
subject/ns/NAMESPACE/sa/KSA_NAME \

--condition=None
```

An IAM policy can be used to bind an IAM service account to a GKE service account.

Annotate the Kubernetes ServiceAccount object with the name of the IAM service account, and assign it to the workloads that will use the Google Cloud APIs.

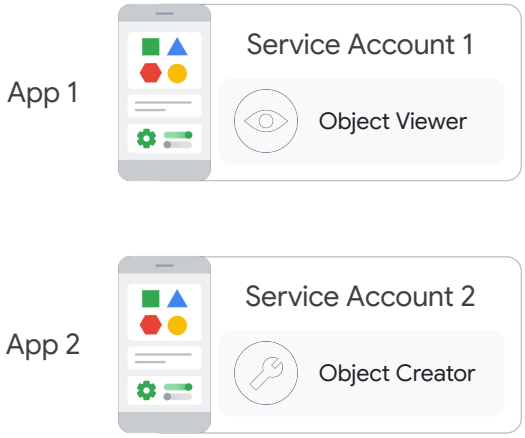
The IAM service account must have an IAM role that matches your application's required permissions.

After Workload Identity is enabled on your cluster, an IAM policy can be used to bind a IAM service account to a GKE service account.

From there, you can annotate the Kubernetes ServiceAccount object with the name of the IAM service account, and assign the GKE service account to the workloads that will use the Google Cloud APIs.

The IAM service account must have an IAM role that matches your application's required permissions.

It is a best practice to use separate service accounts



- Create a dedicated service account for each application that interacts with Google Cloud services.
- Grant each service account only the specific IAM roles and permissions it needs.
- If an attacker notices many service accounts, the value of each will appear to be less valuable.

Separate service accounts improve security and simplify the management, monitoring, and auditing of API requests.

For example, if you need to revoke API access for a specific application, you can delete the service account associated with that application instead of having to revoke access to a shared service account, which would impact multiple applications.

Instead of using the default Compute Engine service accounts, you can create a dedicated service account for each application that interacts with Google Cloud services, then grant each service account only the specific IAM roles and permissions it needs.

Separate service accounts also provide protection in the event of a security breach. If an attacker notices many service accounts, the value of each will appear to be less valuable.

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

01 Managed and self-managed storage services

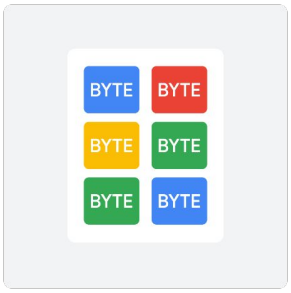
02 Cloud Storage use cases

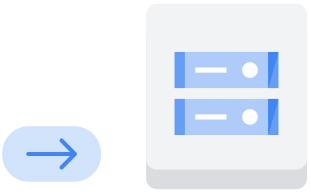
03 Google Cloud managed database services

04 Cloud SQL and Cloud SQL Auth Proxy

05 Google Cloud storage options

Cloud Storage is an object storage service





Cloud Storage

- Objects are groups of bytes.
- Cloud Storage can store bytes.
- The structure and semantics are determined by the application.

Objects in GKE are groups of bytes, and Cloud Storage is an object storage service that can store those bytes. The structure and semantics, however, are determined by the application.

Cloud Storage

Cloud Storage

- Commonly used for media hosting: serving images, streaming music and videos.
- It can be used as a data lake for analytics and machine learning workloads.
- It can store unstructured and structured files and data, but it's not optimized for querying.
- It's not designed to be a file system, so it won't replace persistent volumes.

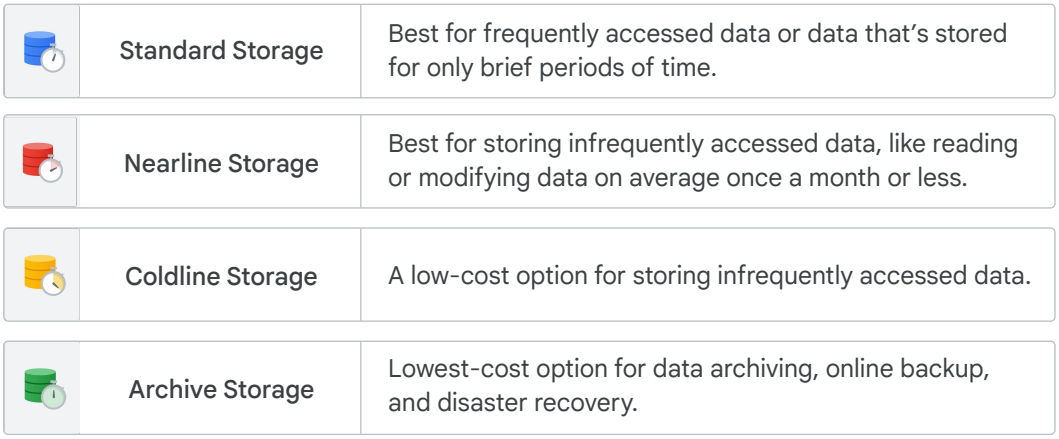
Cloud Storage is commonly used for media hosting, such as serving images for a website or streaming music and videos. For example, a mobile app deployed on GKE might need to access images from Cloud Storage buckets.

Cloud Storage can also be used as a data lake for analytics and machine learning workloads, such as genomics and data analytics.

Cloud Storage can store unstructured and structured files and data, but it is not optimized for querying.

It's also not designed to be a file system, so it doesn't replace Persistent Volumes.

Cloud Storage's four primary storage classes



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, compared to Nearline Storage, Coldline Storage is intended for reading or modifying data, at most, once every 90 days.

And 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.

Cloud Storage access control features

Encryption

Data is encrypted on the server-side with encryption keys.

Pub/Sub integration

Notifies apps when changes are made to a Cloud Storage bucket.

Versioning

Versioning can be enabled for Cloud Storage.

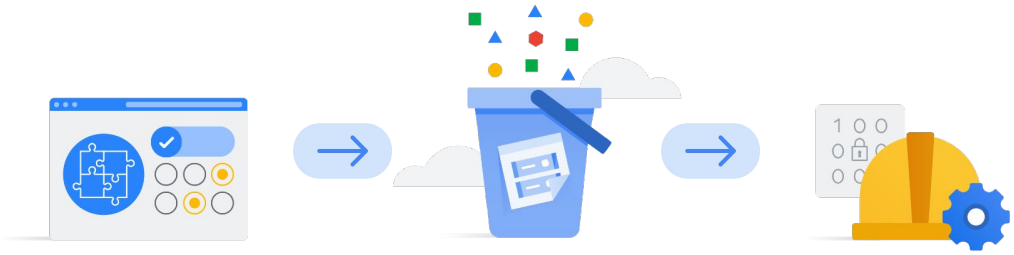
Cloud Storage offers robust access control features to protect your data.

Before it's written to disk and stored, your data is encrypted on the server-side with either customer-supplied or customer-managed encryption keys.

Cloud Storage can also be integrated with Pub/Sub, a fully-managed real-time messaging service, to notify applications whenever changes are made to a Cloud Storage bucket.

And you can also enable versioning for Cloud Storage.

Use APIs to access Cloud Storage



After Google Cloud APIs are turned on for your applications...

