BigQuery

BigQuery

- **Fully Managed Data Warehouse –** BigQuery is a serverless, fully managed cloud-based data warehouse designed for analytics at any scale.
- **High Performance –** It can process terabytes in seconds and petabytes in minutes, enabling real-time analysis of massive datasets.
- **SQL & ML Integration –** BigQuery supports standard SQL for queries and integrates with BigQuery ML for machine learning directly within the platform.
- **Separation of Storage & Compute –** Data storage and processing are handled independently, ensuring scalable and cost-efficient operations.
- **Wide Ecosystem Support –** Integrates seamlessly with Google Cloud services, BI tools (Looker, Tableau), and external data sources for analytics.

What is Serverless?

- Serverless computing is a cloud model where the cloud provider manages servers, scaling, and infrastructure automatically.
- Developers focus only on writing code or running queries without worrying about provisioning or maintaining servers.
- Resources scale up or down automatically based on workload, ensuring cost efficiency and high availability.
- Examples: BigQuery, Cloud Functions, and Cloud Run on Google Cloud.

Benefits of BigQuery

- Eliminates infrastructure management through a serverless model.
- Provides real-time streaming and batch ingestion of structured and unstructured data.
- Supports open formats like Apache Iceberg, Delta, and Hudi, ensuring flexibility.

BigQuery Storage

- Data is stored in a columnar format optimized for analytical queries.
- Storage is replicated automatically across regions for durability and availability.
- Supports ACID transactions and multiple data-loading methods including streaming, batch, and Data Transfer Service.

BigQuery Storage

- BigQuery storage is designed to handle very large datasets — from gigabytes to petabytes.
- No need to buy or configure disks, clusters, or servers.
- Storage is automatically managed by BigQuery.
- Charges are based only on the amount of data stored, separate from queries.

Columnar Storage Format

- Data is stored in a columnar format rather than row-based storage.
- Columnar storage accelerates analytical queries by reading only required columns.
- The Capacitor storage engine powers this design for maximum efficiency.

Data Availability

- BigQuery storage provides 99.999999999% durability.
- Data is automatically replicated across multiple availability zones.
- Protection against hardware failures and zone outages ensures availability.

Tables in BigQuery

- Data is organized into tables within datasets.
- Types of tables include:
  - Standard tables / Native tables for primary storage
  - External tables - Data can remain outside BigQuery while still being queried
  - Snapshots for time-based recovery
  - Clones as lightweight, zero-copy references
  - Views
  - Materialized views for precomputed results

BigQuery Table ID: Structuring Resources

- A Table ID is a fully qualified identifier used to uniquely reference a table in BigQuery.
- Structure: **<project_id>.<dataset_id>.<table_id>**
  - project_id → Google Cloud project containing the dataset
  - dataset_id → Dataset within the project
  - table_id → Table name within the dataset
- Note: In BigQuery SQL, backticks (`) are often used to enclose table or column names that contain special characters, spaces, or reserved keywords

Analysing Data using BigQuery

- BigQuery querying is the process of retrieving and analyzing data using SQL in a serverless, fully managed environment.
- It enables rapid analysis of large datasets—from gigabytes to petabytes—without infrastructure setup.

Analysing Data using BigQuery

- Runs ANSI-standard SQL (2011) with joins, nested fields, window functions, and aggregations.
- Enables queries to access data in Cloud Storage, Bigtable, Spanner, and Google Sheets.
- Integrates with BI tools such as Looker Studio, Tableau, Power BI, and Google Sheets.

What is Information Schema?

- A set of read-only views in BigQuery providing metadata about datasets, tables, jobs, and resources.
- Use standard SQL, just like any other BigQuery table.
- Helps monitor, audit, and optimize BigQuery usage.
- Provides governance and compliance insights on data location and access.

Information Schema – Key Metadata Categories

- Dataset Metadata → Dataset size, location, labels.
- Table Metadata → Schema, row counts, partitions, clustering.
- Job Metadata → Query runtime, execution costs, job details.

Examples of Information Schema Views

- **INFORMATION_SCHEMA.SCHEMATA** → Datasets in a project.
- **INFORMATION_SCHEMA.TABLES** → Table details in a dataset.
- **INFORMATION_SCHEMA.JOBS_BY_PROJECT** → Query and job history.
- **INFORMATION_SCHEMA.COLUMNS** → Column-level metadata.

