

Storing and Analyzing Your Data in Google's Cloud

"Learn about the different ways to store your data in Google's cloud."

This document discusses the options for storing and analyzing your data in the Google Cloud Platform. This is an introductory article. If you are already familiar with the services provided by the Google Cloud Platform, you might want to dive straight into the developer documentation: [App Engine](#), [Google Cloud Storage](#), [Google Cloud SQL](#), [BigQuery](#), [Google Compute Engine](#).

Contents

- [Introduction](#)
- [Google Drive](#)
- [Google Cloud Storage](#)
- [Google Cloud SQL](#)
- [BigQuery](#)
- [App Engine Datastore](#)
- [Google Compute Engine](#)
- [Summary](#)
- [Read More](#)

Introduction

Ever seen those bumper stickers that say “my other computer is a data center?” Why settle for just one data center when you can use an entire network of [data centers](#) connected by a super fast, high-performance network? The same infrastructure that provides the robust data, rapid access, fast response times, reliability and scalability that Google uses to index the web, serve search results, run GMail, and more, is available for you to run your own applications.

For all of Google’s Cloud Platform services, there is no upfront cost, and you pay only for what you use. We keep your data safe and we make it available fast. We keep your service running, replicate and backup your data, and do all the maintenance work for you.

At a glance, here are the options for storing and analyzing your data in Google’s Cloud Platform:



You want to store your private files online and access them in a web browser. You want to choose who can view, share, and edit your files.



Google Drive



You want to store your application data, consisting of files of almost any kind and size, in the cloud.



Google Cloud Storage



You know and love MySQL, and want to host your databases in the cloud.



Google Cloud SQL



You want to interactively analyze massive datasets.



Google BigQuery



You’re developing App Engine applications and you need scalable, fast queries over your data without the schema requirements imposed by a relational database.



Google App Engine



You’re happy managing your own virtual machine.



Google Compute Engine

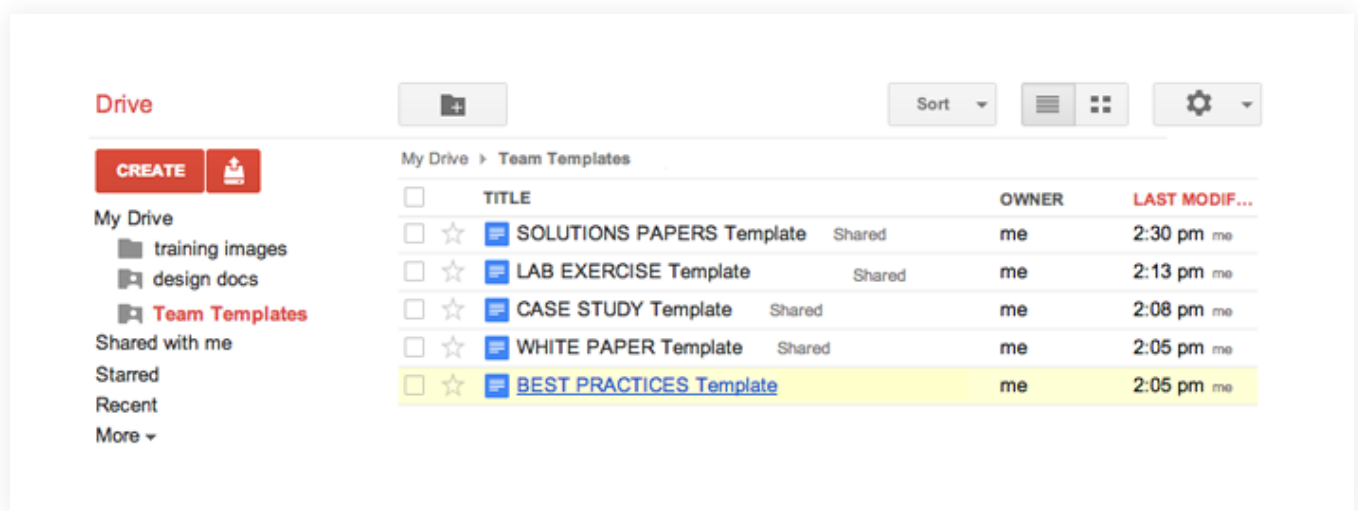
In most cases, the best solution will be a combination of these services.

The rest of this document explores each service in turn, highlighting the use cases, data storage options, and the ingestion and export options.

A note about terminology: This article uses the term “Google’s cloud” to mean Google’s infrastructure, including its data centers, networks, and software.

Google Drive

Google Drive is a service for users to store and share their private files. Google Drive is intended for use by individuals, and has a UI that offers many features for creating, editing and sharing your work, in addition to uploading files for storage.



