Big Data de Google

Ismael Yuste
Strategic Cloud Engineer
KSchool / Septiembre 2019

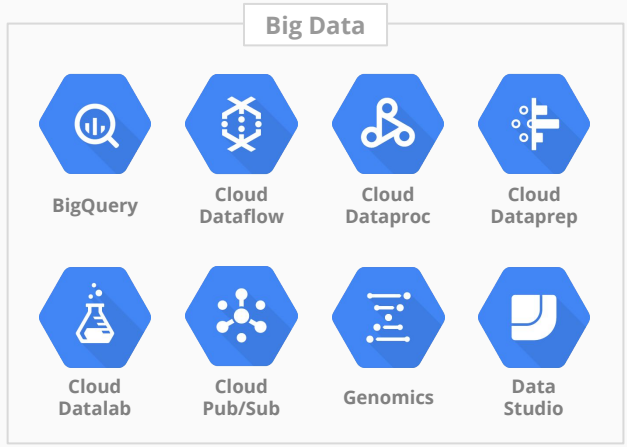
Agenda

- Big Data
- Pub-Sub
- DataProc
- DataFlow
- BigQuery
- DataLab
- Composer
- Data Fusion
- Data Catalog

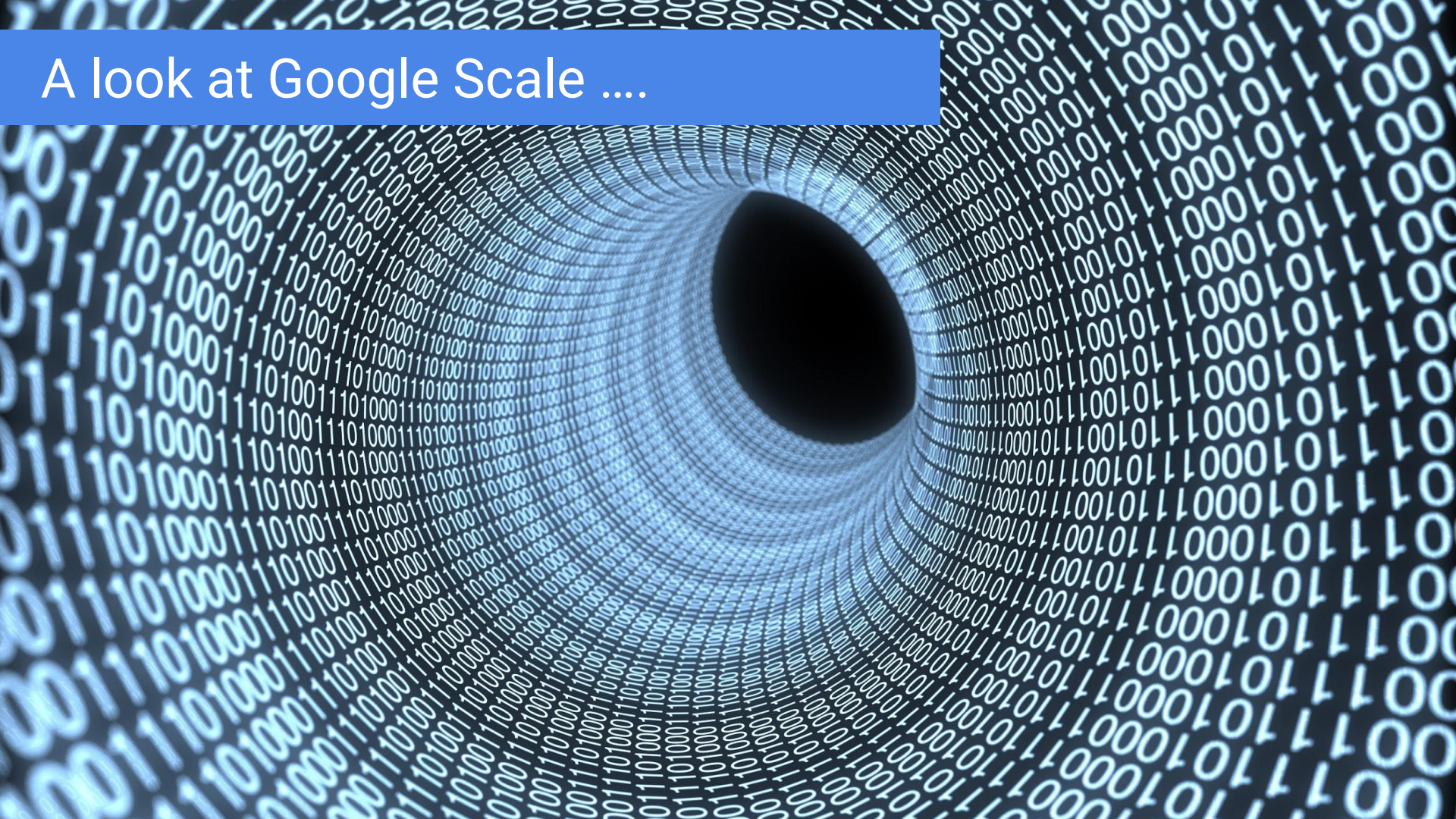
GCP Big Data

Big Data

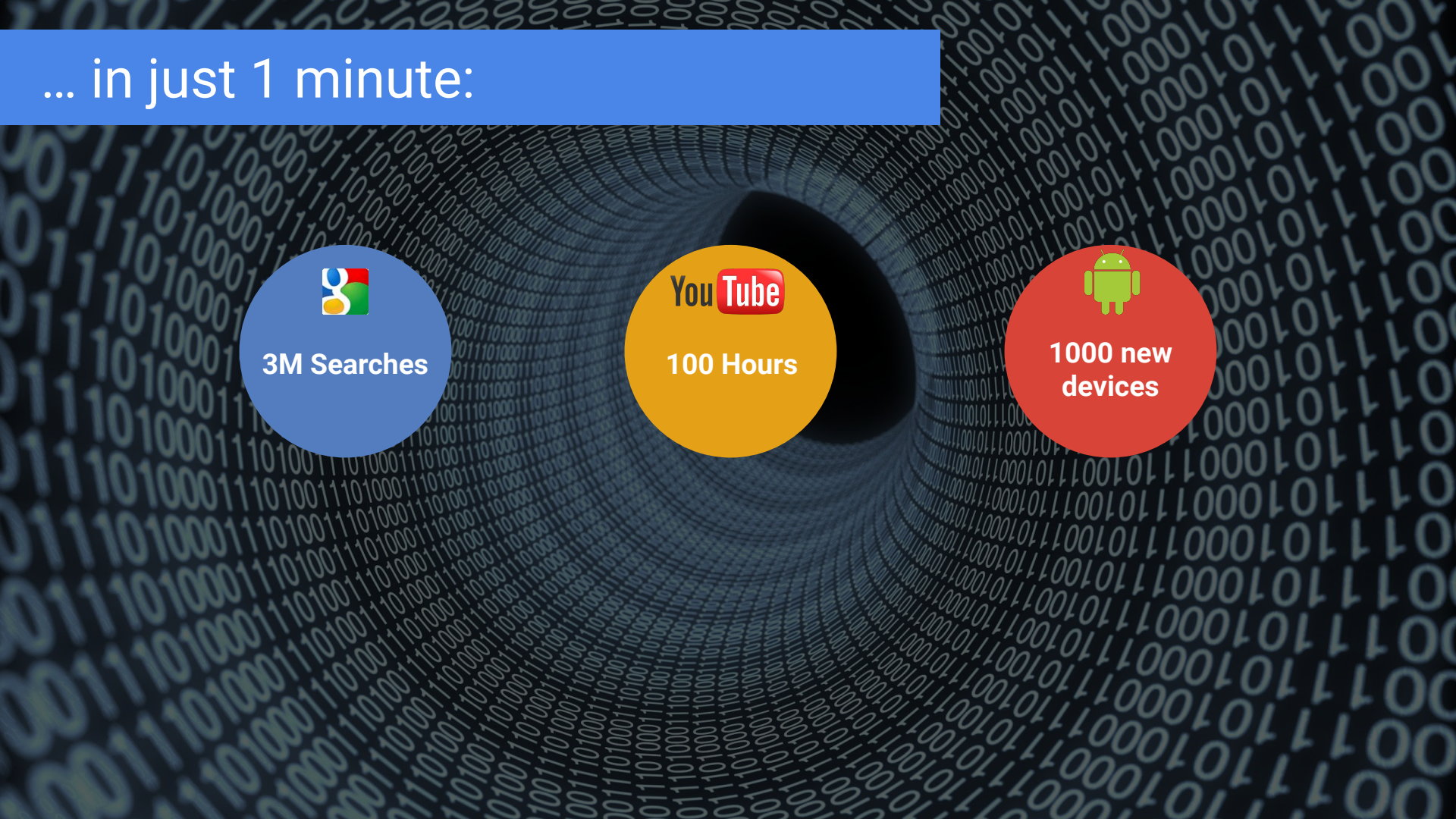
Soluciones de Big Data integradas de principio a fin, que permite capturar los datos, procesarlos y almacenarlos en una plataforma integrada. Combina servicios nativos en la nube y herramientas Open Source gestionadas, tanto en tiempo real como por lotes.