...they can be used to access Cloud Storage.

Important:

Your application will need to use a service account to access the service you want to use.

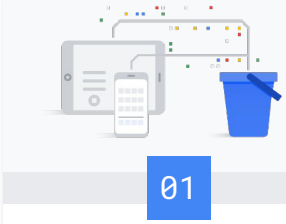
After Google Cloud APIs are turned on for your applications, they can be used to access Cloud Storage.

Please note that your application will need to use a service account to access the service you want to use.

Cloud Storage ensures any data written is instantly available for reading. This is ideal for Kubernetes applications where different Pods need to quickly and reliably share information.

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

Google Cloud's database purposes



01

Support online application data storage and retrieval

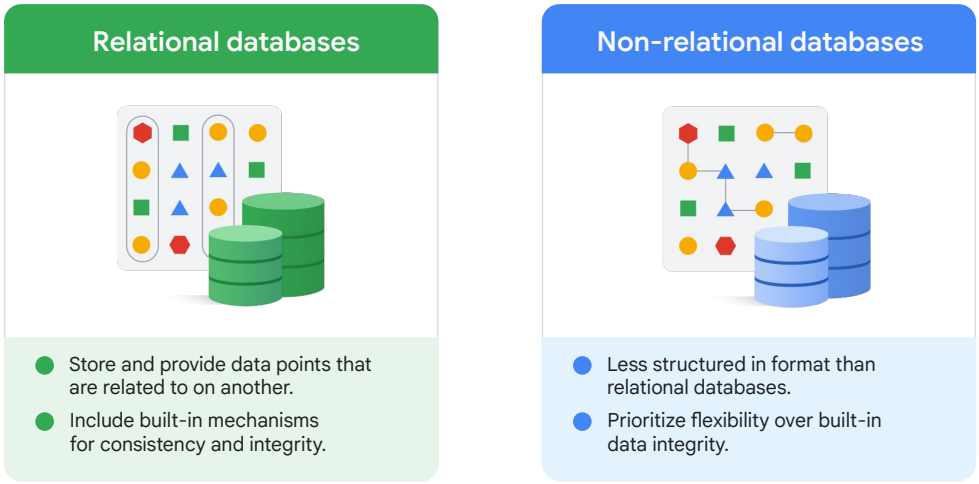
02

Provide an analytics database optimized for data mining and trend analysis

Google Cloud's database services primarily serve two purposes.

The first is to support online application data storage and retrieval, and the second is to provide an analytics database optimized for data mining and trend analysis

Google Cloud's database options

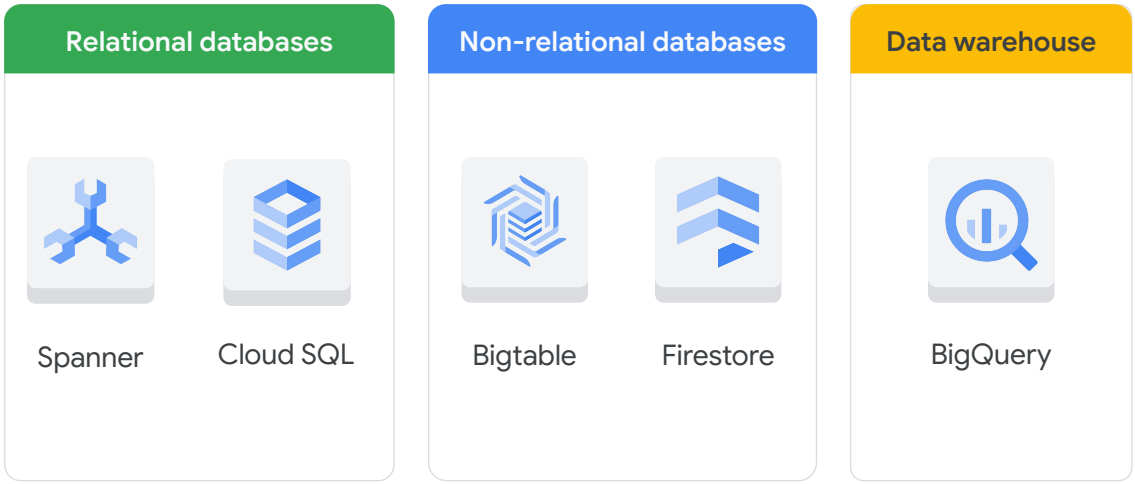


Google Cloud offers relational and non-relational database options for applications.

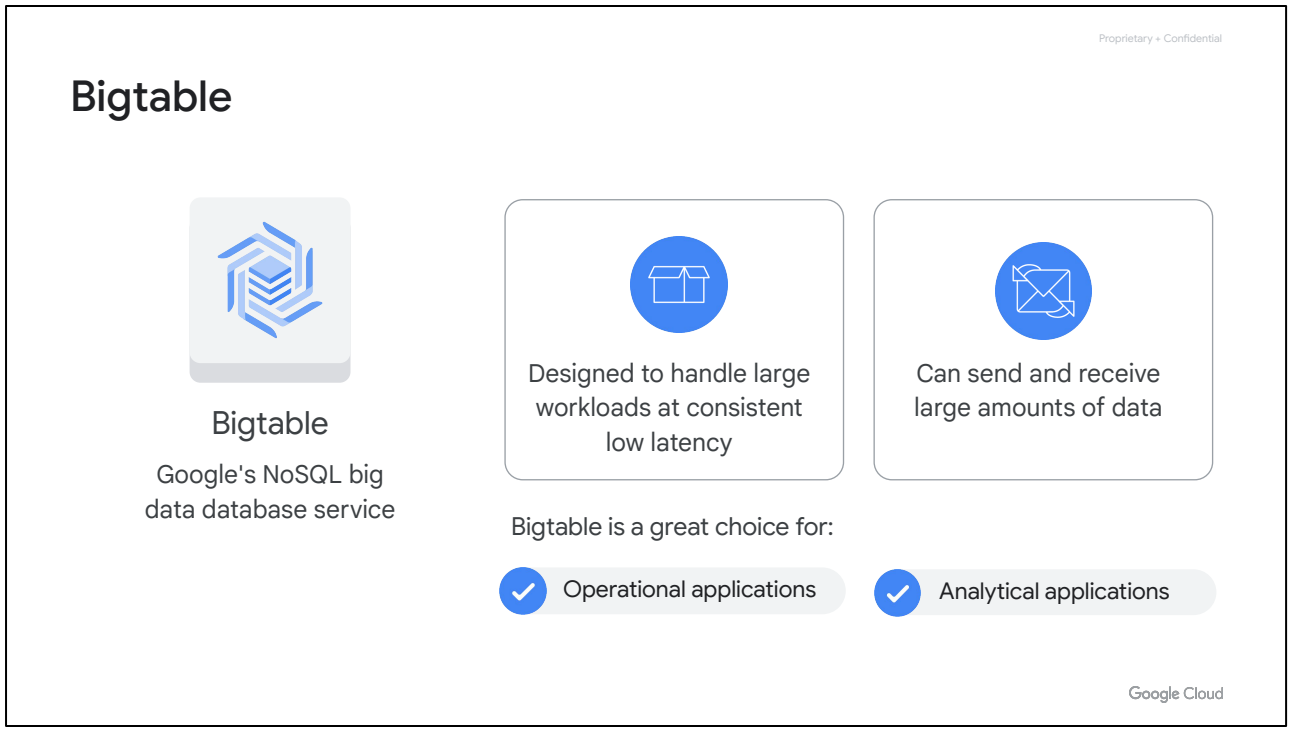
SQL-based relational databases store and provide access to data points that are related to one another. This includes built-in mechanisms to ensure the consistency and integrity of your database structure.