Google Drive enables users to access and manage all their file content in the Google’s cloud and have it accessible from anywhere. While Google Drive provides an API for uploading files and for searching and retrieving stored items, the UI is intended to be the primary mechanism for interaction. If your application is working with files that have historically been stored locally on a user’s computer or phone, Google Drive is a good option.

For more information, see [Google Drive](#). The rest of this document discusses the storage and analysis options that are primarily intended for use by developers building applications.

Google Cloud Storage

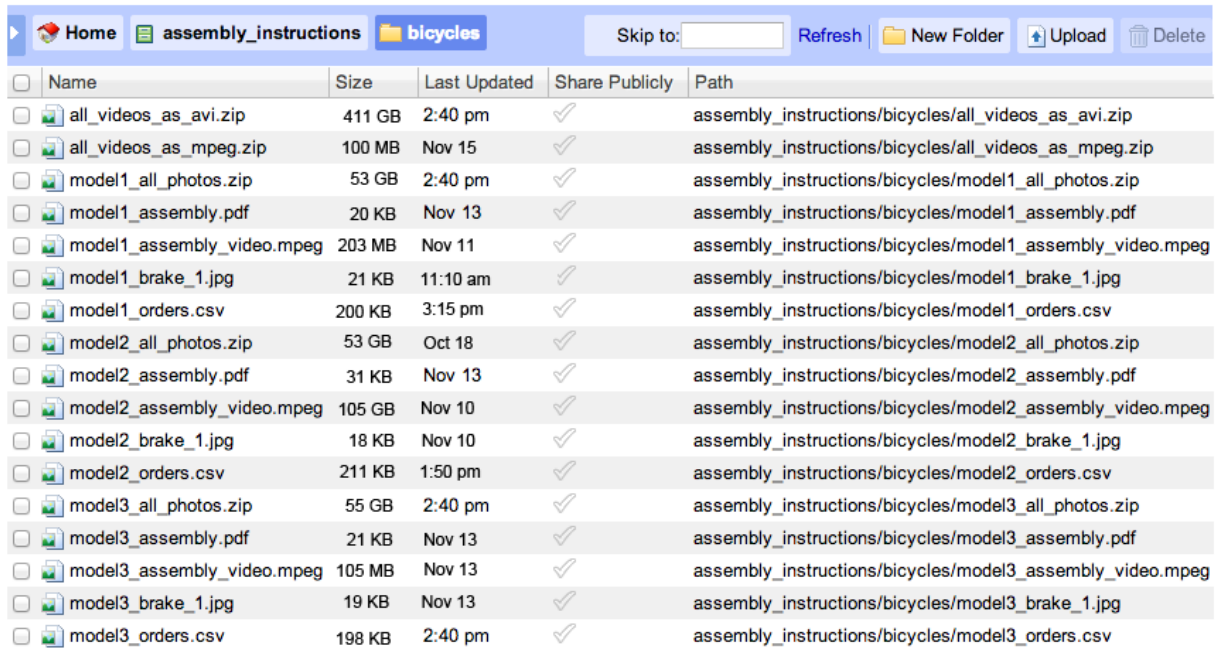
Google Cloud Storage is a service for storing and accessing data in Google’s cloud. It is primarily intended for programmatic use within applications. It has an interactive UI, which is helpful for learning about the product, getting started using it, and quickly uploading or deleting content.

Google Cloud Storage offers direct access to Google’s scalable storage and networking infrastructure, as well as powerful authentication and data sharing mechanisms. It lets you store files of any size and manage access to your data on an individual or group basis.

Data stored in Google Cloud Storage can be designated as public or private. Public data can be shared

with anyone, enabling you to use Google Cloud Storage as a conduit to making selected parts of your data available outside your company.

Google code Google Cloud Storage



The screenshot shows the Google Cloud Storage interface. The breadcrumb navigation is Home > assembly_instructions > bicycles. The interface includes a search bar, a Refresh button, and buttons for New Folder, Upload, and Delete. Below the navigation is a table listing files with columns for Name, Size, Last Updated, Share Publicly, and Path.