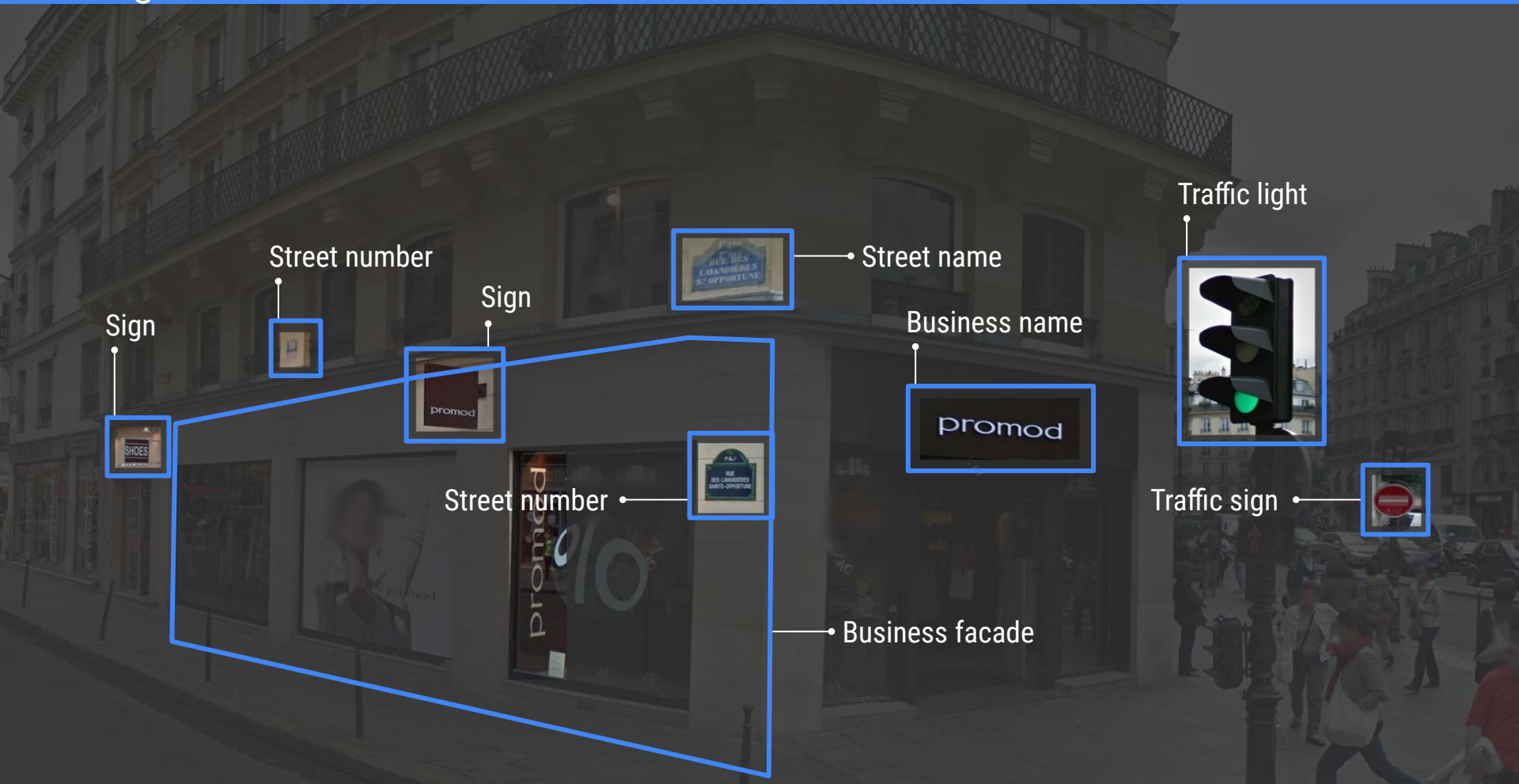
A look at Google Scale ….



… in just 1 minute:



Finding new value in data



Data Lifecycle Steps

Ingest

The first stage is to pull in the raw data, such as streaming data from devices, on-premises batch data, application logs, or mobile-app user events and analytics.

Store

After the data has been retrieved, it needs to be stored in a format that is durable and can be easily accessed.

Process & Analyze

In this stage, the data is transformed from raw form into actionable information.

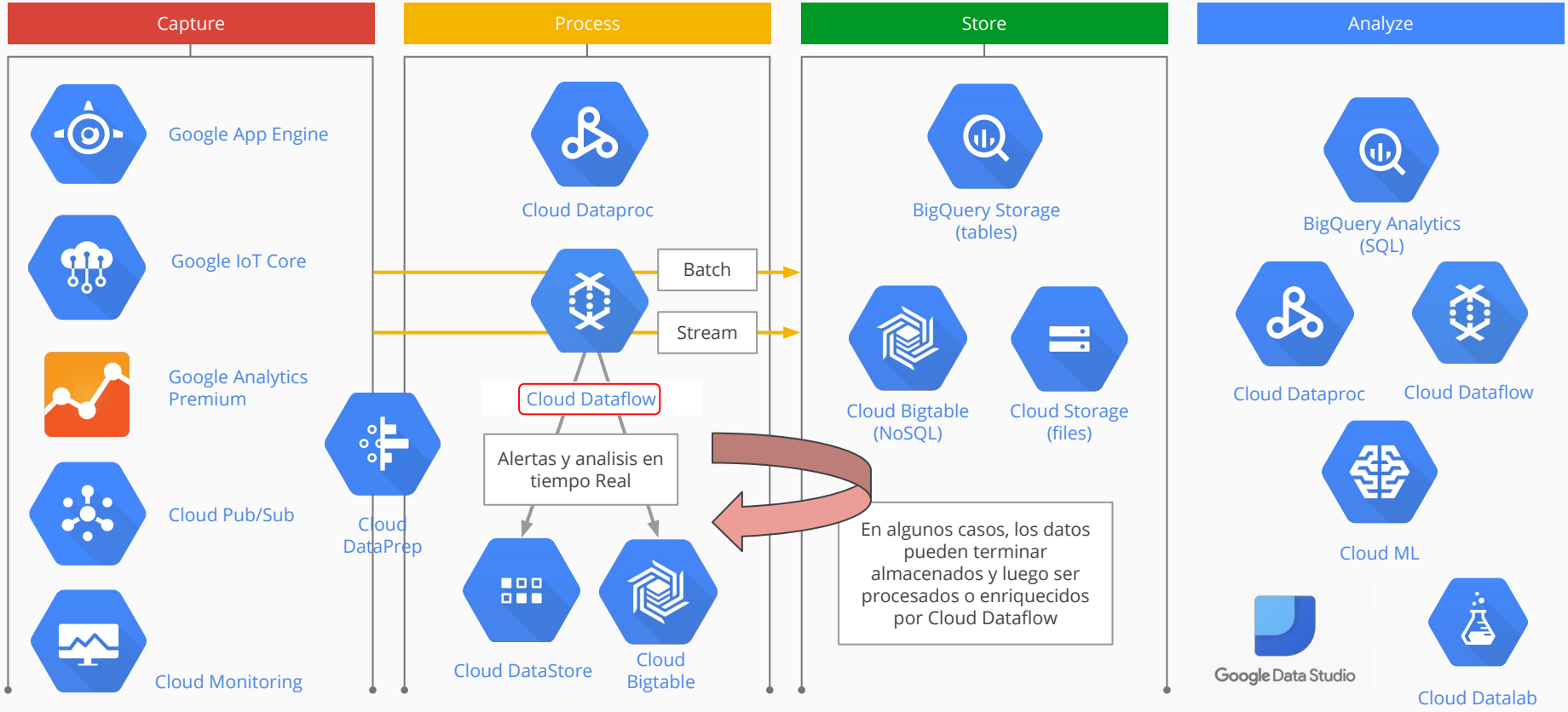
Explore & Visualize

The final stage is to convert the results of the analysis into a format that is easy to draw insights from and to share with colleagues and peers.

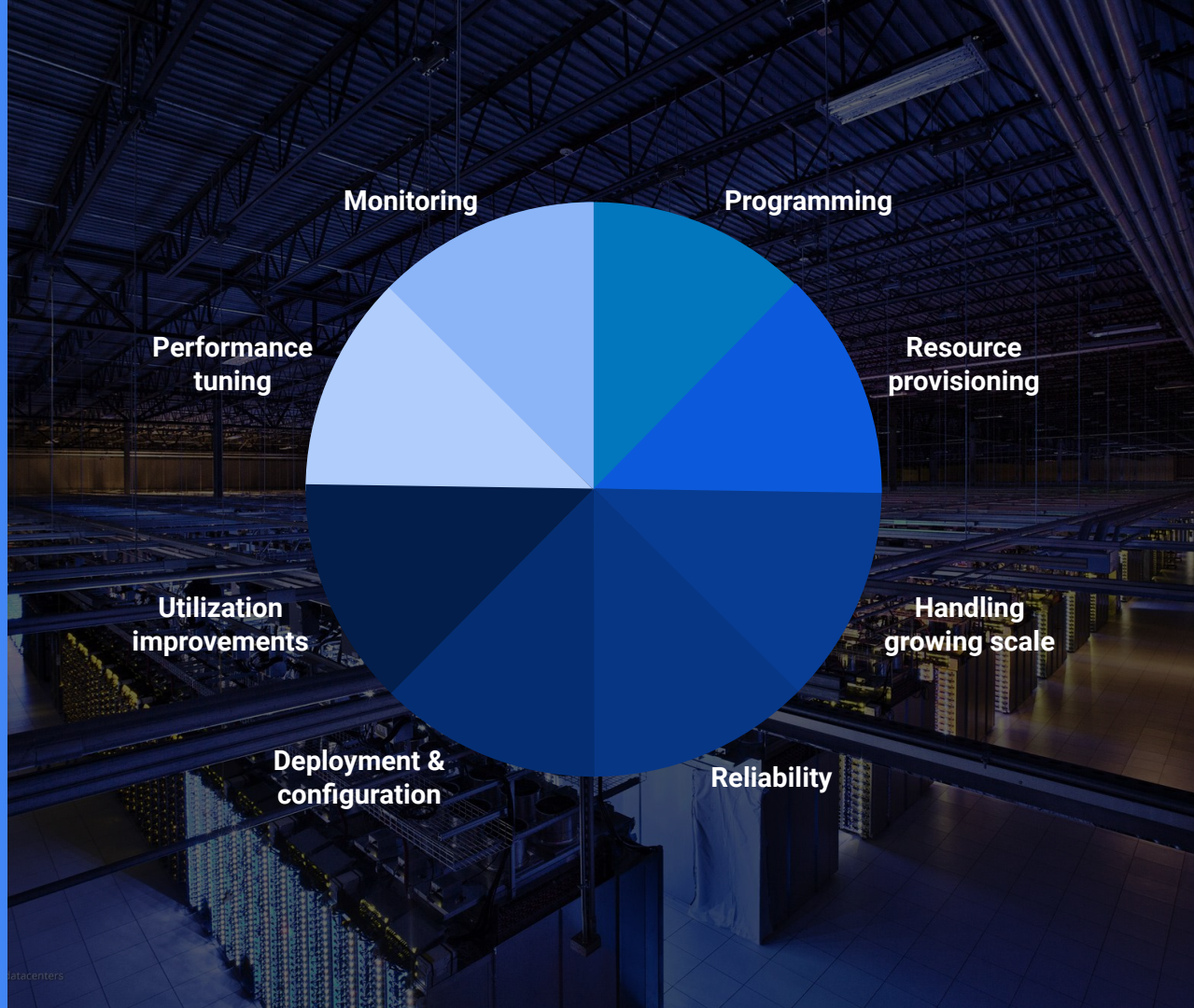
Products to Support Data Lifecycle

Ingesta	Almacenamiento	Procesar y Analizar	Explorar y Visualizar
Cloud Pub/Sub	Cloud Storage	Cloud Dataflow	Cloud Console
Cloud IoT Core	Cloud SQL	Cloud Dataproc	Google Data Studio
	Cloud Datastore	BigQuery	Google Sheets
	Cloud BigTable		Cloud Datalab
	BigQuery		BI/Analytics Partners
	Cloud Spanner		

