BigQuery architecture

How does a Data Warehouse drive business decisions?

A data warehouse consolidates data from disparate sources and performs analytics on the aggregated data to add value into the business operations by providing insights. Data warehouses are the custodians of the most important business data in the enterprise for the last two decades.

Today, enterprises need to:

1. **Have a 360⁰ view of their businesses:** Data is valuable. As cost of storage and data processing reduce, enterprises want to process, store, and analyze all relevant datasets, both internal and external to their organization.
2. **Be situationally aware of and responsive to real-time business events:** Enterprises need to gain insights from real time events and not wait for days or weeks to analyze data. The data warehouse needs to reflect the present state of business at all times
3. **Reduce time to insights:**Enterprises need to get up and running fast without waiting days or months. for hardware or software to be installed or configured.
4. **Make insights available to business users to enable data driven decision making across the enterprise:** In order to embrace a data-driven culture, enterprises need to democratize access to data.
5. **Secure their data and govern its use:** Data needs to be secure and accessible to the right stakeholders inside and outside of the enterprise.

BigQuery—Cloud Data Warehouse

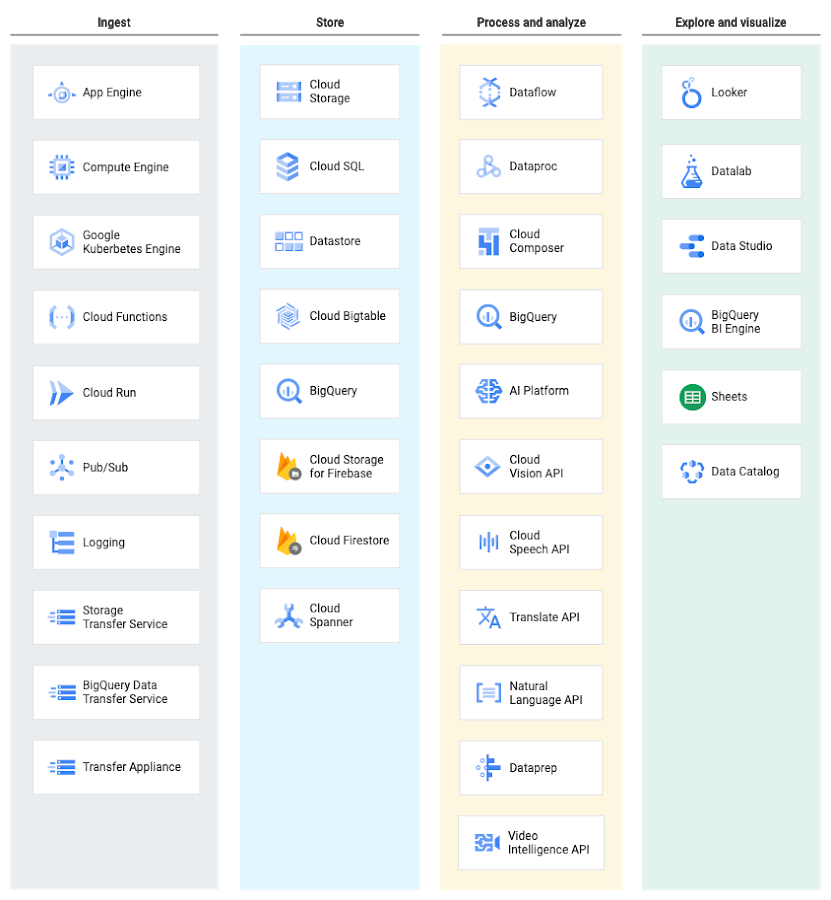
Google BigQuery was designed as a “cloud-native" data warehouse. It was built to address the needs of data driven organizations in a cloud first world.

BigQuery is GCP’s serverless, highly scalable, and cost effective cloud data warehouse. It allows for super-fast queries at petabyte scale using the processing power of Google’s infrastructure. Because there’s no infrastructure for customers to manage, they can focus on uncovering meaningful insights using familiar SQL without the need for a database administrator. It’s also economical because they pay only for the processing and storage they use.

Where does BigQuery fit in the data lifecycle?