<input type="checkbox"/>	Name	Size	Last Updated	Share Publicly	Path
<input type="checkbox"/>	all_videos_as_avi.zip	411 GB	2:40 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/all_videos_as_avi.zip
<input type="checkbox"/>	all_videos_as_mpeg.zip	100 MB	Nov 15	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/all_videos_as_mpeg.zip
<input type="checkbox"/>	model1_all_photos.zip	53 GB	2:40 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model1_all_photos.zip
<input type="checkbox"/>	model1_assembly.pdf	20 KB	Nov 13	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model1_assembly.pdf
<input type="checkbox"/>	model1_assembly_video.mpeg	203 MB	Nov 11	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model1_assembly_video.mpeg
<input type="checkbox"/>	model1_brake_1.jpg	21 KB	11:10 am	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model1_brake_1.jpg
<input type="checkbox"/>	model1_orders.csv	200 KB	3:15 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model1_orders.csv
<input type="checkbox"/>	model2_all_photos.zip	53 GB	Oct 18	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model2_all_photos.zip
<input type="checkbox"/>	model2_assembly.pdf	31 KB	Nov 13	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model2_assembly.pdf
<input type="checkbox"/>	model2_assembly_video.mpeg	105 GB	Nov 10	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model2_assembly_video.mpeg
<input type="checkbox"/>	model2_brake_1.jpg	18 KB	Nov 10	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model2_brake_1.jpg
<input type="checkbox"/>	model2_orders.csv	211 KB	1:50 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model2_orders.csv
<input type="checkbox"/>	model3_all_photos.zip	55 GB	2:40 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model3_all_photos.zip
<input type="checkbox"/>	model3_assembly.pdf	21 KB	Nov 13	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model3_assembly.pdf
<input type="checkbox"/>	model3_assembly_video.mpeg	105 MB	Nov 13	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model3_assembly_video.mpeg
<input type="checkbox"/>	model3_brake_1.jpg	19 KB	Nov 13	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model3_brake_1.jpg
<input type="checkbox"/>	model3_orders.csv	198 KB	2:40 pm	<input checked="" type="checkbox"/>	assembly_instructions/bicycles/model3_orders.csv

Typical Use Cases

Google Cloud Storage enables developers to store their data in Google's cloud. Google Cloud Storage is ideally suited to serve as a content repository containing an unlimited number of files of any size that can be shared with others and rapidly accessed. For example, one biotechnology company uses it to store large genomic datasets and make them broadly available to the research community.

Other use cases are backing up data, as well as quickly accessing archived data. A lower-cost option is available for archiving data that does not need continuous rapid retrieval access.

In many cases, Google Cloud Storage acts as the intermediary storage facility for other services in the Google Cloud Platform. For example, it acts as the staging service for Google Cloud SQL and BigQuery to access data from other systems and export data to other systems.

For use case writeups, see cloud.google.com/products/cloud-storage.

Google Cloud Storage Data Creation and Ingestion

You don't create data in Cloud Storage as such. You store existing data in Cloud Storage.

You can upload and download files:

- interactively using the online browser
- from a command line using the gsutil tool
- programmatically using Google Cloud Storage's REST API

In addition to simply uploading or downloading data, you can serve content via HTTP directly from Google Cloud Storage. For example, you can embed a hyperlink (or paste a URL into your browser's address bar) and Google Cloud Storage serves up the content in a highly scalable fashion. You can even serve entire static web sites from Google Cloud Storage.

Google Cloud Storage API

Google Cloud Storage uses buckets to contain objects, where a bucket is similar to a directory and an object is similar to a file. The Google Cloud Storage API provides a web interface for making HTTP requests to work with buckets and objects. The Google Cloud Storage API supports HTTP methods for:

- Listing buckets
- Creating and deleting buckets
- Changing and listing who can access buckets
- Uploading and downloading objects
- Deleting objects in a bucket
- Uploading objects using HTML forms

Google Cloud SQL

Google Cloud SQL allows you to create, configure, and use MySQL databases that live in Google's cloud. It is a fully-managed service that maintains, manages, and administers your databases.