Data processing tools on GCP

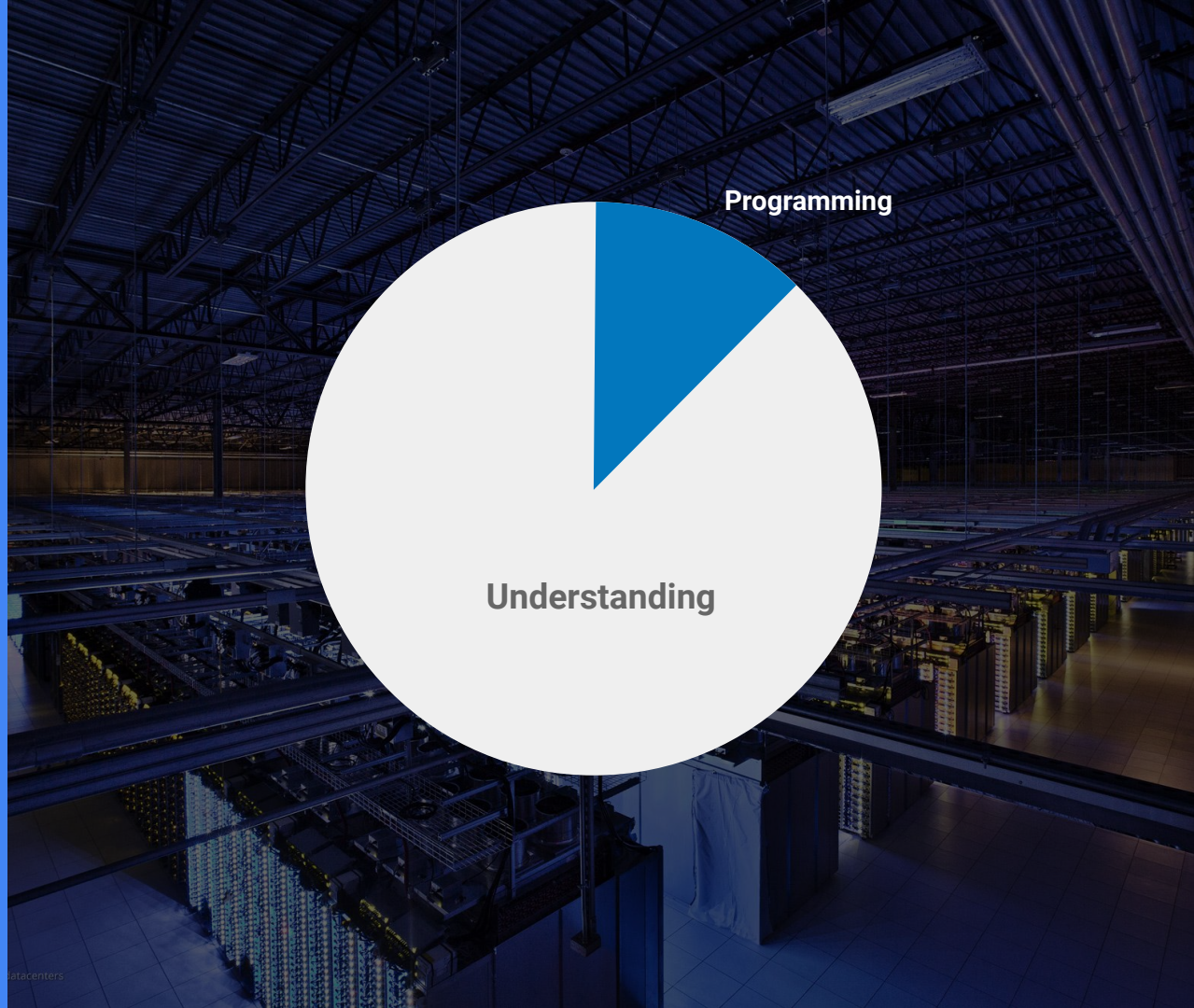


Typical Big Data Jobs



Big Data with Google

Focus on insights.
Not infrastructure.
From batch to real-time.

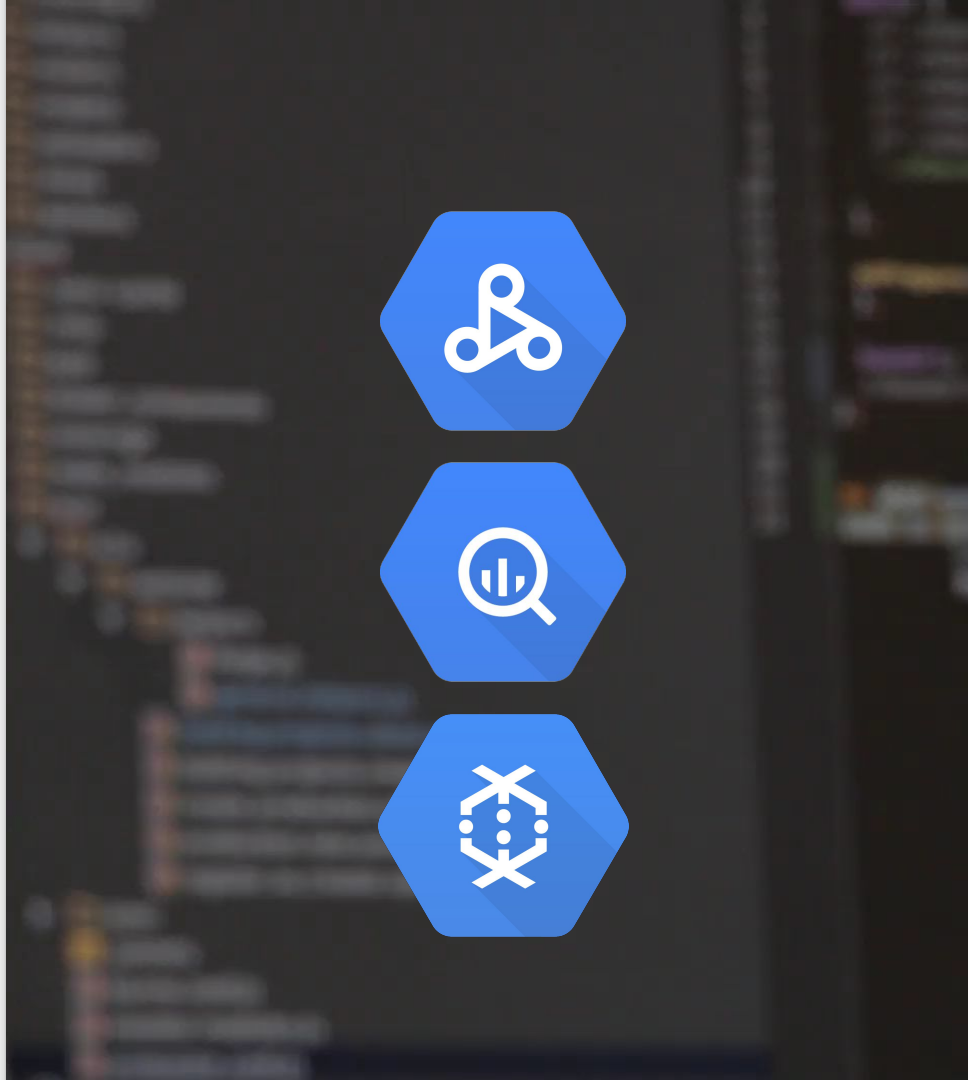


Data & Analytics

Cloud Dataproc
Fully managed Hadoop and Spark with industry-leading performance

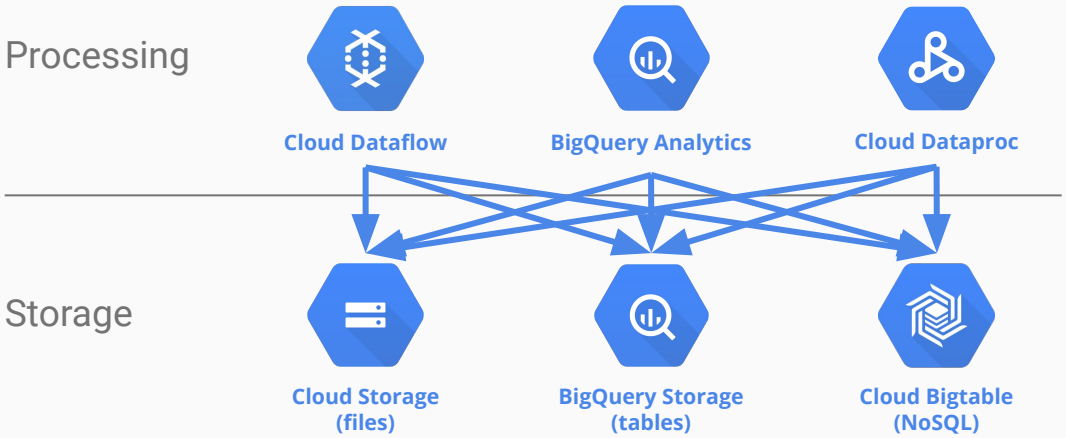
BigQuery
Fully managed data warehouse for large-scale analytics