A non-relational database, sometimes known as a NoSQL database, is less structured in format and prioritizes flexibility over built-in data integrity enforcement. This means your application needs to take on the task of ensuring data quality.

Google Cloud database products



Let's explore five Google Cloud database services– Spanner, Cloud SQL, Bigtable, Firestore, and BigQuery.



Bigtable, Google Cloud's NoSQL database service, is designed to handle massive workloads at consistent low latency and high throughput.

Bigtable powers many Google services like Search, Analytics, Maps, and Gmail, and is an excellent option for operational and analytical applications.

Consider Bigtable for:

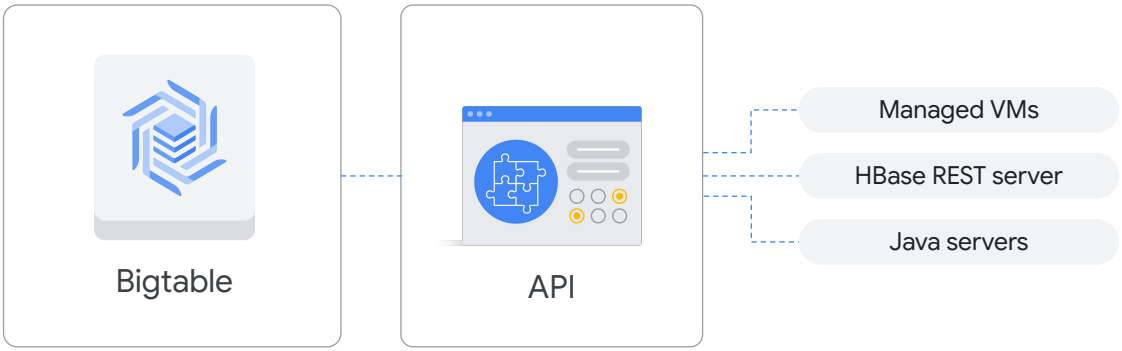
Bigtable

- Storing and accessing more than one terabyte of semi-structured or structured data.
- High throughput or rapidly changing data.
- Data transactions where strong relational semantics are not required.
- Time-series data or data with natural semantic ordering.
- Running asynchronous batch or synchronous real-time processing on big data.
- Running machine learning algorithms on the data.

When choosing which storage option to use, consider Bigtable for:

- Storing and accessing more than one terabyte of semi-structured or structured data.
- High throughput or rapidly changing data.
- Data transactions where strong relational semantics are not required.
- Time-series data or data with natural semantic ordering.
- Running asynchronous batch or synchronous real-time processing on big data.
- Running machine learning algorithms on the data.

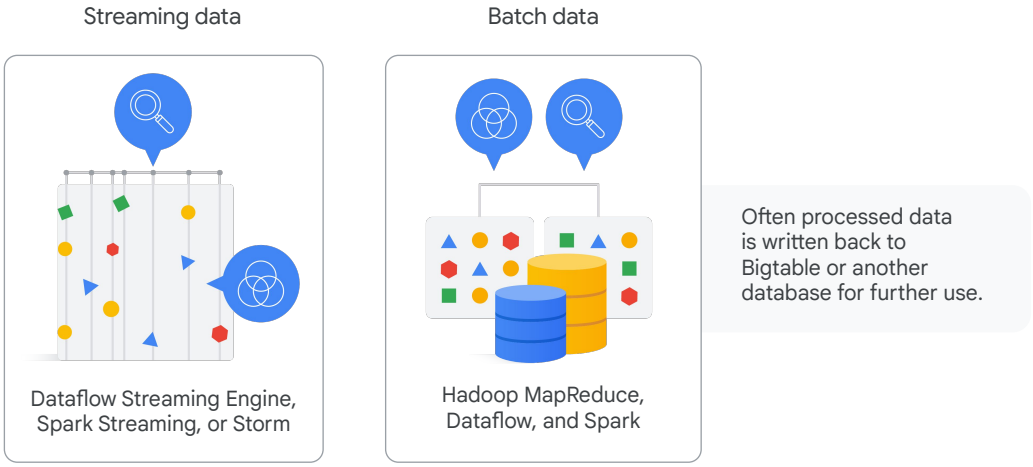
APIs and Bigtable



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

APIs let you read and write Bigtable data by using data service layers like Managed VMs, HBase REST Server, or Java Servers with the HBase client. This data is commonly used by applications, dashboards, and data services.

Bigtable supports streaming and batch data



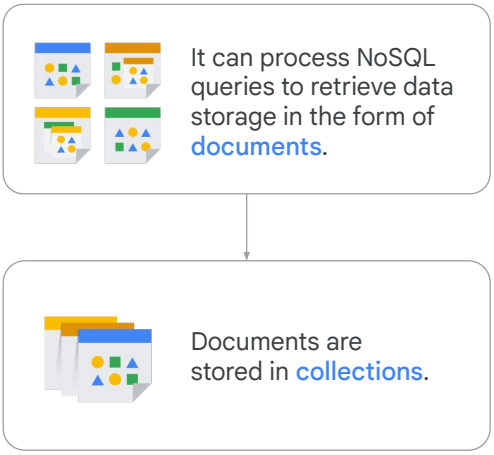
Bigtable supports both streaming and batch data ingestion.

For real-time data, you can use popular frameworks like Dataflow Streaming Engine, Spark Streaming, or Storm, and for batch processing, Hadoop MapReduce, Dataflow, and Spark are options.

It's worth noting that often processed data, whether summarized or newly calculated, is written back to Bigtable or another database for further use.

Firestore

Firestore is a flexible and scalable non-relational database commonly used for mobile, web, and server development.



Now let's switch your focus to Firestore, which is a flexible and scalable non-relational database commonly used for mobile, web, and server development.

Firestore can process NoSQL queries to retrieve a single or multiple documents within a collection that matches your criteria.

Firestore features

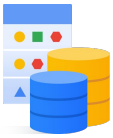
Firestore

- Queries can be filtered and sorted in Firestore, and are indexed by default.
- Query performance is dependent on the result set instead of the dataset.
- Firestore syncs data across all connected devices, but it's also optimized for simple, one-time queries.
- Firestore uses data caching to enable an application to write, read, monitor, and query data even while a device is offline.

Queries can be filtered and sorted in FireStore, and are indexed by default, so query performance is dependent on the result set instead of the dataset.

Firestore syncs data across all connected devices, but it's also optimized for simple, one-time queries. It uses data caching to enable an application to write, read, monitor, and query data even while a device is offline.

Spanner



A SQL-based, fully managed, scalable relational database.

It's the same database that powers Google's mission-critical apps and services.