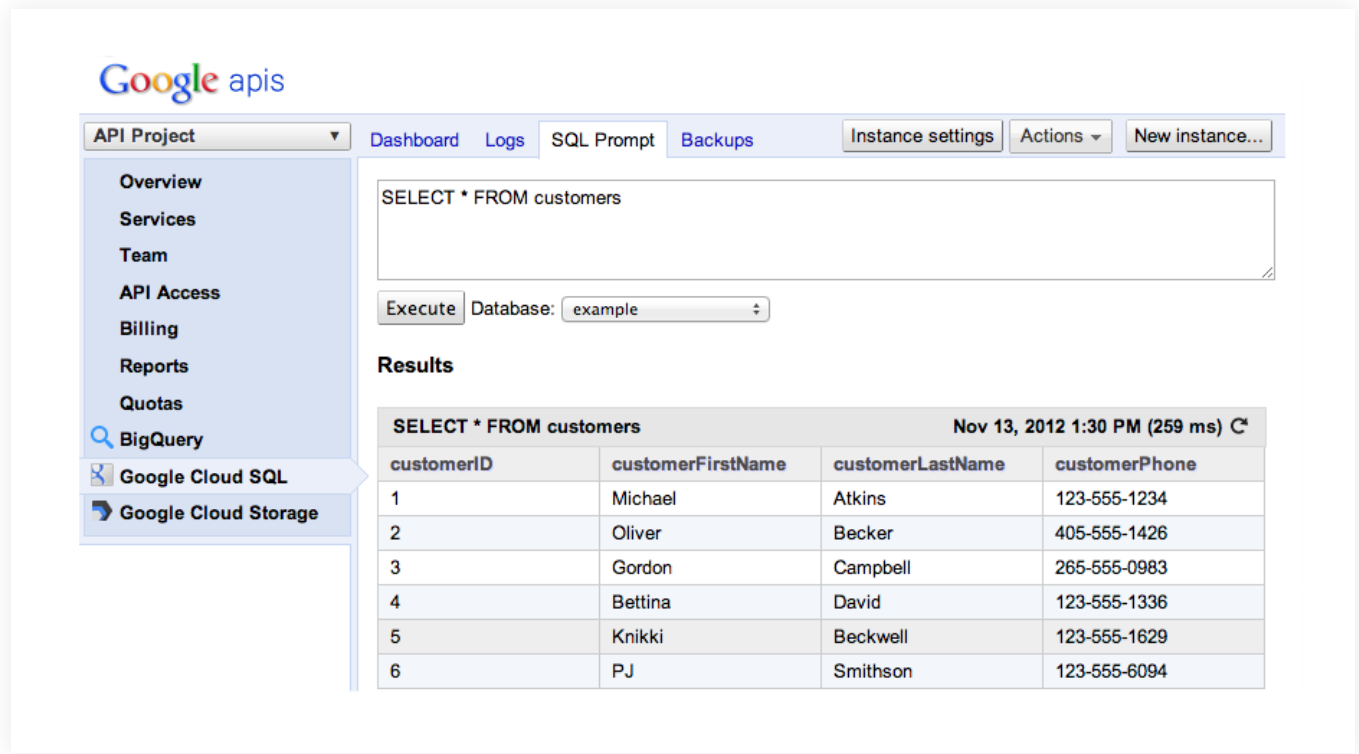
Google Cloud SQL is primarily intended for programmatic use within applications. It has an interactive UI, which is helpful for learning about the product, getting started using it, investigating the schema, and submitting trial queries.

MySQL is a full relational database system that supports full SQL syntax and table management tools. Google Cloud SQL supports a subset of MySQL, which includes most of the features of MySQL. For a list of differences, see the [Google Cloud SQL FAQ](#).

Typical Use Cases

Google Cloud SQL is good for small or medium data sets that:

- must be kept consistent
- are updated frequently
- are queried frequently in many different ways



Google Cloud SQL is typically used to manage, rather than analyze, data because it supports update, append, and delete queries. In database terms, Google Cloud SQL is an [OLTP](#) (online transactional processing) system.

As of Feb 2014, the limit for databases in Google Cloud SQL is 500GB, but check the [Google Cloud SQL Pricing documentation](#) for the latest information.

Typical uses include keeping track of user orders, product catalogs, discussion boards and blogs, content management systems, and workflow applications.

Note: For case studies, see cloud.google.com/products/cloud-sql.

Cloud SQL Data Creation and Ingestion

Google Cloud SQL lets you import existing databases or create them from scratch. You can perform the usual SQL commands to create and drop database tables, and to create, update, and delete rows and data as follows:

- interactively from the online SQL prompt
- from the command line with the `google_sql` tool
- from App Engine applications
- programmatically from other applications using JDBC
- from Apps Script scripts
- using third-party tools, such as the Squirrel SQL Client