Cloud Dataflow
Real-time data pipelines, with open source SDK via Apache Beam

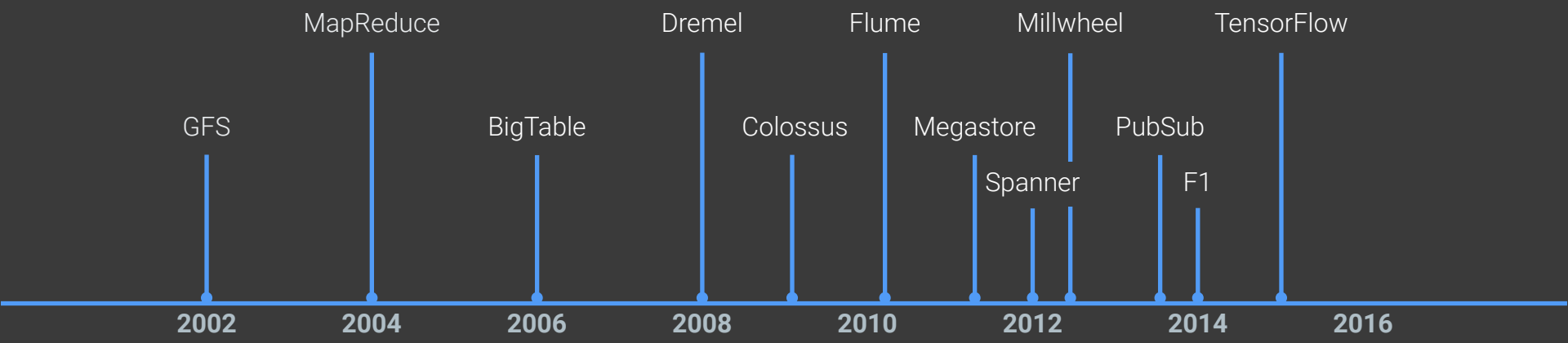


Separation of Storage and Compute

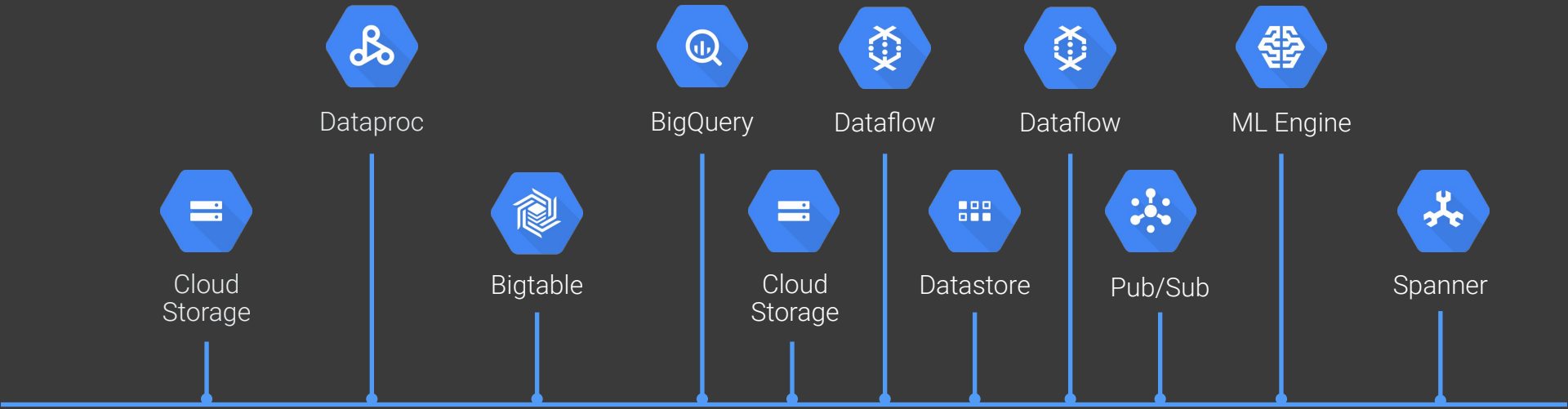
- Access any storage system from any processing tool
- Keep as much data as you want, economically
- Share data in place, no more FTP and copying



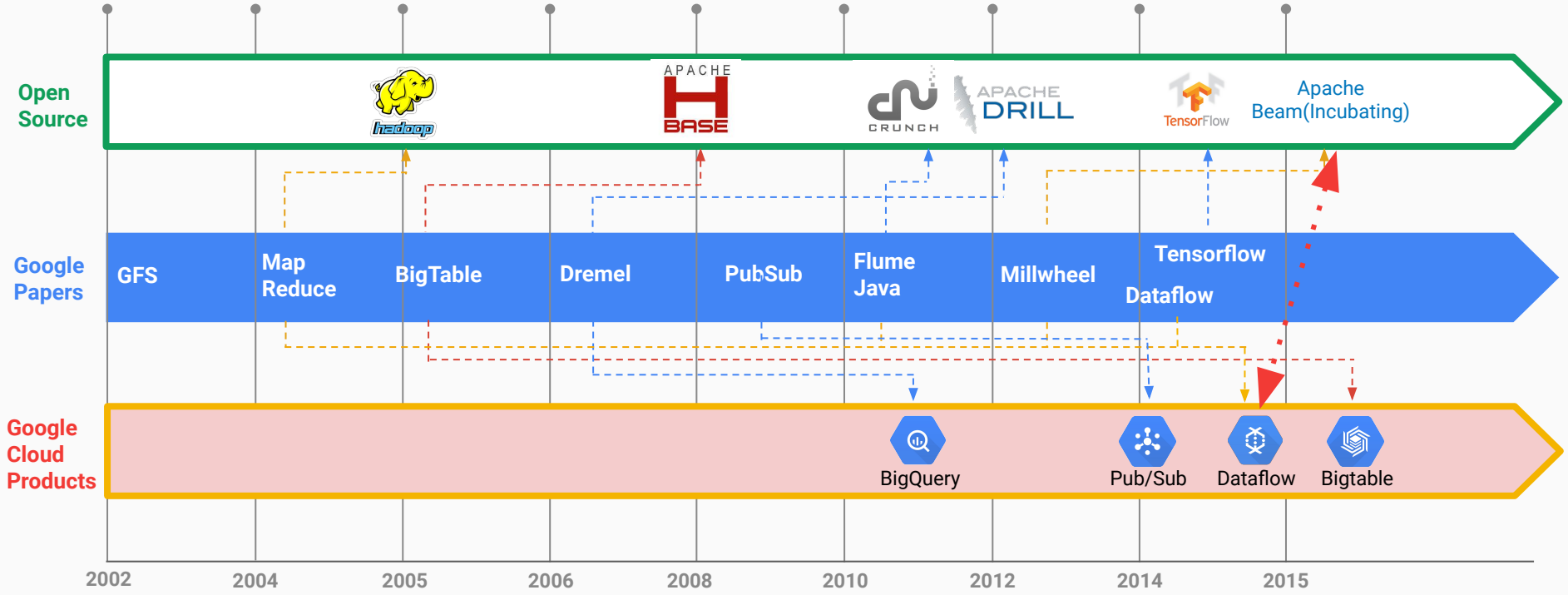
Google's Data Research



Google's Data Products



10+ years of Big Data innovation - Open Source



Product Mapping

kafka

RabbitMQ

Spark SQL

APACHE DRILL

HORTONWORKS

presto

TERADATA

cloudera IMPALA

Spark Streaming

APEX

beam

Flink

hadoop

cloudera hadoop

APACHE Spark

jupyter

amazon web services

Microsoft Azure

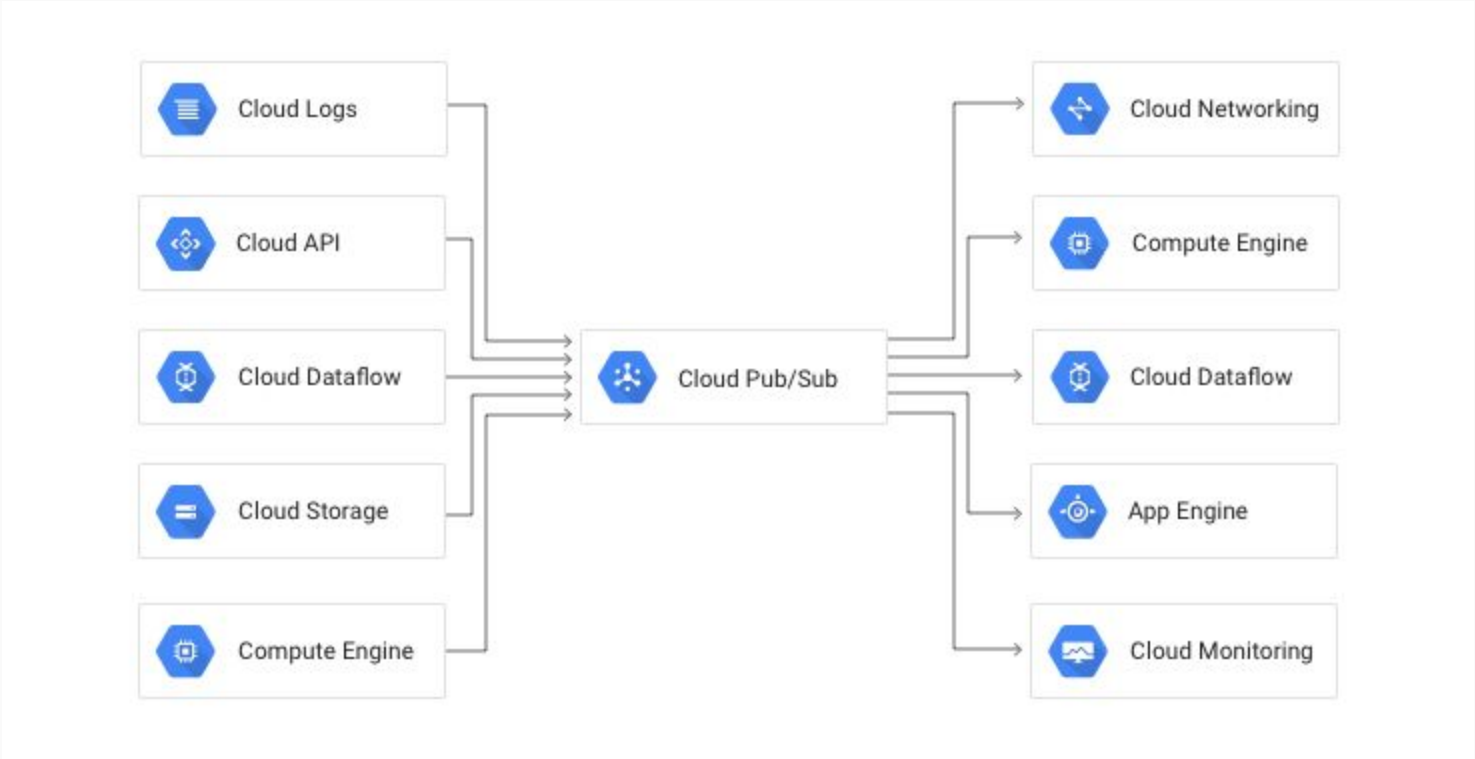
Pub Sub

Pub/Sub: 100% serverless event delivery

Google Cloud
Pub/Sub

- Reliable and real-time messaging
- Global by design and highly available
- Uses Google's private fiber network and worldwide points of interconnect
- Only pay for what you use