BigQuery is part of Google Cloud’s comprehensive data analytics platform that covers the entire analytics value chain including ingesting, processing, and storing data, followed by advanced analytics and collaboration. BigQuery is deeply integrated with GCP analytical and data processing offerings, allowing customers to set up an enterprise ready cloud-native data warehouse.

At each stage of the data lifecycle, GCP provides multiple services to manage data. This means customers can select a set of services tailored to their data and workflow.

BigQuery in Data Lifecycle

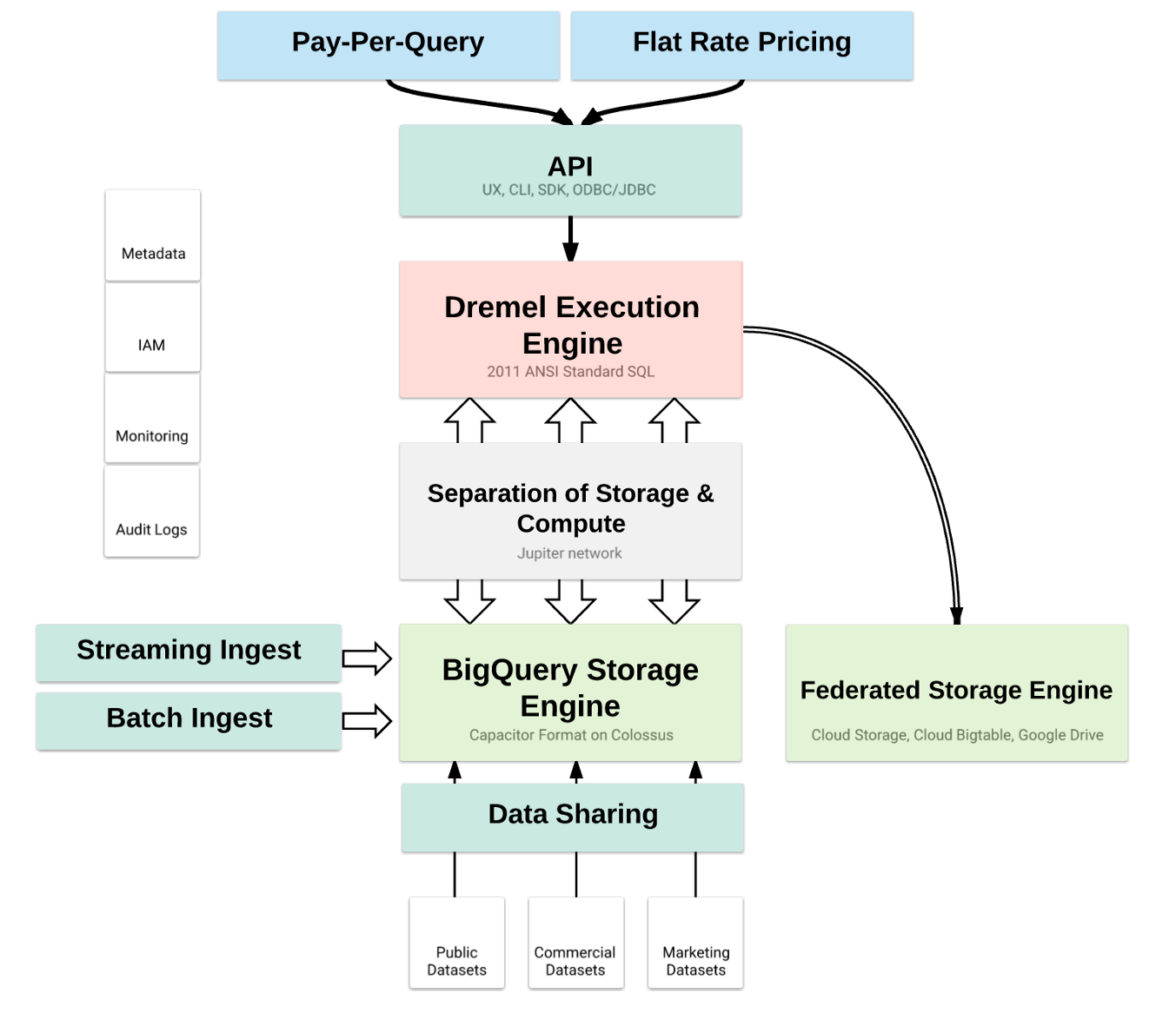
Ingesting data into BigQuery

BigQuery supports several ways to ingest data into its managed storage. The specific ingestion method depends on the origin of the data.