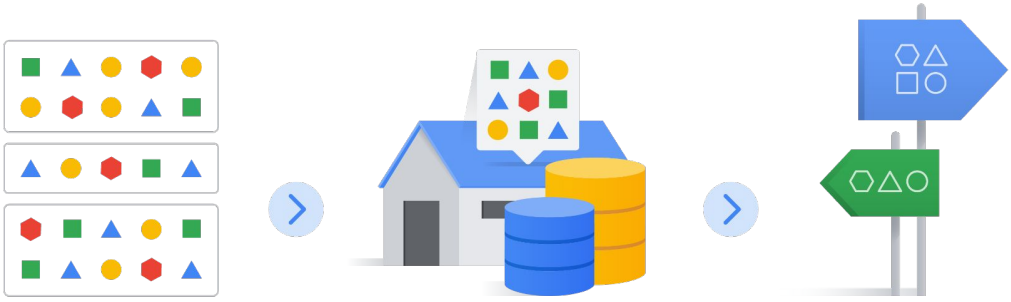
- Has high availability.
- Is globally consistent.
- Is capable of performing tens of thousands of read/write operations per second.

Next up is Spanner, which is a SQL-based, fully managed, scalable relational database. It's the same database that powers Google's mission-critical applications and services.

Spanner has high availability, is globally consistent, and is capable of performing tens of thousands of read/write operations per second.

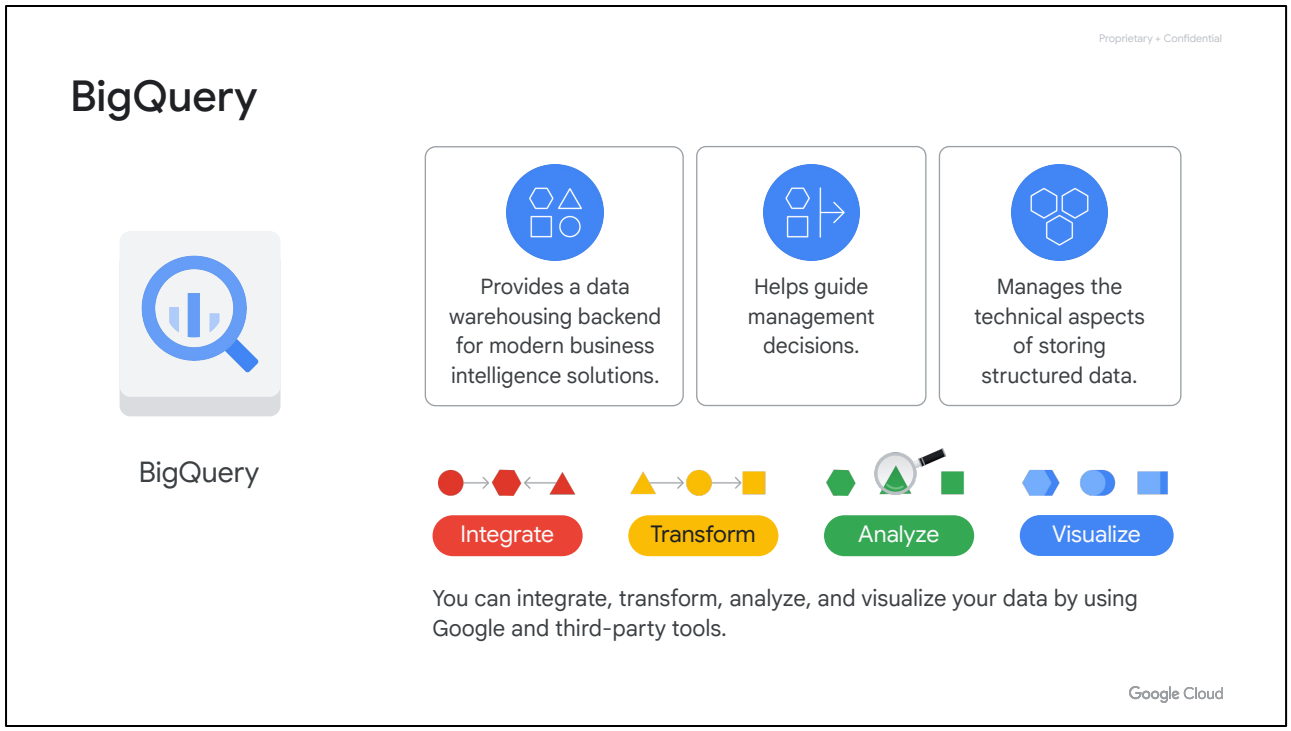
We'll discuss the other relational database, Cloud SQL, shortly.

BigQuery



BigQuery is a fully-managed data warehouse.

And finally, there is BigQuery, which is Google Cloud's fully managed data warehouse offering.



It provides a data warehousing backend for modern business intelligence solutions, and is used to guide management decisions.

Because it's fully managed, BigQuery manages the technical aspects of storing structured data, such as compression, encryption, replication, performance tuning, and scaling.

With BigQuery, you can integrate, transform, analyze, and visualize your data by using Google and third-party tools.

BigQuery features

BigQuery

- Excels at storing and analyzing massive amounts of event and sensor data generated by IoT applications.
- Optimized for 'write-once, read-many' data patterns.
- Separates storage and compute costs. You only pay for query execution, not for data at rest.

BigQuery excels at storing and analyzing massive amounts of event and sensor data generated by IoT (Internet of Things) applications. Its optimization for 'write-once, read-many' data patterns makes it ideal for handling this specific type of workload.

And it even separates storage and compute costs so; you only pay for query execution, not for data at rest.

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

01 Managed and self-managed storage services

02 Cloud Storage use cases

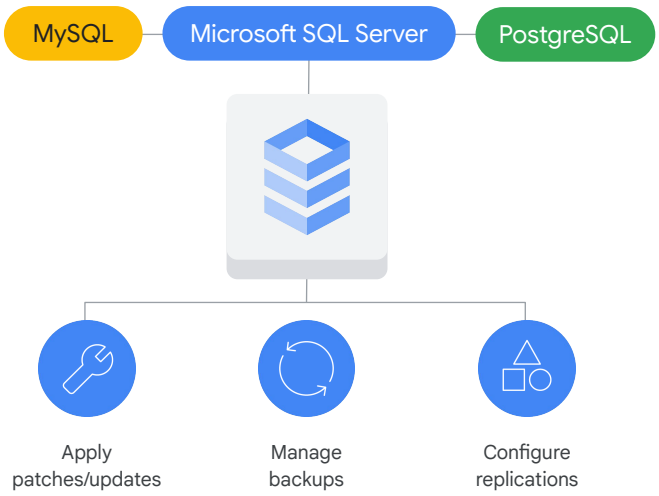
03 Google Cloud managed database services

04 Cloud SQL and Cloud SQL Auth Proxy

05 Google Cloud storage options

Cloud SQL

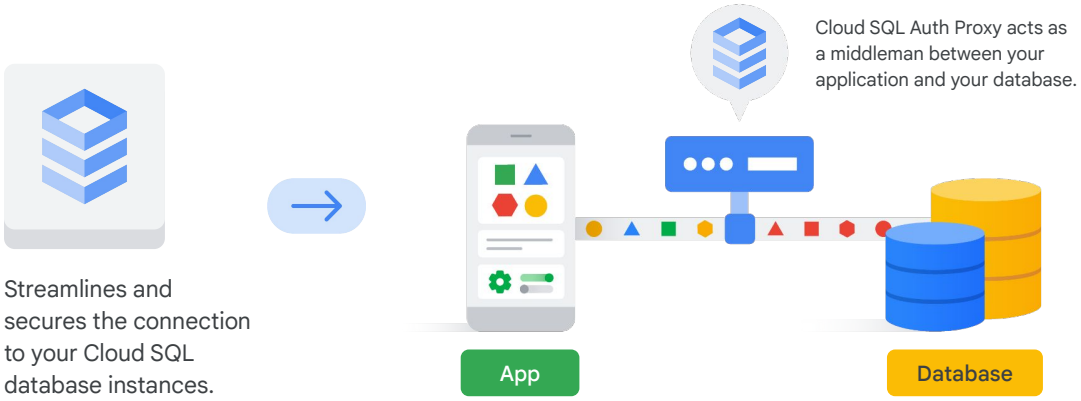
A fully managed database service that simplifies the process of setting up, maintaining, and scaling relational databases.