Pub/Sub: Google Cloud Pub/Sub passes messages

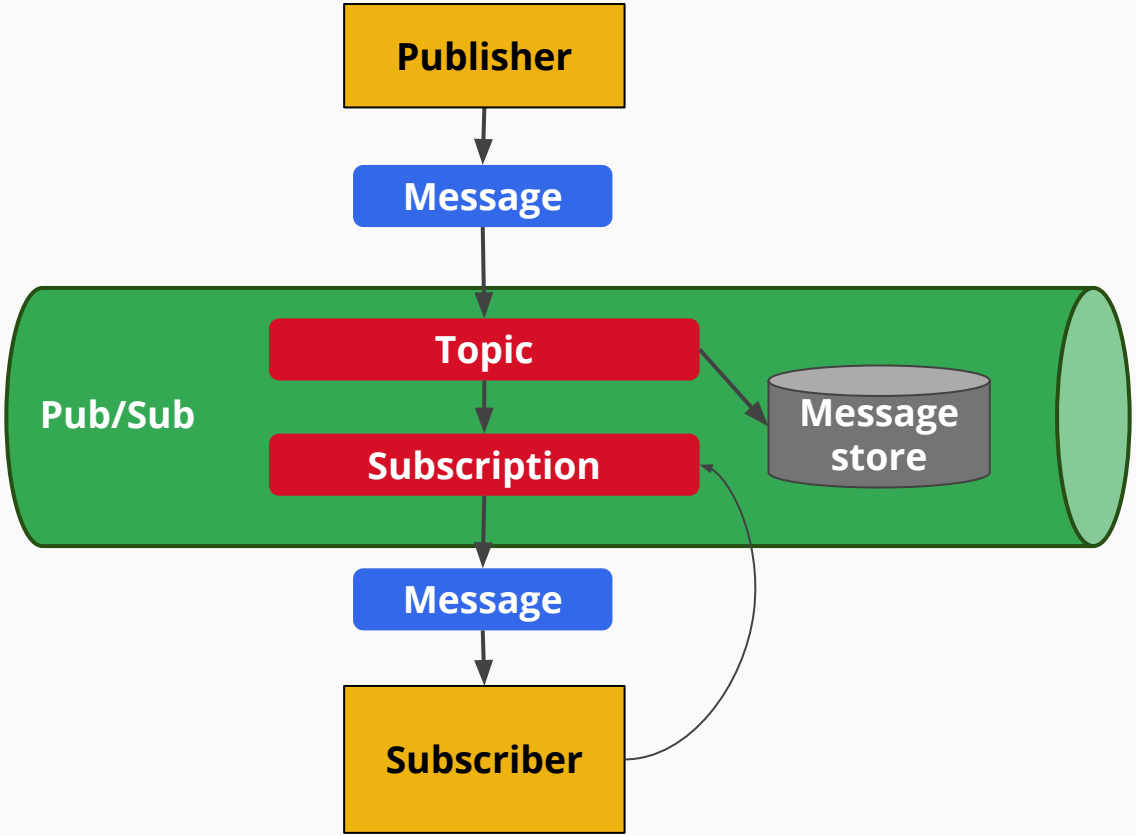


Pub/Sub: What is Google Cloud Pub/Sub

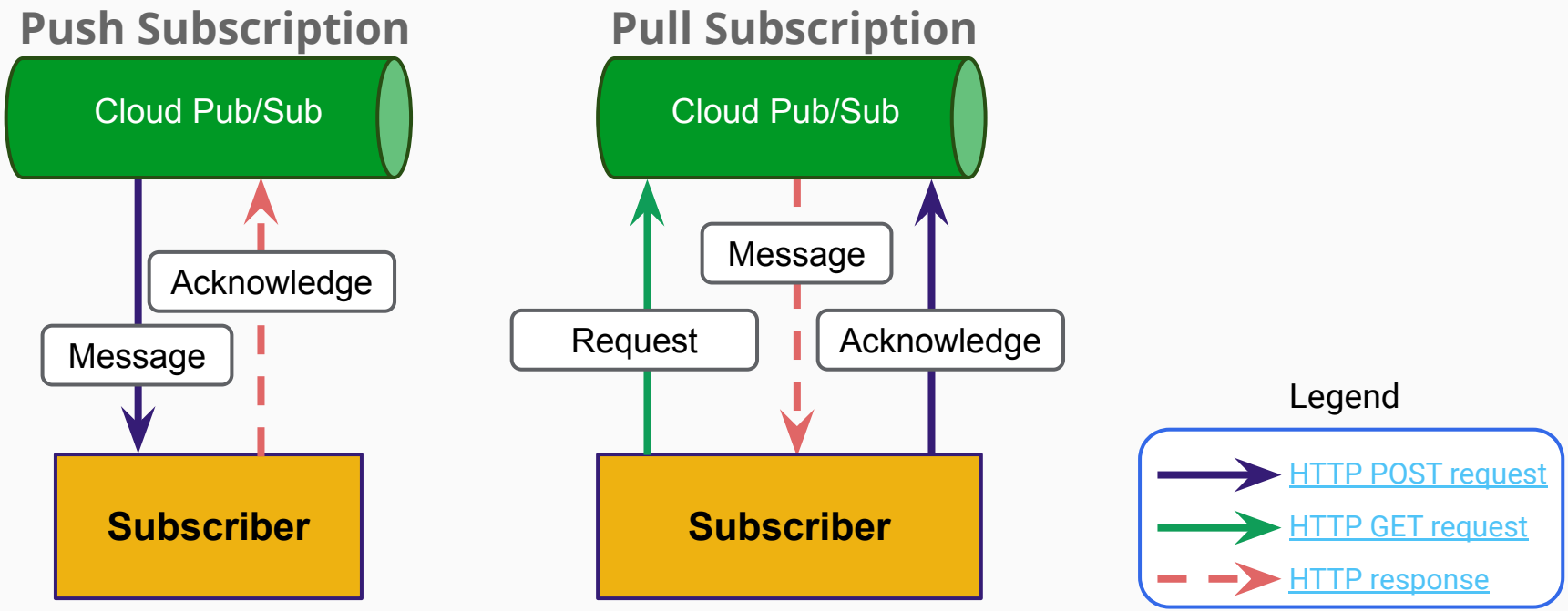
- Cloud Pub/Sub is a service to capture and rapidly pass massive amounts of data (or messages) between software applications with world-class security.
- It uses the **Publish - Subscribe** pattern:
  - **Publisher** applications can send **messages** to a **topic**.
  - **Subscriber** applications can subscribe to that topic to receive the message when the subscriber is ready (asynchronously).
- Pub/Sub acts as a buffer between sending and receiving software applications, making it easier for developers to connect applications.

- Cloud Pub/Sub provides:
  - **Scale**—It supports Google's services, such as Gmail and ads.
  - **Reliability**—Dedicated resources in every Google Cloud Platform region enhance availability without increasing latency.
  - **Performance**—Sub-second notifications even when tested at over 1 million messages per second.
  - **Cost-efficiency**—A "pay for what you use" service.
  - **Ease of use and implementation**—Because it's a fully managed service, there's no need to manage your own open source software implementation. Get started in minutes, not days.

Pub/Sub: Architecture



Pub/Sub: Architecture



Pub/Sub: Messaging is a shock-absorber

Availability

- Buffer messages and requests during outages
- Prevent message overloads that cause outages
- Redirect requests to recover from outages

Throughput

- Smooth out spikes in new request rate
- Balance load across multiple servers
- Balance arrival rate with service rate
- “Fan-in” from many devices

Latency

- Accept requests closer to the network edge
- Optimize message flow across regions

Pub/Sub: Messaging is a shock-absorber

Sources

- New data sources can plug into old data flows
- New data sources can use new schemas
- Common security policies for all sources

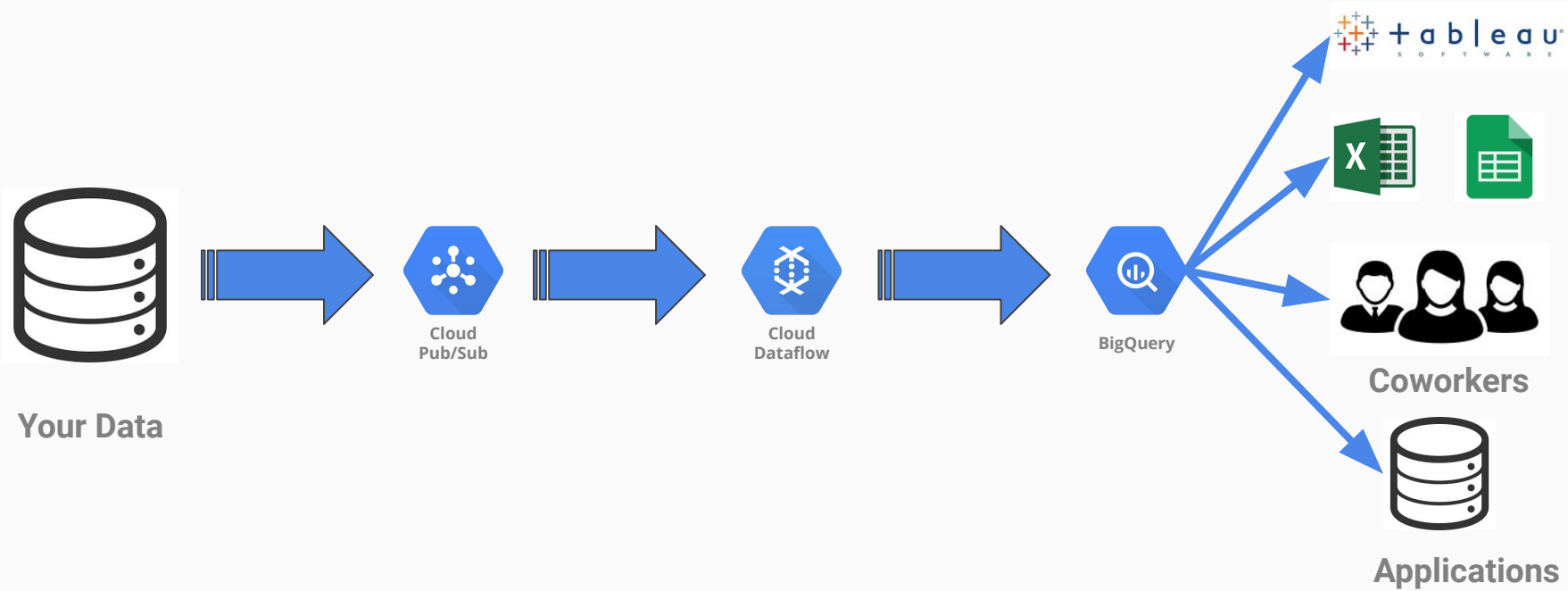
Sinks

- Data can be sent to new destinations
- “Push” and “pull” delivery are both available
- Spans organizational boundaries

Transforms

- Can merge streams into new topics
- Sends messages to Dataproc and Dataflow for transformation

Pub/Sub: Use case: big data stream



Pub/Sub: examples in use at Google

Chat & Mobile

Every time your Gmail displays a new message, it's because of a push notification to your browser or mobile device.