For example, some data sources in GCP, like Cloud [Logging](https://cloud.google.com/logging/docs/export/configure_export_v2) and [Google Analytics](https://support.google.com/analytics/answer/3416092), support direct exports to BigQuery.

[BigQuery Data Transfer Service](https://cloud.google.com/bigquery-transfer/docs/transfer-service-overview) enables data transfer to BigQuery from Google SaaS apps (Google Ads, Cloud Storage), Amazon S3, and other data warehouses (Teradata, Redshift).

Streaming data, such as logs or IoT device data, can be written to BigQuery using [Cloud Dataflow](https://cloud.google.com/dataflow/docs/guides/templates/provided-streaming#cloudpubsubtobigquery) pipelines, [Cloud Dataproc](https://cloud.google.com/solutions/using-apache-spark-dstreams-with-dataproc-and-pubsub) jobs, or directly using the [BigQuery stream ingestion API](https://cloud.google.com/bigquery/streaming-data-into-bigquery).

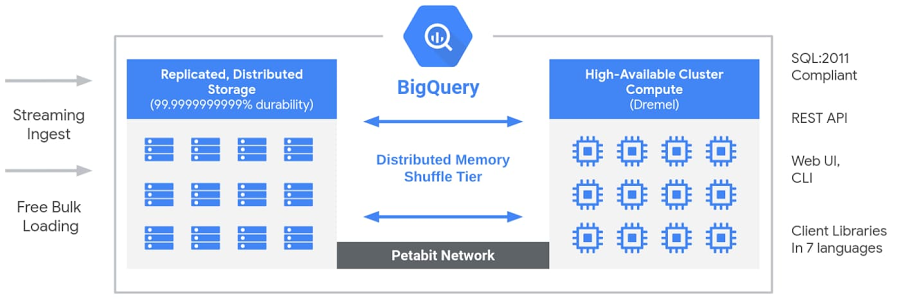


BigQuery Architecture

BigQuery’s serverless architecture decouples storage and compute and allows them to scale independently on demand.

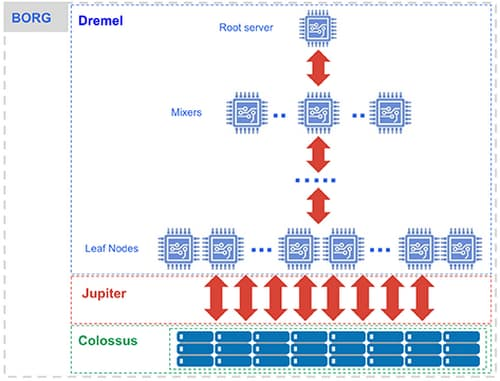
This structure offers both immense flexibility and cost controls for customers because they don’t need to keep their expensive compute resources up and running all the time.

This approach also allows customers of any size to bring their data into the data warehouse and start analyzing their data using Standard SQL without worrying about database operations and system engineering.



BigQuery Architecture

Under the hood, BigQuery employs a vast set of multi-tenant services driven by low-level Google infrastructure technologies like [Dremel, Colossus, Jupiter and Borg](https://cloud.google.com/blog/big-data/2016/01/bigquery-under-the-hood).



BigQuery: Under the hood [[source](https://cloud.google.com/blog/big-data/2016/01/bigquery-under-the-hood)]

**Compute is**[Dremel](https://research.google.com/pubs/pub36632.html)**, a large multi-tenant cluster that executes SQL queries.**

* Dremel turns SQL queries into execution trees. The leaves of the tree are called slots and do the heavy lifting of reading data from storage and any necessary computation.

The branches of the tree are ‘mixers’, which perform the aggregation.

* Dremel dynamically apportions slots to queries on an as-needed basis, maintaining fairness for concurrent queries from multiple users. A single user can get thousands of slots to run their queries.