Cloud SQL is a fully managed database service that simplifies the process of setting up, maintaining, and scaling relational databases.

Available for MySQL, Microsoft SQL Server, and PostgreSQL, Cloud SQL is 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.

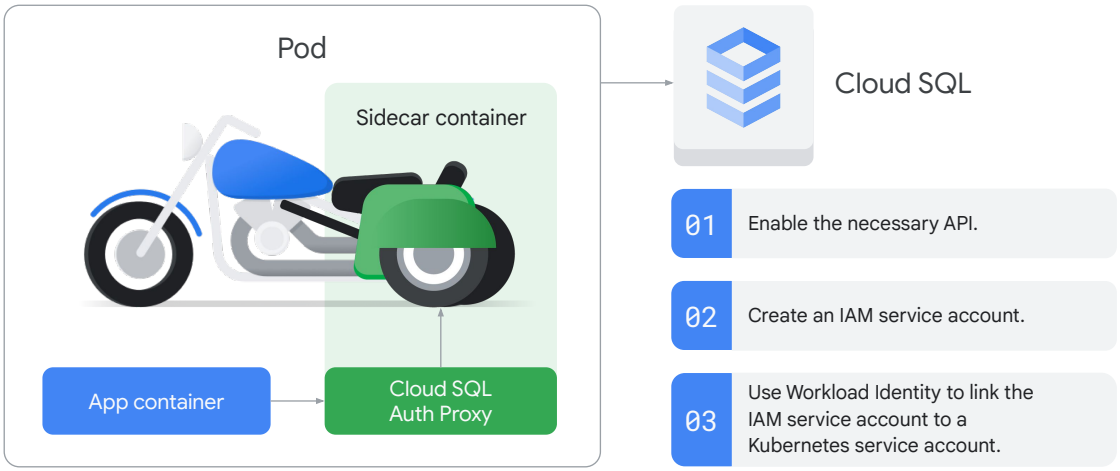
Cloud SQL Auth Proxy



And then there is the Cloud SQL Auth Proxy, which is a tool designed to streamline and secure connections to your Cloud SQL database instances. It acts as a middleman between your application and your database, eliminating the need to manage complex network configurations or expose your database directly to the internet.

In-transit traffic is automatically encrypted, and authentication is handled with SQL.

All Pods can reliably access the database



All Pods that compose your application can reliably access the database, even if they're dynamic.

In GKE, Cloud SQL Auth Proxy is set up as a “sidecar” container in the same Pod that contains your application.

Your application can communicate with the Cloud SQL Auth Proxy container using the localhost network address.

To complete this setup, you'll first need to enable the necessary API. This includes Cloud SQL API, and sqladmin API.

Next, create an IAM service account.

And third, use Workload Identity to link the IAM service account to a Kubernetes service account. Be sure to use Secrets to securely provide credentials to your Pods.

Using Google Cloud Managed Storage Services with Google Kubernetes Engine

01 Managed and self-managed storage services

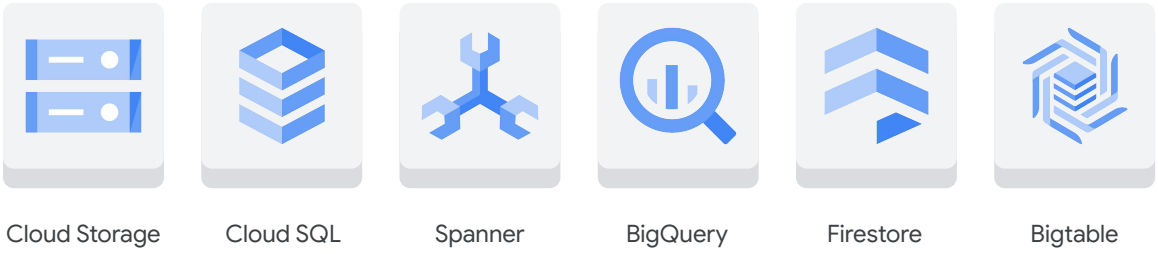
02 Cloud Storage use cases

03 Google Cloud managed database services

04 Cloud SQL and Cloud SQL Auth Proxy

05 Google Cloud storage options

Choosing a storage option

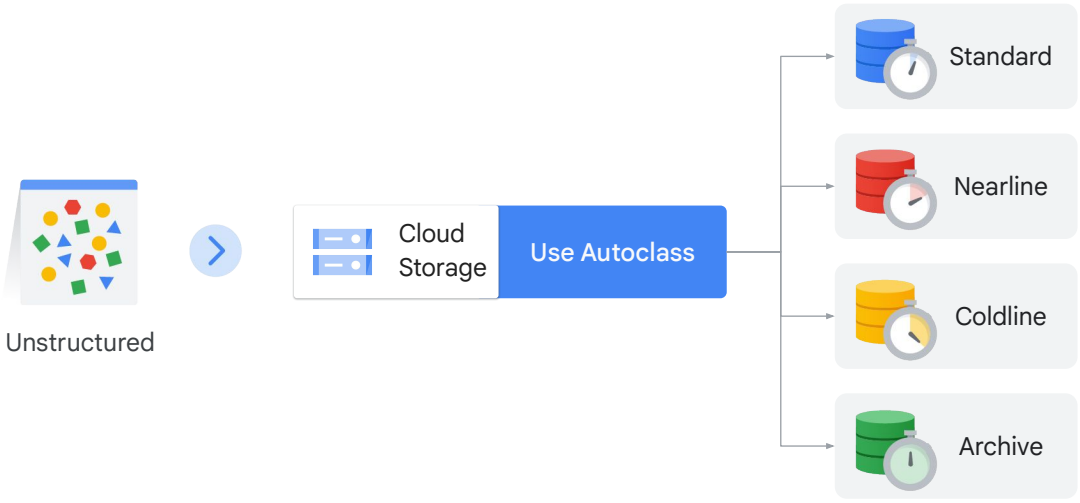


Data type + Business need

So, you've learned about the different storage options that Google Cloud offers, but in what scenarios should you use each one?

Ultimately, it's a combination of the data type that needs to be stored and the business need.