Ads and budgets

One of the most important real-time information streams in the company is advertising revenue—we use Pub/Sub to broadcast budgets to our entire fleet of search engines.

Push notifications

Google Cloud Messaging for Android delivers billions of messages a day, reliably and securely, for Google's own mobile apps and the entire developer community.

Instant search

Updating search results as you type is a feat of real-time indexing that depends on Pub/Sub to update caches with breaking news.

Ejercicio Pub-Sub

Paso 1

Quickstart: Using the Console

This page shows you how to perform basic tasks in Google Cloud Pub/Sub using the Google Cloud Platform Console.

Before you begin

1. Select or create a Cloud Platform project.
   GO TO THE PROJECTS PAGE
2. Enable billing for your project.
   ENABLE BILLING
3. Enable the Pub/Sub API.
   ENABLE THE API
4. Install and initialize the Cloud SDK.

Bonus

Quickstart: Using the gcloud Command-Line Tool

Pub/Sub is a messaging service for exchanging event data among applications and services. A producer of data publishes a messages to a Pub/Sub topic. A consumer creates a subscription to that topic. From this point on, Pub/Sub guarantees that the message will be delivered to every consumer of the message at least once. Consumers either pull messages from a subscription or are configured as webhooks for push subscriptions. Every subscriber must acknowledge each message within a configurable time window. Unacknowledged messages are redelivered. Pub/Sub is geographically global and does not require sharding or additional configuration to scale with demand.

This page shows you how to perform basic tasks in Google Cloud Pub/Sub.

Dataproc

Dataproc

Google Cloud Dataproc

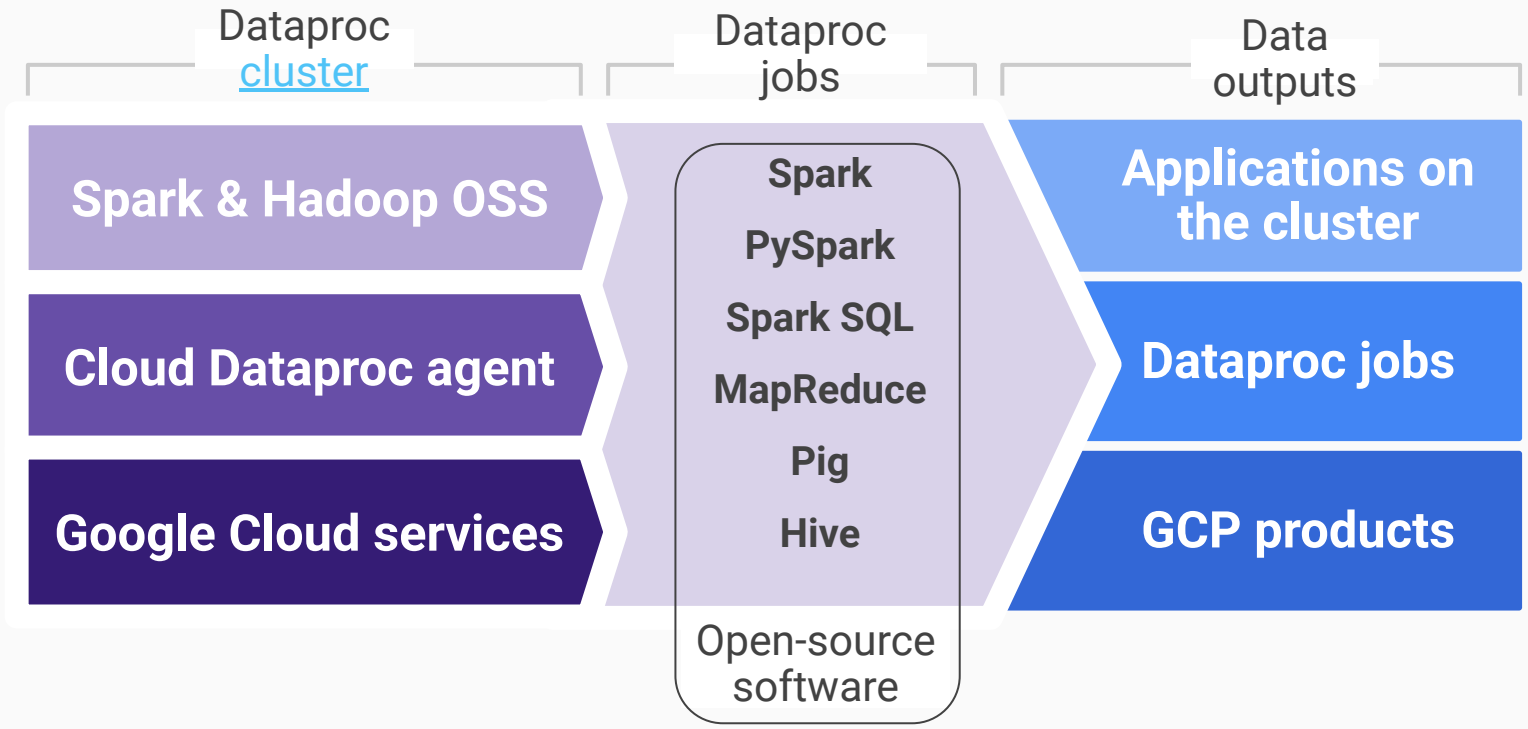
- It is a managed Hadoop MapReduce, Spark, Pig, and Hive service, to easily process big datasets at low cost
- Is a fast, easy to use, low cost and fully-managed service, powered by Google Cloud Platform

Dataproc

Servicio gestionado Spark y Hadoop

- Gestión de Cluster integrado.
- Cluster dimensionables.
- Integración.
- Versionado.
- Herramientas de Gestión.
- Acciones de inicialización.
- Gestión manual o automática.
- Máquinas Virtuales flexibles.

Dataproc


Dataproc cluster
Dataproc jobs
Data outputs
Spark & Hadoop OSS
Cloud Dataproc agent
Google Cloud services
Spark
PySpark
Spark SQL
MapReduce
Pig
Hive
Open-source software
Applications on the cluster
Dataproc jobs
GCP products

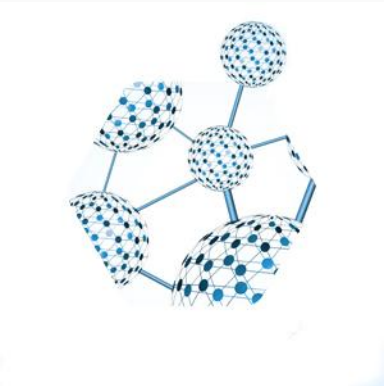
Dataproc

- **Apache Spark™** is a fast and general engine for large-scale data processing.
- The Spark Python API (**PySpark**) exposes the Spark programming model to Python.
- **Spark SQL** is Apache Spark's module for working with structured data.
- **MapReduce** is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.
- **Apache Pig** is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.
- The **Apache Hive** ™ data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive.

Dataproc: Features

Native Spark and Hadoop

Run Spark and Hadoop applications out of the box without modification.



Cloud Integrated

Integrated with Cloud Storage, Cloud Logging, Cloud Monitoring, and more.

Minute-by-Minute Billing

While active, Dataproc clusters are billed minute-by-minute.

Preemptible VMs

Dataproc clusters can make use of low-cost preemptible Compute Engine VMs.

Dataproc: Features

Anytime Scaling

Manually scale clusters up or down based on need, even when jobs are running.

Initialization Actions

Execute scripts on cluster creation to quickly customize and configure clusters.

Developer Tools

REST API and Integration with Google Cloud SDK for rapid development.

Easily Configured

Select between multiple Spark and Hadoop versions; configure properties easily.

Dataproc: Benefits

Low-cost

Enjoy lower total cost of ownership due to low prices and minute-based billing.

Superfast

Spend less time waiting for things to happen and more time hands-on with data.

Customizable

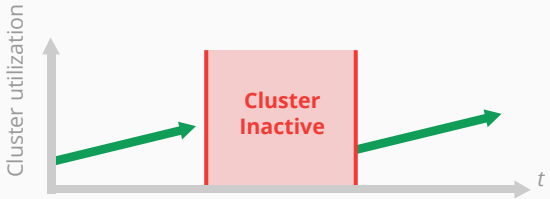
Customizable virtual machine types let you create right-sided clusters.

Easy

Vanilla Spark and Hadoop supported by purpose-built Cloud products.

Dataproc: Scaling

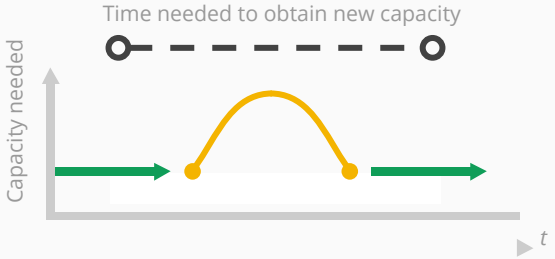
Idle clusters



Most traditional clusters are utilized only a portion of the time they're online

Idle cluster capacity (time, money, computing capacity) is wasted

Scaling inflexibility

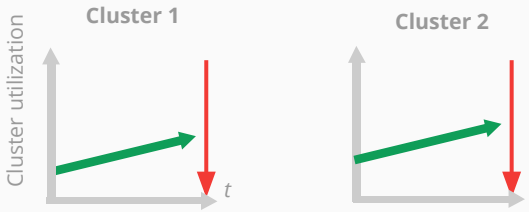


Job demands can be hard to predict, and scaling can take considerable time

Clusters may be constrained at the time when you need them most

Dataproc: Scaling

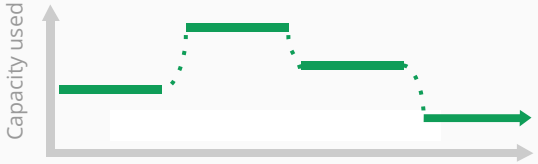
Anytime clusters



Clusters aren't idle. When work is done, simply turn off the cluster and create a new one when required.