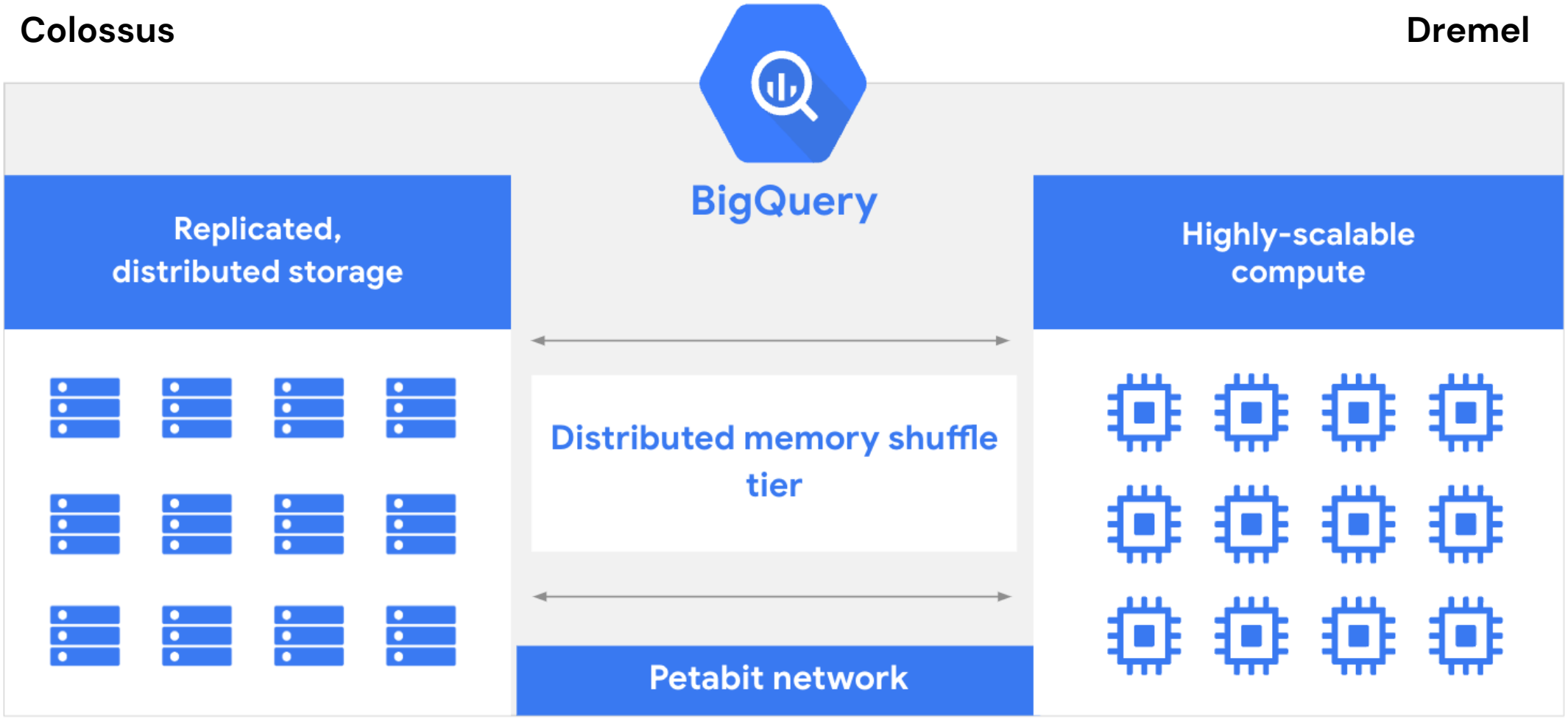
Information Schema Cost

- Every INFORMATION_SCHEMA query is billed with a minimum of **10 MB** of data processed, even if the query scans less than that.
- On capacity-based projects, INFORMATION_SCHEMA queries consume purchased slots.
- INFORMATION_SCHEMA queries are not cached; running the same query again still incurs charges/slot usage.
- No storage fees: INFORMATION_SCHEMA views do not incur storage charges.

BigQuery Architecture

- BigQuery consists of two layers: a Storage Layer for ingestion and persistence, and a Compute Layer for analytics.
- These layers communicate through Google’s high-speed petabit network.
- The separation allows independent scaling and better performance.

BigQuery Architecture



Capacitor — BigQuery's Storage Format

- **Columnar Storage Format:** Capacitor stores data in columns rather than rows, making queries faster for analytics workloads.
- **Compression & Encoding:** Optimized for high compression and encoding techniques to reduce storage cost.
- **Streaming-Friendly:** Supports both batch and streaming data ingestion for real-time analytics.

Colossus — BigQuery's Persistent File System

- **Next-Gen GFS:** Colossus is the successor to the Google File System (GFS) providing global-scale replicated storage.
- **High Durability:** Data is replicated across multiple locations for reliability and fault tolerance.

Jupiter Networking Fabric (Petabit Network)

- **High-Speed Network:** Jupiter is Google's internal data center network operating at petabit-per-second scale.
- **Low Latency:** Enables extremely fast data transfer between storage and compute layers.
- **Scalable Backbone:** Connects thousands of servers to support BigQuery's distributed architecture.

Dremel — Query Execution Engine

- **Tree Architecture:** Dremel uses a multi-level serving tree to break queries into smaller pieces.
- **Parallel Execution:** Workers process query fragments in parallel, combining results at the root node.
- **High Performance:** Executes SQL queries on massive datasets in seconds.

Slots — BigQuery’s Compute Units

- **Definition:** A slot is a unit of computational capacity in BigQuery.
- **Slot Allocation:** Queries use slots dynamically; more complex queries require more slots.
- **Reservation Model:** Slots can be purchased for predictable pricing (flat-rate) or used on-demand.

BORG — Cluster Management System

- **Google’s Internal Scheduler:** BORG manages compute resources across Google’s data centers.
- **Job Scheduling & Scaling:** Allocates CPU, memory, and disk resources to BigQuery jobs efficiently.
- **Fault Tolerance:** Reschedules tasks automatically if nodes fail.

Why Separate Compute and Storage?

- Traditional databases share resources between reads, writes, and queries, causing conflicts.
- BigQuery separates the storage layer (data) from the compute layer (processing).
- Independent scaling of these layers avoids resource conflicts.
- Queries run efficiently without being slowed down by storage operations.

Machine Learning and AI in BigQuery

- BigQuery ML allows you to build and run ML models using simple SQL queries.
- Integrates directly with business intelligence workflows for predictive analytics.
- BigQuery Studio provides Python notebooks and version control to streamline ML workflows.

Governance and Security in BigQuery

- Built-in governance ensures metadata management, semantic search, and lineage tracking.
- IAM-based access control allows fine-grained security and resource permissions.
- Integration with Dataplex Universal Catalog provides centralized visibility and control.