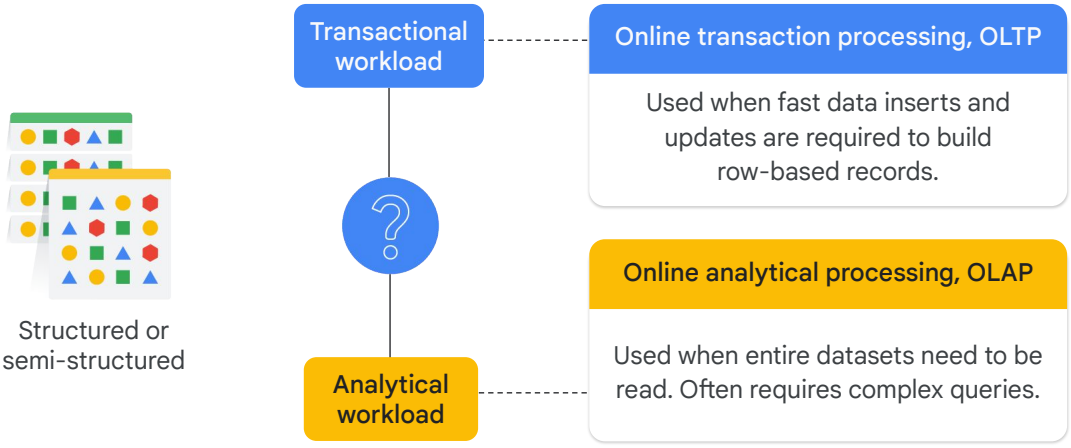
Use Cloud Storage for unstructured data



If data is unstructured, then Cloud Storage is the most appropriate option.

You have to decide on a storage class: Standard, Nearline, Coldline, or Archive. Or whether to let the Autoclass feature decide that for you.

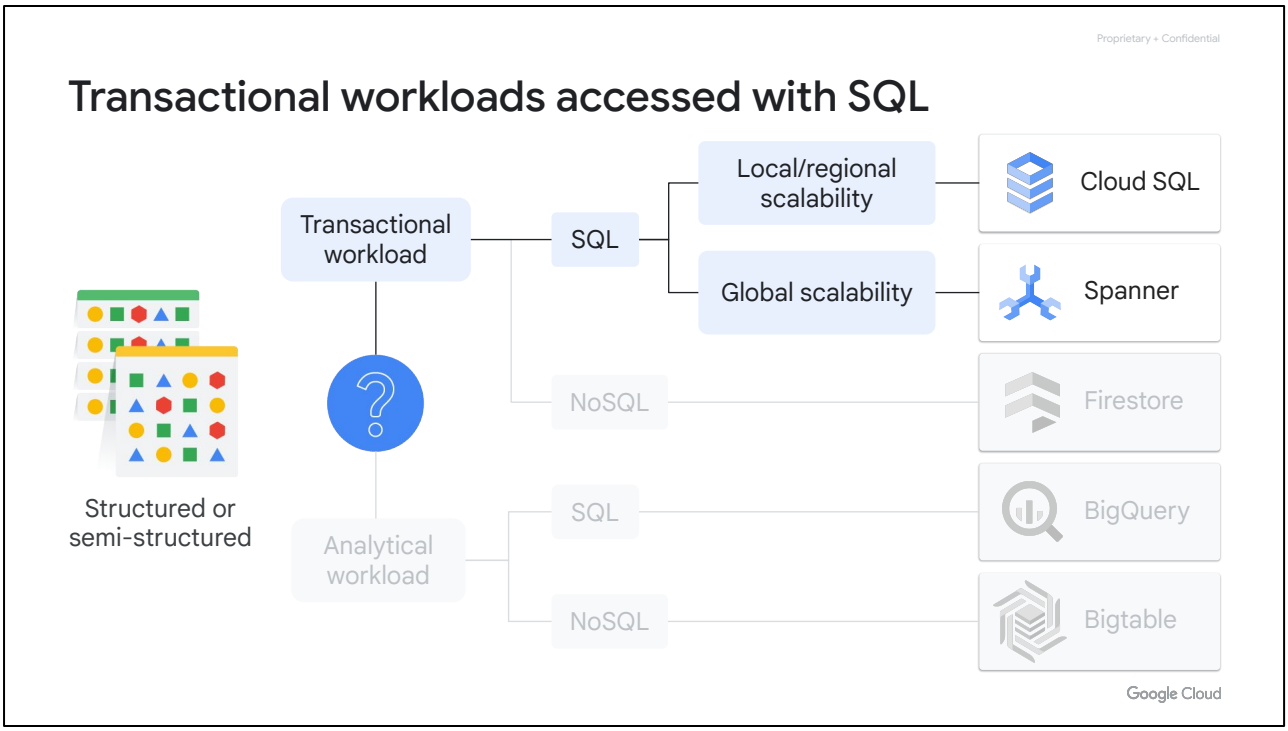
Structured or semi-structured data



If data is structured or semi-structured, choosing a storage product will depend on whether workloads are transactional or analytical.

Transactional workloads stem from online transaction processing, or OLTP, systems, which are used when fast data inserts and updates are required to build row-based records. An example of this is point-of-sale transaction records.

Then there are analytical workloads, which stem from online analytical processing, or OLAP systems, which are used when entire datasets need to be read. They often require complex queries, for example, aggregations. An example here would be analyzing sales history to see trends and aggregated views.

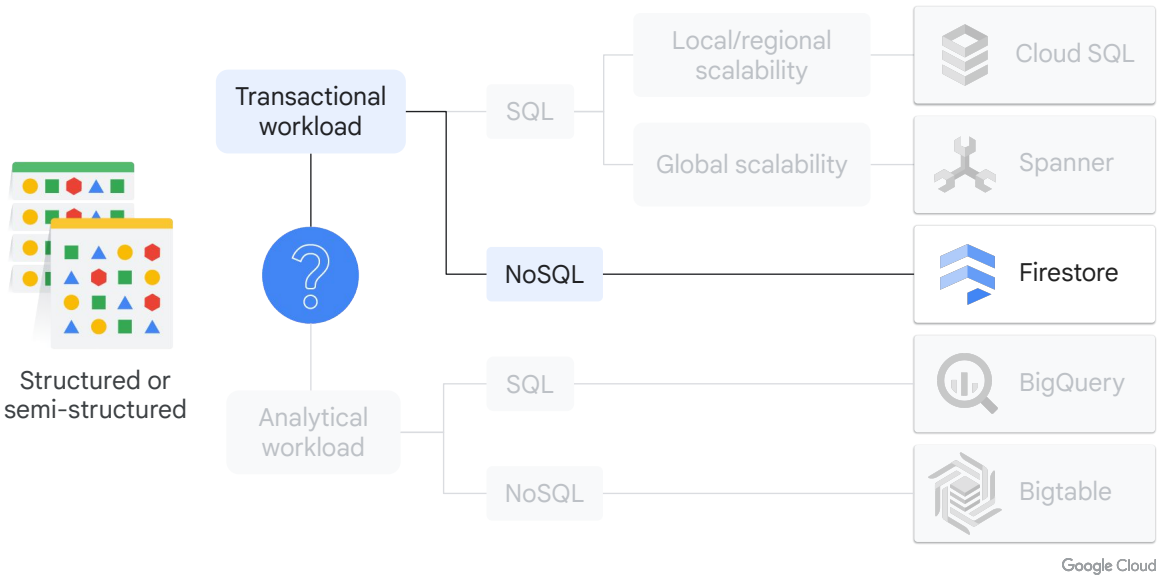


After you determine if the workloads are transactional or analytical, you must determine whether the data will be accessed by using SQL.