Run clusters only when you need them

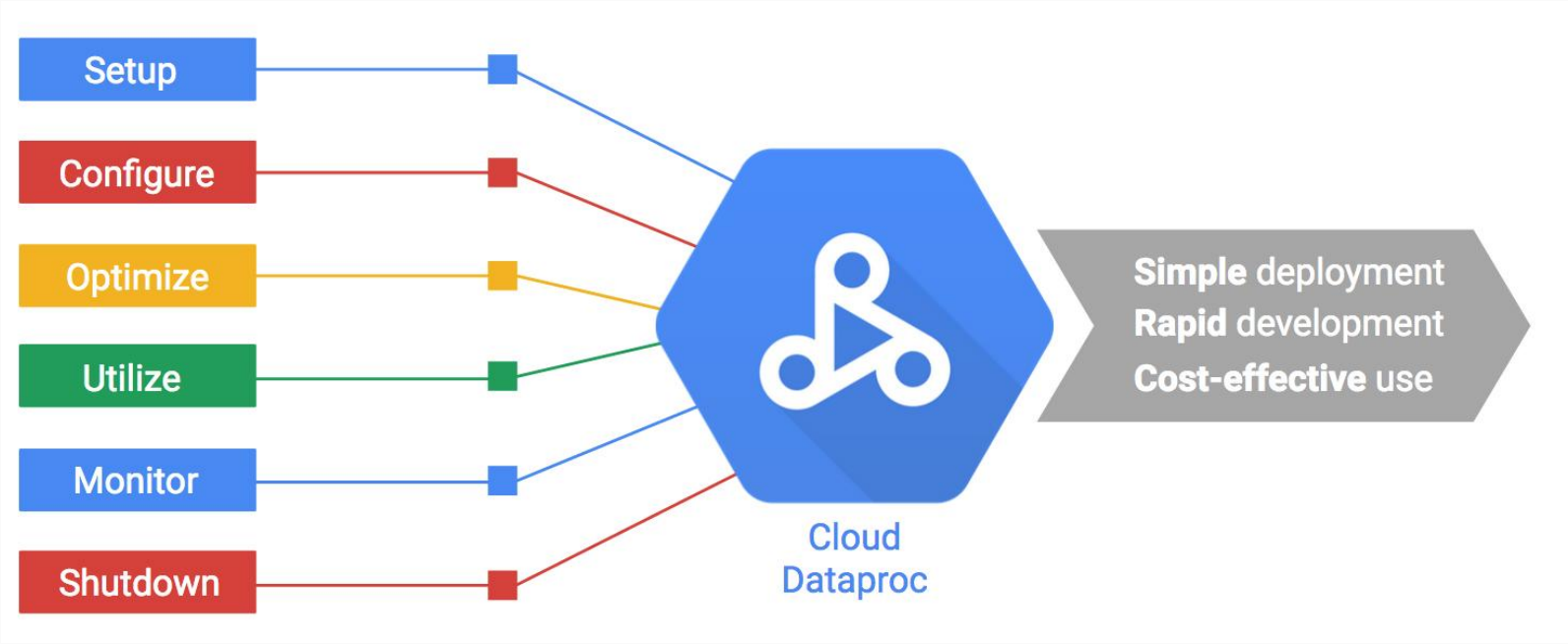
Flexible scaling



By scaling clusters at anytime, your jobs can get exactly the resources they need when required.

Clusters are the right size, anytime

Dataproc: Managed Service



Dataproc: Use Case #1: Log Processing

Need

A customer processes 50 gigabytes of (text) log data per day to produce aggregated metrics. They have used a persistent on-premise cluster to store and process the logs with MapReduce.

How Dataproc addresses this need

Google Cloud Storage can act as a landing zone for the log data for low-cost and high-durability storage. A Dataproc cluster can be created in less than 2 minutes to process this data with their existing MapReduce. Once finished, the Dataproc cluster can then be removed immediately.

Dataproc value

Instead of running all the time and incurring costs even when not used, Dataproc only runs to process the logs which saves money and reduces complexity.

Dataproc: Use Case #2: Ad-hoc Data Mining & Analysis

Need

A customer uses Spark standalone on one computer to perform data mining and analysis. The data is stored locally and they are using the Spark shell to examine the data along with Spark SQL.

How Dataproc addresses this need

Dataproc can create clusters that scale for speed and mitigate any single point of failure. Since Dataproc supports Spark, Spark SQL, and PySpark, they could use the web interface, Cloud SDK, or the native spark shell via SSH to perform their analysis safe from a single machine failure.

Dataproc value

Dataproc quickly unlocks the power of the cloud for anyone without added technical complexity. Running complex computations now take seconds instead of minutes or hours.

Dataproc: Use Case #3: Machine Learning

Need

A customer uses the Spark Machine Learning Libraries (MLlib) to run classification algorithms on very large datasets. They rely on cloud-based machines where they install, and customize Spark.

How Dataproc addresses this need

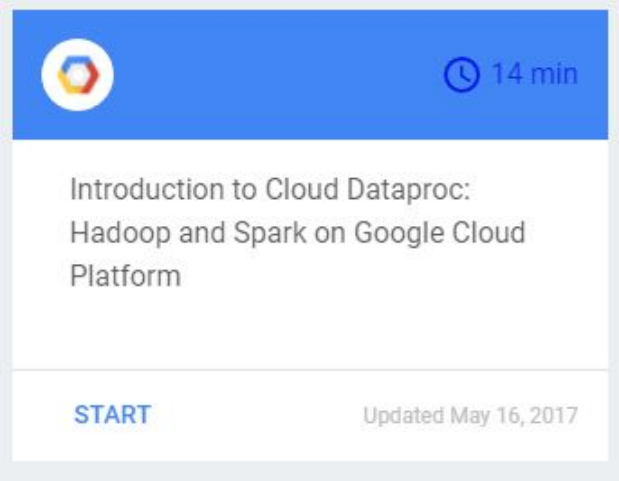
Since Spark and the MLlib installed on any Dataproc cluster, the customer can save time by quickly creating Dataproc clusters. Any additional customizations can be applied easily to the entire cluster via initialization actions. To keep an eye on workflows, they can use the built-in Cloud Logging and Monitoring.

Dataproc value

With Dataproc, resources spent on cluster creation and management can now be focused on the data. Integrations with new Google Cloud products unlock new features for Spark clusters.

Ejercicio DataProc

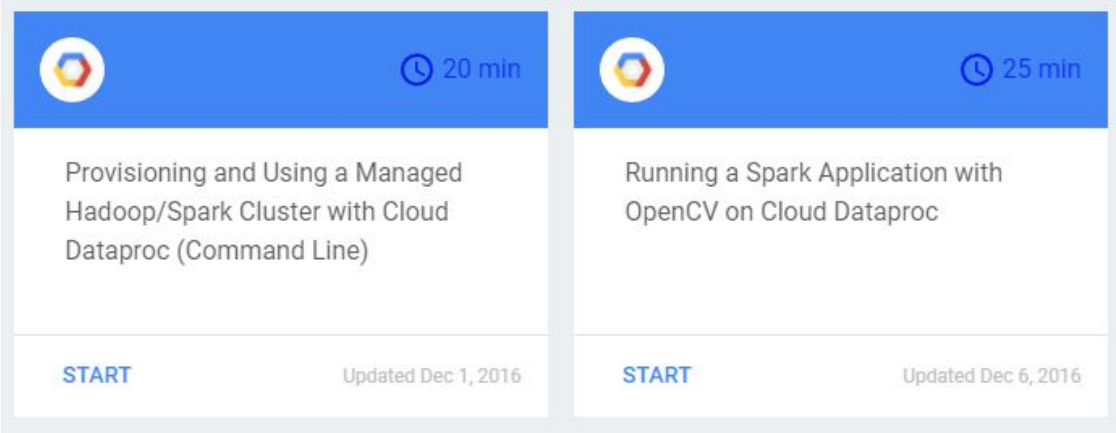
Inicio


14 min
Introduction to Cloud Dataproc: Hadoop and Spark on Google Cloud Platform
START
Updated May 16, 2017