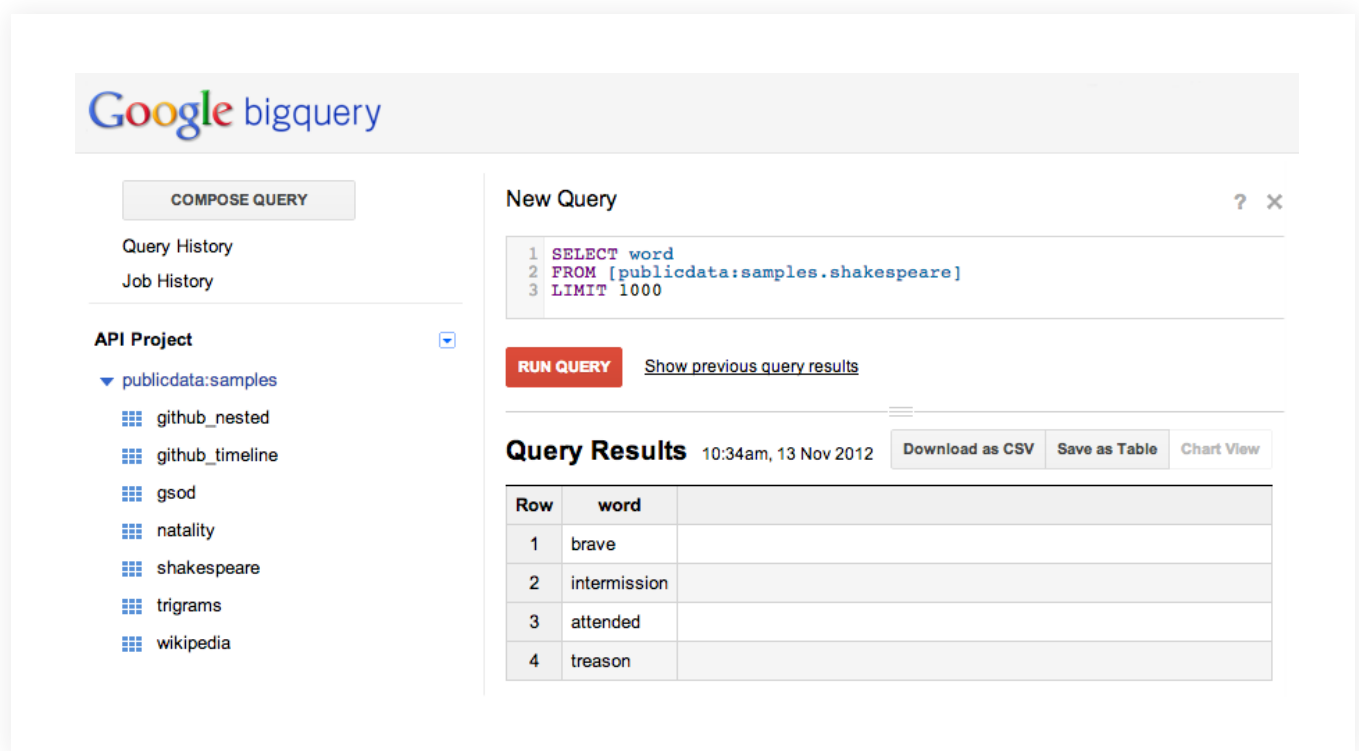
Importing and Exporting Data

To import databases from other MySQL databases, copy the data (for example, as a mysqldump.data file) to Google Cloud Storage, and import it from there into Google Cloud SQL.

To export your data, use the Export option in the Google Cloud console to export your data to Google Cloud Storage.

BigQuery

Google BigQuery Service is a massively parallel query datastore that allows you to run SQL-like queries against very large datasets, with potentially billions of rows, in a matter of seconds. It is primarily intended for programmatic use within applications. It provides an interactive UI, which is helpful for learning about the product and running interactive queries.



The screenshot displays the Google BigQuery web interface. On the left, there is a sidebar with a 'COMPOSE QUERY' button, 'Query History', and 'Job History' links. Below these is the 'API Project' section, which is expanded to show a list of datasets under 'publicdata:samples', including 'github_nested', 'github_timeline', 'gsod', 'natality', 'shakespeare', 'trigrams', and 'wikipedia'. The main area is titled 'New Query' and contains a text editor with the following SQL query:

```
1 SELECT word
2 FROM [publicdata:samples.shakespeare]
3 LIMIT 1000
```

Below the query editor are two buttons: a red 'RUN QUERY' button and a link for 'Show previous query results'. Underneath, the 'Query Results' section shows the execution time as '10:34am, 13 Nov 2012' and three action buttons: 'Download as CSV', 'Save as Table', and 'Chart View'. The results are presented in a table with the following data:

Row	word
1	brave
2	intermission
3	attended
4	treason

BigQuery is based on one of Google's core technologies, and has been used internally by Google for various analytical tasks since 2006.

BigQuery supports analysis of datasets up to hundreds of terabytes.

To use BigQuery, you upload your data into BigQuery and then you can query it interactively or programmatically. You can also query publicly available datasets as well as datasets that other people have shared with you.

You can use BigQuery in the following ways:

- interactively through the [BigQuery browser tool](#)
- using the [bq command-line tool](#)

- programmatically by making calls to the REST API using various client libraries in multiple languages, such as Java and Python

Typical Use Cases

BigQuery is ideal for running queries over vast amounts of data —up to billions of rows— in seconds. It is good for analyzing vast quantities of data quickly, but not for modifying it. In data analysis terms, BigQuery is an [OLAP](#) (online analytical processing) system, and works best for interactive analysis of very large datasets, typically using a small number of very large, append-only tables.

One specific example use of BigQuery is by RedBus, an online travel agency which introduced Internet bus ticketing to India in 2006. Using BigQuery, they analyzed customer travel activity to identify which routes needed more buses, where new bus routes were needed, and whether reduced bookings on specific routes were caused by server problems or simply by less demand. According to Pradeep Kumar, the author of their [technical case study](#), “We had a table which contained 2TB of data but still returned query results in under 30 seconds for most queries.”

See more case studies at [BigQuery Case Studies](#).

BigQuery Data Creation and Ingestion

BigQuery can import data in the following formats:

- CSV is a simple, relatively compact format for flat data structures
- JSON is a more verbose format that’s great for representing nested and repeated data, and easily parseable by both humans and code.
- App Engine Datastore backups.