If your data is transactional and you need to access it by using SQL, then Cloud SQL and Spanner are two options.

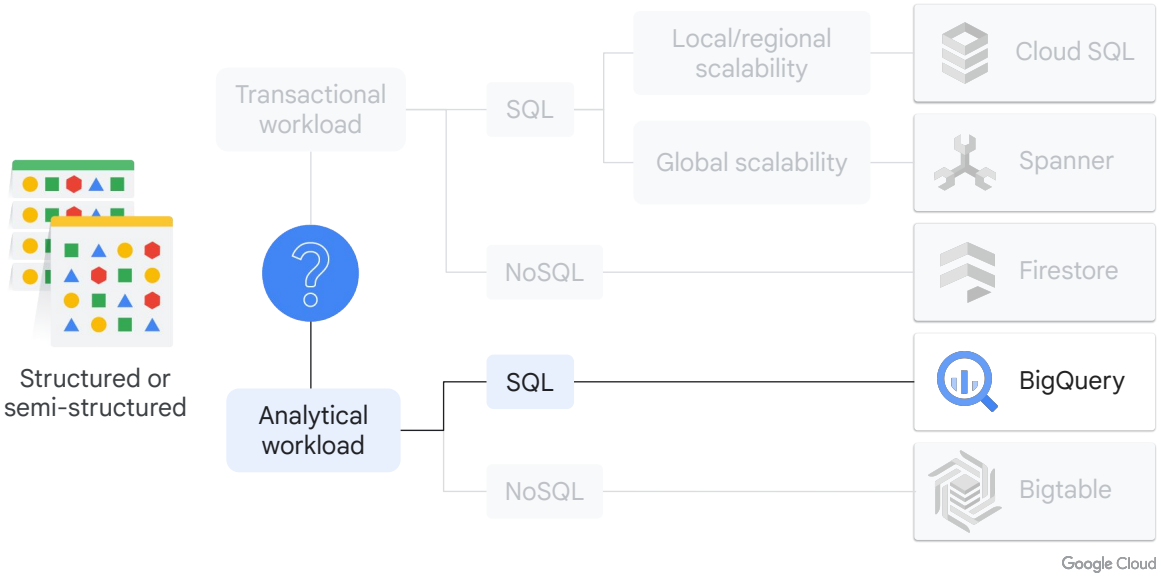
Cloud SQL works best for local to regional scalability, and Spanner is best to scale a database globally.

Transactional workloads accessed without SQL



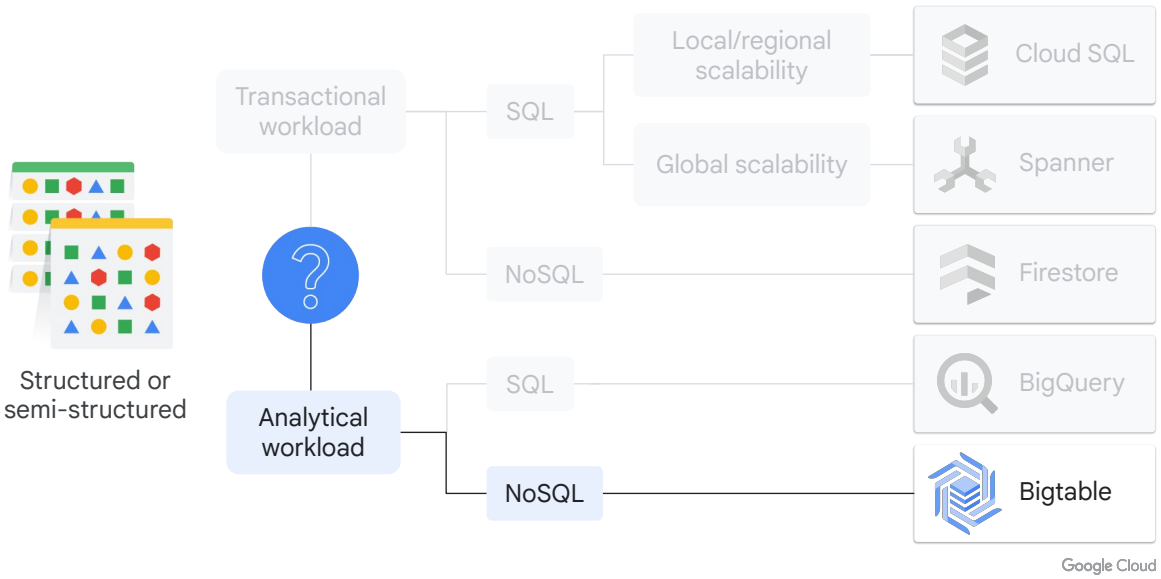
If the transactional data will be accessed without SQL, Firestore might be the best option. Firestore is a transactional NoSQL, document-oriented database.

Analytical workloads that require SQL commands



If you have analytical workloads that require SQL commands, BigQuery might be the best option. BigQuery, Google's data warehouse solution, lets you analyze petabyte-scale datasets.

Analytical workloads that don't require SQL commands



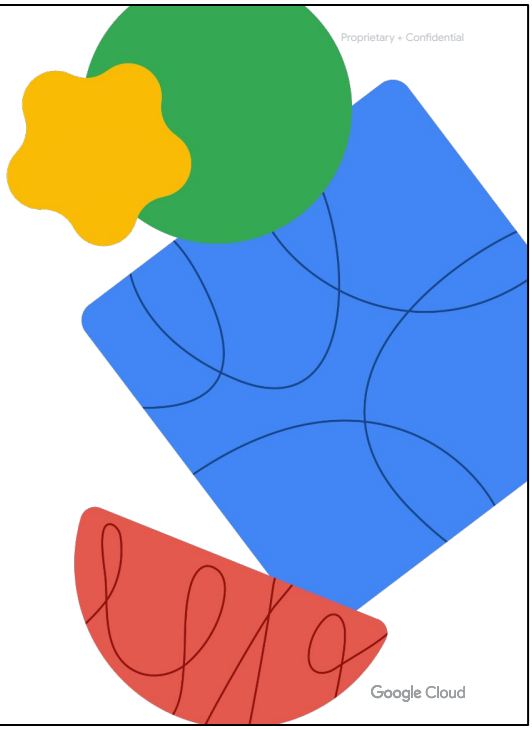
Alternatively, Bigtable provides a scalable NoSQL solution for analytical workloads. It's best for real-time, high-throughput applications that require only millisecond latency.

You can use Bigtable to store and query all of the following types of data:

- Time-series data, such as CPU and memory usage over time for multiple servers
- Marketing data, such as purchase histories and customer preferences
- Financial data, such as transaction histories, stock prices, and currency exchange rates
- Internet of Things data, such as usage reports from energy meters and home appliances
- Graph data, such as information about how users are connected to one another

Quiz questions

Let's pause for a quick check in.



Quiz | Question 1

Question

How do you enable a GKE Pod to securely access a Google Cloud storage service?

A. Give the application the credentials of a Google IAM account with the necessary permissions to access the Google Cloud service.