**Storage is**[Colossus](https://www.systutorials.com/3202/colossus-successor-to-google-file-system-gfs/)**, Google’s global storage system.**

* BigQuery leverages the [columnar storage format](https://cloud.google.com/blog/products/gcp/inside-capacitor-bigquerys-next-generation-columnar-storage-format) and compression algorithm to store data in Colossus, optimized for reading large amounts of structured data.
* Colossus also handles replication, recovery (when disks crash) and distributed management (so there is no single point of failure). Colossus allows BigQuery users to scale to dozens of petabytes of data stored seamlessly, without paying the penalty of attaching much more expensive compute resources as in traditional data warehouses.

**Compute and storage talk to each other through the petabit**[Jupiter](https://cloudplatform.googleblog.com/2015/06/A-Look-Inside-Googles-Data-Center-Networks.html)**network.**

* In between storage and compute is ‘shuffle’, which takes advantage of Google’s Jupiter network to move data extremely rapidly from one place to another.

**BigQuery is orchestrated via**[Borg](https://research.google.com/pubs/pub43438.html)**, Google’s precursor to**[Kubernetes](https://kubernetes.io/)**.**

* The mixers and slots are all run by Borg, which allocates hardware resources.

BigQuery users get the benefit of continuous improvements in performance, durability, efficiency, and scalability, without downtime and upgrades associated with traditional technologies.

How can you get started with BigQuery?

You can start using BigQuery simply by loading data and running SQL commands. There’s no need to build, deploy, or provision clusters; no need to size VMs, storage, or hardware resources; no need to setup disks, define replication, configure compression and encryption, or any other setup or configuration work necessary to build a traditional data warehouse..

You can access BigQuery in multiple ways:

* Using the [GCP console](https://cloud.google.com/bigquery/docs/quickstarts/quickstart-web-ui)
* Using the [command line tool](https://cloud.google.com/bigquery/docs/quickstarts/quickstart-command-line) bq
* Making calls to the [BigQuery REST API](https://cloud.google.com/bigquery/docs/reference/rest/)
* Using the variety of [client libraries](https://cloud.google.com/bigquery/docs/quickstarts/quickstart-client-libraries) such as Java, .NET or Python

Let’s try it out now. Navigate to [BigQuery web UI](https://console.cloud.google.com/bigquery) on Google Cloud Console, copy and paste the following query, and then hit the “Run” button.

Language: SQL

SELECT

EXTRACT(YEAR FROM creation\_date) AS year,

EXTRACT(MONTH FROM creation\_date) AS month,

COUNT(creation\_date) AS number\_posts

FROM

`bigquery-public-data.stackoverflow.stackoverflow\_posts`

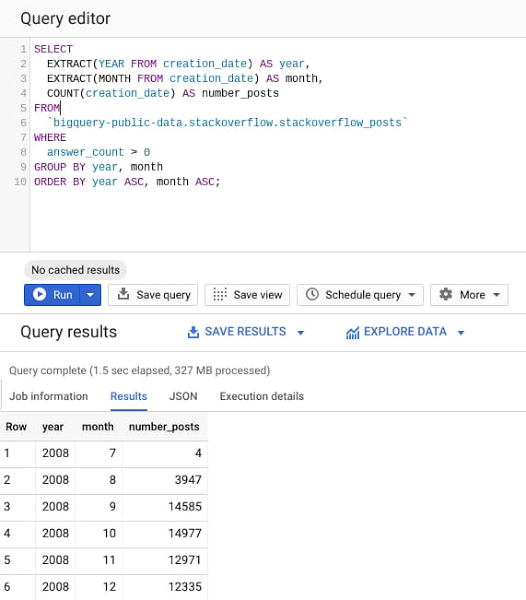
WHERE

answer\_count > 0

GROUP BY year, month

ORDER BY year ASC, month ASC;

The query processes ~30GB of StackOverflow posts available from 2008 to 2016 in [public BigQuery datasets,](https://cloud.google.com/bigquery/public-data/)to find the number of posts with at least one answer posted, grouped by year and month.



As evident from the query results, it takes less than 2s to analyze 28GB of data and return the results. BigQuery engine is smart to read only the columns required to execute the query and process only 327MB of data out of the entire 28GB dataset.

Users are able to seamlessly scale to dozens of petabytes because BigQuery engineers have already deployed the resources required to reach this scale. Therefore, scaling is simply a matter of using BigQuery more, rather than provisioning larger clusters.