It is possible to ingest up to 500 source files in a single batch, with a maximum of 1TB of total data per load job (as of Jan 2013).

BigQuery can import data from:

- Google Cloud Storage
- Local files
- Excel
- Third party systems using third party tools (such as Informatica, Knime or Pervasive)

BigQuery supports the following ways to import data:

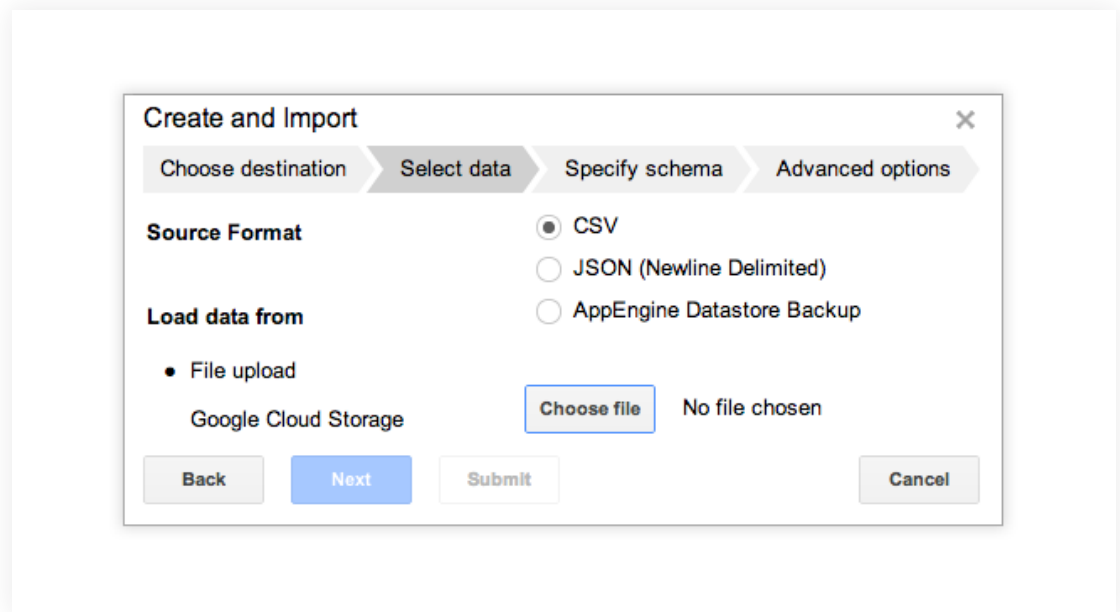
- Interactively in the BigQuery UI
- Using the bq command line tool
- Programmatically using the REST API
- Using the Excel connector
- Using third party tools, discussed [later](#) in this document.

For hints and tips on ingesting data into BigQuery, see [BigQuery Data Ingestion and Best Practices Cookbook](#).

Figure 5.
Example of using filter
to eliminate data that
caused uneven distrib-
uted data

Using the BigQuery UI to Upload Data

You can use the online UI to interactively upload data either from local files or from Google Cloud Storage. Here's a screenshot of the dialog box for interactive upload:



Exporting Query Results from BigQuery

You can export query results as a CSV file to your local computer, or as a permanent table to your dataset. You can also export an entire table to Google Cloud Storage from the BigQuery UI.

Should I Use Google Cloud Storage or Upload Directly to BigQuery?

There are two ways to load source data into Google BigQuery: import from a local file or import files from a Google Cloud Storage bucket. Loading data from a local file can be convenient if you have a single source file that is relatively small.

In general, we suggest using Google Cloud Storage to stage source data files for BigQuery. The major advantage of using a Google Cloud Storage bucket to stage BigQuery source files is that they can be easily ingested in batches. Another advantage to ingesting data from a Google Cloud Storage bucket is that the source files can be kept there as an archive. This can be convenient in case the data needs to be re-ingested.

What Third Party Tools Can I Use To Visualize and Ingest BigQuery Data?

A variety of third parties provide tools for importing and visualizing data from BigQuery, and for loading data into BigQuery.

Visualization and business intelligence tools include:

- [QlikView](#) provides a custom connector which pulls data into their in-memory data model for analysis.
- [Bime Analytics](#) provides a connector which enables live querying from BigQuery with interactive data analysis dashboards.
- [Jaspersoft](#) has integrated its business intelligence product suite with BigQuery.

- [Metric Insights](#) provides a BigQuery plugin for their BI platform.
- [Tableau](#) — Tableau provides direct connection to BigQuery for visualizing data.