B. Deploy a proxy container with the service's credentials in the same Pod as your application.

C. Grant the cluster's Kubernetes Service Account the necessary permissions to access the Google Cloud service using Workload Identity.

D. Add a DNS SRV record for the managed service to the GKE cluster's DNS server.

Quiz | Question 1

Answer

How do you enable a GKE Pod to securely access a Google Cloud storage service?

A. Give the application the credentials of a Google IAM account with the necessary permissions to access the Google Cloud service.

B. Deploy a proxy container with the service's credentials in the same Pod as your application.

C. Grant the cluster's Kubernetes Service Account the necessary permissions to access the Google Cloud service using Workload Identity.

D. Add a DNS SRV record for the managed service to the GKE cluster's DNS server.

How do you enable a GKE Pod to securely access a Google Cloud storage service?

A. Give the application the credentials of a Google IAM account with the necessary permissions to access the Google Cloud service.: This is the **incorrect answer** because embedding credentials directly in the application is generally discouraged due to security risks.
B. Deploy a proxy container with the service's credentials in the same Pod as your application.: This is the **incorrect answer** because this requires careful management of credentials within the proxy.
C. Grant the cluster's Kubernetes Service Account the necessary permissions to access the Google Cloud service using Workload Identity.: This is the **correct answer** because Workload Identity is a recommended way to securely connect GKE workloads to Google Cloud services. It allows your GKE service accounts to act as IAM service accounts, granting them the necessary permissions without embedding credentials directly in your application..
D. Add a DNS SRV record for the managed service to the GKE cluster's DNS server.: This is the **incorrect answer** because DNS SRV records help with service discovery, but they don't address the core issue of secure authentication and authorization..

Quiz | Question 2

Question

You need a cloud storage solution for archiving business data to comply with regulations. The archives need to be kept for 5 years, audited quarterly, and rarely accessed otherwise. What solution best fits these needs?

A. Cloud Storage - Standard

B. Cloud Storage - Coldline

C. Cloud Storage - Archive

D. Cloud Storage - Nearline

Quiz | Question 2

Answer

You need a cloud storage solution for archiving business data to comply with regulations. The archives need to be kept for 5 years, audited quarterly, and rarely accessed otherwise. What solution best fits these needs?

A. Cloud Storage - Standard
B. Cloud Storage - Coldline 
C. Cloud Storage - Archive
D. Cloud Storage - Nearline

You need a cloud storage solution for archiving business data to comply with regulations. The archives need to be kept for 5 years, audited quarterly, and rarely accessed otherwise. What solution best fits these needs?

A. Cloud Storage - Standard: This is the **incorrect answer** because Standard storage is designed for frequently accessed data and would be more expensive.
B. Cloud Storage - Coldline: This is the **correct answer** because the data is rarely accessed, aligning with Coldline's intended use case.
C. Cloud Storage - Archive: This is the **incorrect answer** because Archive storage is designed for data archiving, online backup,
and disaster recovery
D. Cloud Storage - Nearline: This is the **incorrect answer** because while Nearline cheaper than Standard, it's still intended for data accessed more frequently than once a quarter.

Quiz | Question 3

Question

You're building an online multiplayer game that needs to propagate a consistent global game state across a massive amount of concurrent sessions. Which Google-managed storage solution should you use?

A. Cloud Bigtable
B. Cloud SQL
C. Cloud Storage
D. Firestore

Quiz | Question 3

Answer

You're building an online multiplayer game that needs to propagate a consistent global game state across a massive amount of concurrent sessions. Which Google-managed storage solution should you use?

A. Bigtable 

B. Cloud SQL

C. Cloud Storage

D. Firestore

You're building an online multiplayer game that needs to propagate a consistent global game state across a massive amount of concurrent sessions. Which Google-managed storage solution should you use?

A. Bigtable: This is the **correct answer** because Bigtable is a NoSQL wide-column database designed for high throughput and low latency, making it ideal for real-time applications with massive amounts of data. It can handle millions of reads and writes per second, which is crucial for a large-scale online game..
B. Cloud SQL: This is the **incorrect answer** because Cloud SQL is a relational database, which isn't the best fit for the schemaless, high-velocity data typical of game state updates.
C. Cloud Storage: This is the **incorrect answer** because Cloud Storage is an object store, not a database. It's not designed for the frequent reads and writes needed for real-time game state management.
D. Firestore: This is the **incorrect answer** because Firestore is also a NoSQL database, it's not optimized for the extremely high throughput and low latency required for real-time game state synchronization..

Quiz | Question 4

Question

What is the purpose of SQL Auth Proxy?

A. It is a tool that helps manage service accounts in Kubernetes.

B. It allows applications running in Kubernetes to connect to Cloud SQL databases.

C. It is a way to connect to databases that are hosted on Google Cloud.

D. It is a way to manage access to databases in Google Cloud.

Quiz | Question 4

Answer

What is the purpose of SQL Auth Proxy?

A. It is a tool that helps manage service accounts in Kubernetes.

B. It allows applications running in Kubernetes to connect to Cloud SQL databases. 

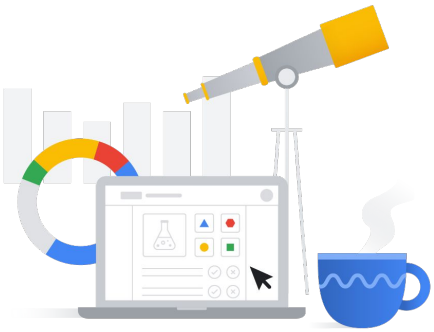
C. It is a way to connect to databases that are hosted on Google Cloud.

D. It is a way to manage access to databases in Google Cloud.

What is the purpose of SQL Auth Proxy?

A. It is a tool that helps manage service accounts in Kubernetes.: This is the **incorrect answer** because Service account management is typically handled through IAM and Kubernetes directly, not specifically through Cloud SQL Auth proxy.
B. It allows applications running in Kubernetes to connect to Cloud SQL databases.: This is the **correct answer** because Cloud SQL Auth proxy is specifically designed to streamline and secure connections between applications running in Kubernetes and Cloud SQL databases.
C. It is a way to connect to databases that are hosted on Google Cloud.: This is the **incorrect answer** because Cloud SQL Auth Proxy is a way to content to Cloud SQL databases, but not other databases that Google Cloud supports.
D. It is a way to manage access to databases in Google Cloud.: This is the **incorrect answer** because Cloud SQL Auth proxy focuses on securing the connection between the application and the database. Broader database access management is usually handled through IAM roles and database-specific permissions.

Lab: Using Cloud SQL with Google Kubernetes Engine



01 Create credentials and Kubernetes Secrets for application authentication.

02 Configure Workload Identity.

03 Configure a Deployment with a Wordpress image to use SQL Proxy.

04 Install SQL Proxy as a sidecar container and use it to provide access to Cloud SQL.

It's time for some hands-on practice with GKE.

In the lab titled "Using Cloud SQL with Google Kubernetes Engine," you'll:

- Create credentials and Kubernetes Secrets for application authentication.
- Configure Workload Identity.
- Configure a Deployment with a Wordpress image to use SQL Proxy.
- Install SQL Proxy as a sidecar container and use it to provide SSL access to a Cloud SQL instance external to the GKE Cluster.