ETL tools for loading data into BigQuery include:

- [Informatica](#) has a cloud connector for Google BigQuery and Google Cloud Storage.
- [Pervasive](#) provides a solution for their RushAnalyzer product which enables BigQuery to be used as an output writer for any workflow.
- [Talend](#) has added support for BigQuery in their Open Studio for Big Data, an open source set of code generation tools to design and implement data integration.
- [SQLstream](#) provides a Continuous ETL connector for BigQuery. When data comes into SQLstream's s-Server matching a specific pattern, it is queued for inserting into BigQuery on a regular basis.

See [BigQuery Third Party Tools](#) for more information.

Should I use BigQuery or Google Cloud SQL?

The use cases for BigQuery and Google Cloud SQL are very different. BigQuery lets you import huge amounts of data and then very quickly analyze it. Google Cloud SQL lets you host a MySQL relational database in Google's cloud, and create and update your tables and rows directly.

OLAP versus OLTP

BigQuery is an online analytical processing (OLAP) tool, meant for periodic or batch processing. You wouldn't use BigQuery, for example, to analyze web traffic data in realtime every time a customer clicks on a link in your website, but you would use it to analyze customer click data every hour or every day.

Google Cloud SQL is an online transaction processing (OLTP) tool, and supports high volume, low latency queries in real time.

Size of Data

BigQuery can process terabytes of data very quickly; Google Cloud SQL has a limited of a hundred gigabytes per database (as of Nov 2012).

Data Modification

BigQuery is append-only, so you can't update or change data that is already in a BigQuery table, whereas with Google Cloud SQL, you have full control over changing the data in the tables.

For a further comparison, see BigQuery versus Google Cloud SQL in the [Overview](#) of BigQuery.

App Engine Datastore

App Engine is a Platform as a Service (PaaS) service. It provides an SDK and a set of tools for developing applications in Java and Python. App Engine is a framework for rapidly developing highly scalable web applications that run on Google's infrastructure. As the traffic to your app increases, more instances of your app are automatically created to handle the load, allowing your application to scale up to handling millions of users.

App Engine provides a full SDK to help you develop your application.

There is much to say about App Engine, but this article focuses on how to store data, and whether to use the inbuilt datastore or whether it is better for your App Engine application to use Google Cloud SQL or Google Cloud Storage to store its data. To learn more about App Engine, see [Google App Engine](#).

What is the Datastore?

The App Engine Datastore is a storage system for data used in, or generated by, App Engine applications. To store and query data in the datastore, you write code in any of the languages supported by App Engine.

The Datastore is a NoSQL (non-relational) key/value store that supports unlimited scaling. You can store any key/value pairs you want; entities stored in the Datastore do not need to conform to the same structure.

The Datastore stores the data on Google's infrastructure, in much the same way that Google stores its own data, such as the key-value pairs that represent indexed values for all the web pages that Google crawls.

You can also use the Memcache service to keep data values in the cache to reduce hits to the datastore

Querying the Datastore

App Engine's Datastore allows you to save data as key value pairs (known as entities). The entity properties are indexed and you can query them. App Engine provides an API for querying the datastore, both in [Java](#) and in [Python](#). Although the Query API does not provide the complex query functionality of SQL, it is optimized for the schemaless format of the datastore entries, and allows you to set filters and criteria for searching for entities. There is also a SQL-like querying language called GQL that is syntactically identical to SQL (but only implements a subset of the functionality)

The Admin Console

The App Engine admin console provides views into the saved data and lets you run GQL queries directly in the Datastore Viewer, which can be very useful for admins to interrogate and trouble-shoot stored data.

Should I use Google Cloud Storage or the App Engine Datastore?

Google Cloud Storage is intended to host write-once data, whereas the datastore is for dynamic application data. For frequently changing key value pairs, use the Datastore. For large chunks of data that will not change frequently (such as documents) and large binary files (blobs), use Google Cloud Storage.

Your App Engine application can use Google Cloud Storage as a conduit for sharing data with consumers outside the scope of the application. If your data already exists as files outside of App Engine, the logical solution is to move it to Google Cloud Storage and pull it into your application. If your App Engine application generates large chunks of data that need to be accessible outside the application, store

the data in Google Cloud Storage.

Using Google Cloud Storage with App Engine

To access Google Cloud Storage, App Engine applications can either use the REST API provided by Google Cloud Storage, or use a custom App Engine Files API.

App Engine's [custom API](#) for storing and serving data from the Google Cloud Storage service is more direct and efficient than the RESTful HTTP interface. [[Java API](#), [Python API](#)]

Should I use Google Cloud SQL or the App Engine Datastore?

The App Engine Datastore and Google Cloud SQL can be used to store the same kind of application data in general, but the datastore uses schemaless, NoSQL data while Google Cloud SQL stores data in MySQL tables.

The App Engine Datastore provides NoSQL key-value storage that is highly scalable. Google Cloud SQL supports complex queries and ACID transactions, but this means the database acts as a 'fixed pipe' and performance is less scalable. The decision likely comes down to whether you are more comfortable with traditional SQL with tightly-managed schema, or NoSQL where there are no requirements of conformity across objects of the same kind. Many applications use both types of storage.

Google Compute Engine

Google Compute Engine is an Infrastructure as a Service (IaaS) offering that lets you run your large-scale computing workloads on Linux virtual machines hosted on Google's infrastructure. The development of the applications is up to you — develop and run whatever services and applications you want. Your applications can use the API and command line tools provided by Google Drive, Google Cloud Storage, Google Cloud SQL and BigQuery to integrate with data hosted across the Google Cloud Platform.

Applications running on Google Compute Engine can store their application data using one of the following::

- Persistent disk — A replicated, network-connected storage service that is comparable to the latency and performance of local disks. Data written to this device is replicated to multiple physical disks in a Google data center. You can also create snapshots of your disks for backup/restore purposes, and can mount these devices in a mode that allows multiple virtual machines to read from a single device.
- Google Cloud Storage — Easily access your Google Cloud Storage data buckets from inside a virtual machine. The seamless authentication makes it easy to securely access your data without having to manage keys in your virtual machines.

How do I choose between App Engine and Google Compute Engine?

App Engine is a Platform as a Service (PaaS) service. It provides an SDK and a set of tools for developing applications in Java and Python. It provides plugins for easy integration with the Eclipse development environment, and comes with templates for running a "Hello World" application right out of the box. The SDK provides boilerplate code for creating web applications, and allows you to deploy your applications with just a click. You still have to write the logic of the application, but once you've deployed your application, Google keeps it running for you and replicates the data. You can use App Engine to get started quickly developing and deploying a web application, but you are constrained to the supported languages and the requirements of the framework.

Google Compute Engine, on the other hand, is an Infrastructure as a Service (IaaS) offering. Google provides the infrastructure for you to run your applications, but you have to design them, develop them,

run them and manage them on your own.

To deploy a simple “Hello World” application using Google App Engine takes a few clicks (in Java) or a few lines of code (Python) to build and deploy the application. From there, you can add files to extend it. ([Java Getting Started codelab](#), [Python Helloworld Exercise](#))

To deploy a simple “Hello World” application on Google Compute Engine, you need to setup a firewall, start the virtual machine instance, login into to the instance, install and configure a web server such as Apache, and create the web page that displays “Hello World.” ([Google Compute Engine Hello World exercise](#))

Summary

The decision as to which of the Google’s Cloud Platform services to use depends on your business and data needs, and whether you’re starting from scratch building web applications, you have existing relational databases, you’re primarily looking for a cloud-hosted content repository, you need to analyze massive datasets, or you want to create your application entirely yourself but run them on Google’s infrastructure.

You can use Google Cloud Storage as a content repository in its own right, for all kinds of files of any size, and you can also use it as a staging area for files to be shared not only by other Google Cloud Platform services but by any external applications. You can use Google Cloud SQL to run your MySQL databases in the cloud, leaving Google to host it, manage it and keep it running. BigQuery lets you analyze massive datasets, and has hooks into third party tools for visualization and ETL. App Engine hosts and runs your web application using Google’s infrastructure, allowing you to store data in the application’s datastore or save it out to Google Cloud SQL and Google Cloud Storage. And Google Compute Engine provides the ultimate freedom in developing and designing applications, and lets you run your programs on Google’s cloud almost as if our data centers were your own private servers.

Read More

This article presented a high-level overview of the offerings in the Google Cloud Platform for storing and analyzing your data. For more information, peruse the following resources at your leisure:

Entry Points

- [Google Cloud home page](#)
- [Google Cloud portal for developers](#)
- [BigQuery Service](#)
- [Cloud Console](#) (for accessing App Engine, Cloud Storage, and Cloud SQL)

Case studies

- [App Engine](#)
- [Google Cloud Storage](#)
- [Google Cloud SQL](#)

- [BigQuery](#)

Developer Documentation and Getting Started Guides

- [App Engine](#) – [\[Get Started\]](#)
- [Google Cloud Storage](#) – [\[Get Started\]](#)
- [Google Cloud SQL](#) – [\[Get Started\]](#)
- [BigQuery](#) – [\[Get Started\]](#)
- [Google Compute Engine](#) – [\[Get Started\]](#)

Articles

- [Extracting and Transforming App Engine Datastore Data](#) (This is a Datastore to BigQuery codelab)
- [BigQuery Data Ingestion and Best Practices Cookbook](#)
- [BigQuery Third Party Tools](#)
- [Insiders guide to BigQuery](#)
- [Best Practices for Writing Scalable App Engine Applications](#)
- [What MySQL syntax does Google Cloud SQL support?](#)

About our Data Centers

- [Inside our Data Centers](#)
- [How we Protect Your Data](#)