Bonus

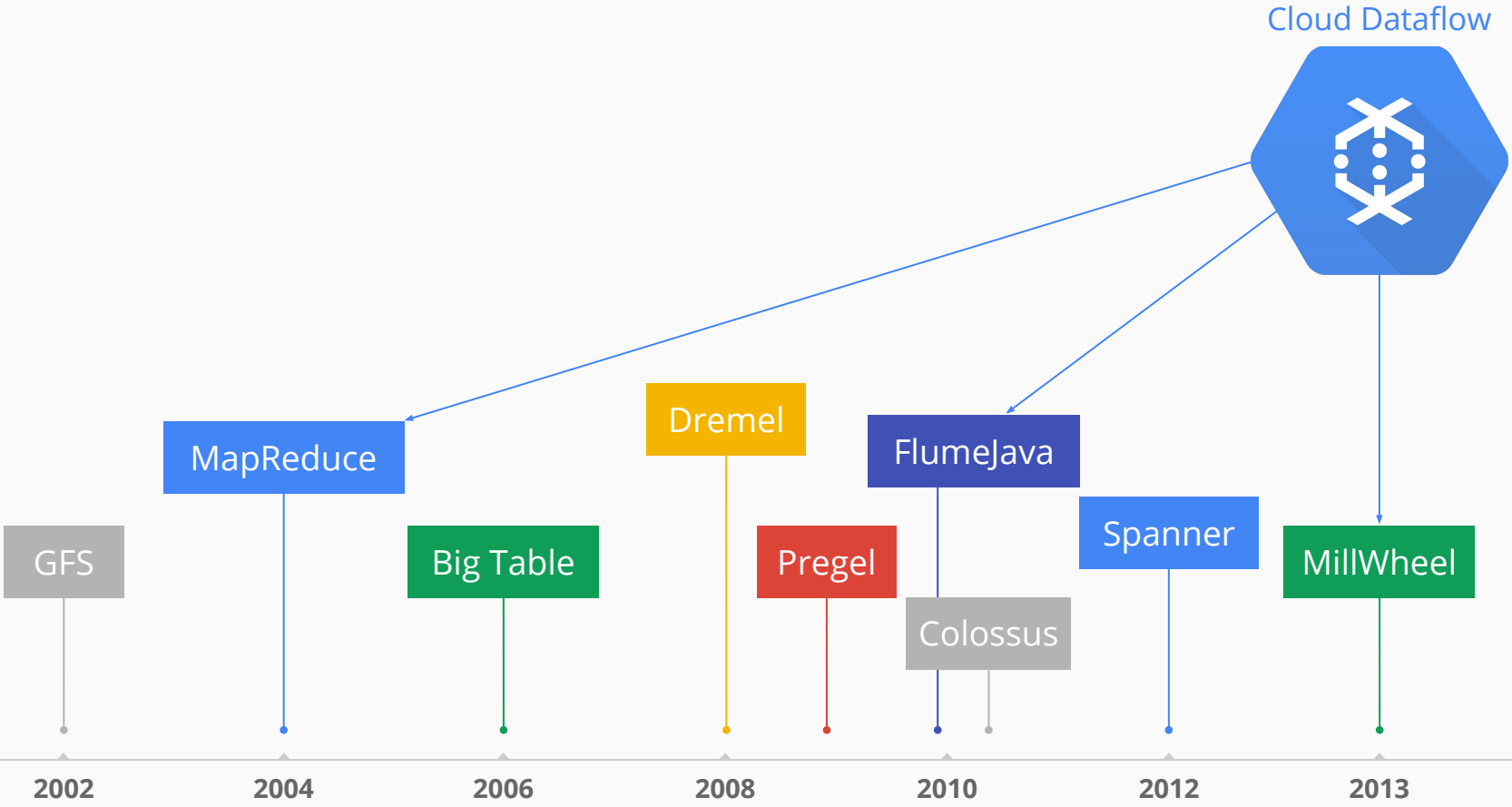
https://codelabs.developers.google.com/

Dataproc



Dataflow

Dataflow - Processing as a Service



Dataflow

Google Cloud Dataflow

- Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation.
- Cloud Dataflow frees you from operational tasks like resource management and performance optimization.

Dataflow

Servicio completamente gestionado y modelo de programación para el proceso de Big Data

- Gestión de Recursos integrado.
- A demanda.
- Ejecución de los trabajos inteligente.
- Auto escalado.
- Modelo de programación unificado.
- Open Source.
- Monitoraje.
- Integración.
- Procesado confiable y consistente.

Dataflow: Managing Massive Data In/Out

ETL

- Movement
- Filtering
- Enrichment
- Shaping

Analysis

- Reduction
- Batch computation
- Continuous computation

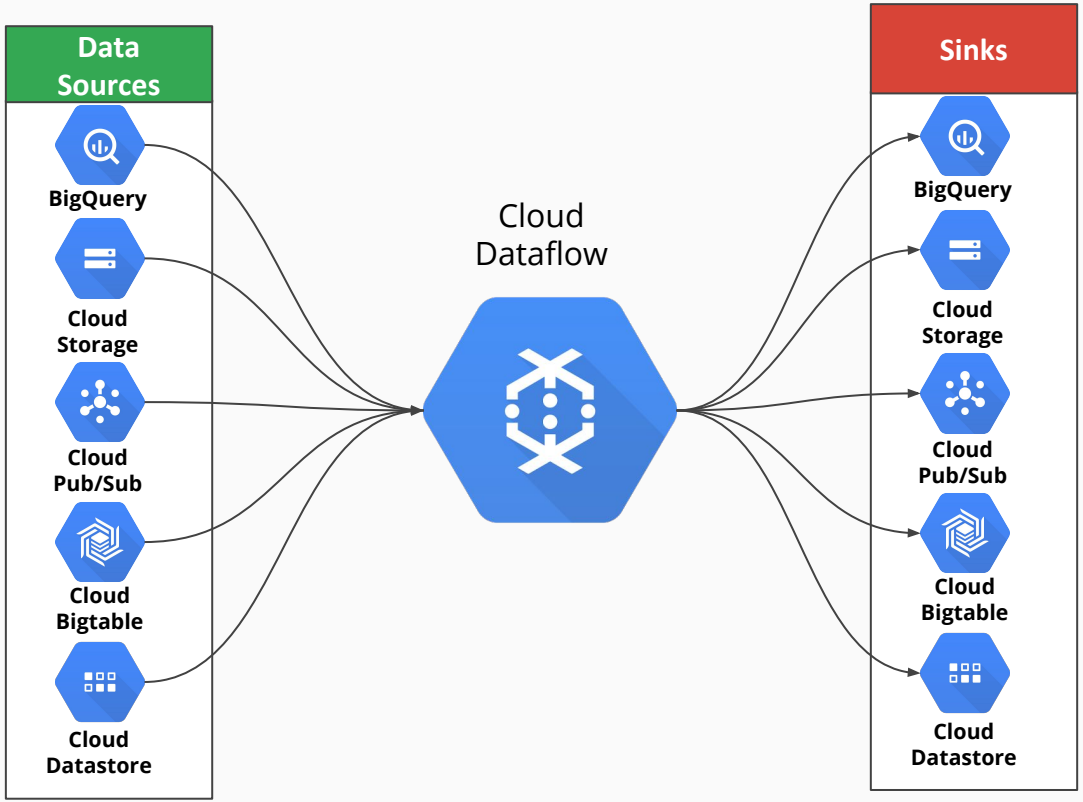
Orchestration

- Composition
- External orchestration
- Simulation

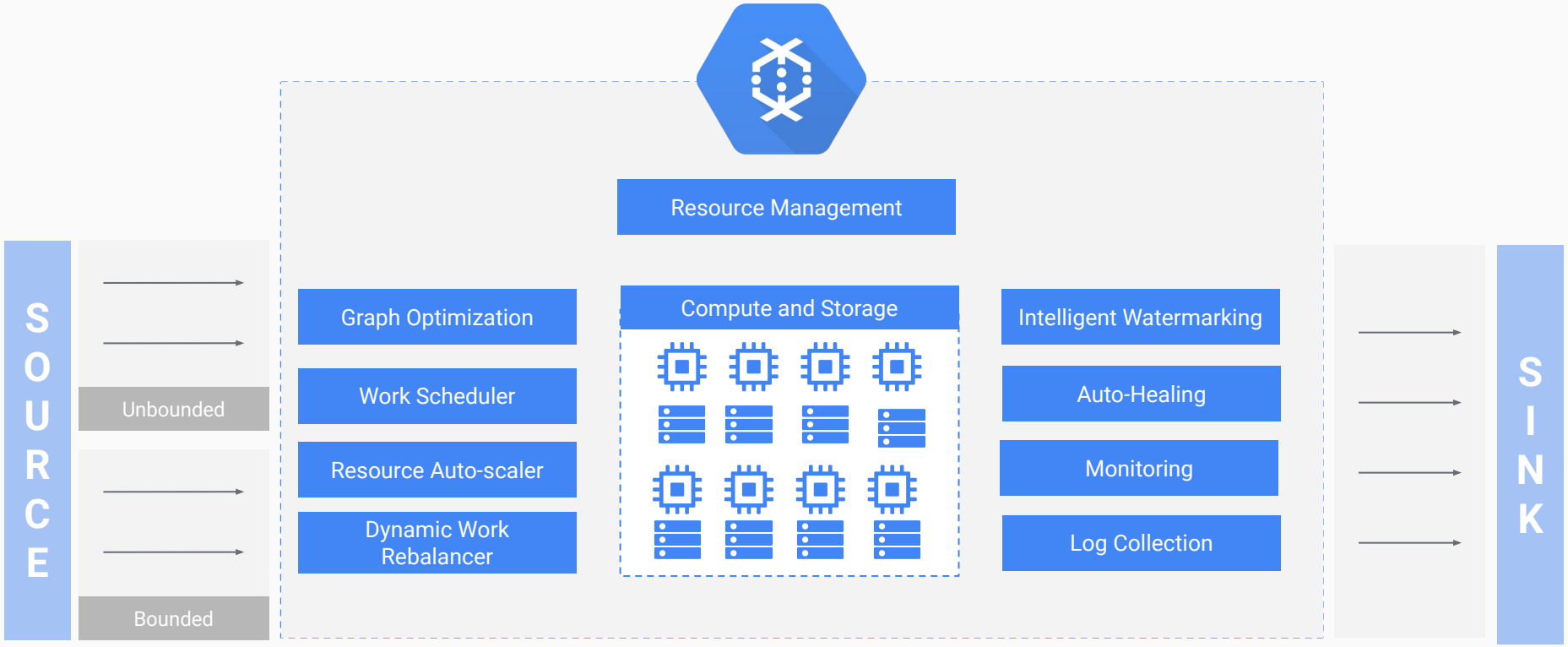
What is Cloud Dataflow?

- Google Cloud Dataflow is a is a tool for developing and executing a wide range of data processing patterns—for example extract, transform, and load (ETL)—on very large data sets.
- Use Cloud Dataflow for nearly any kind of data processing task, encompassing both batch and streaming data processing.
- Dataflow can handle an unbounded or “infinite” data set from a continuously updating source such as Google Cloud Pub/Sub. That is, Dataflow can process practically any amount of data arriving at any time.
- Dataflow is particularly useful for embarrassingly parallel data processing tasks, in which the problem can be decomposed into many smaller bundles of data that can be processed independently, making it very fast (in the same way MapReduce works).

Data Sources and Sinks for Dataflow

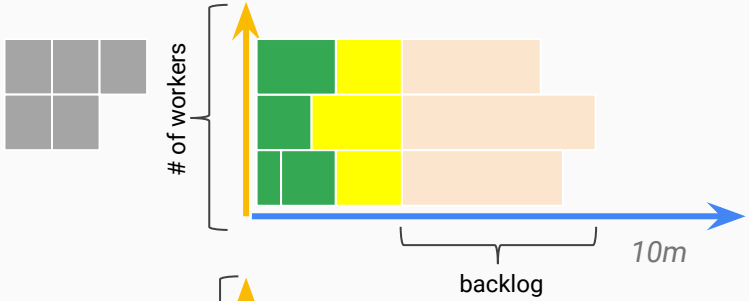


Cloud Dataflow: Under the Hood

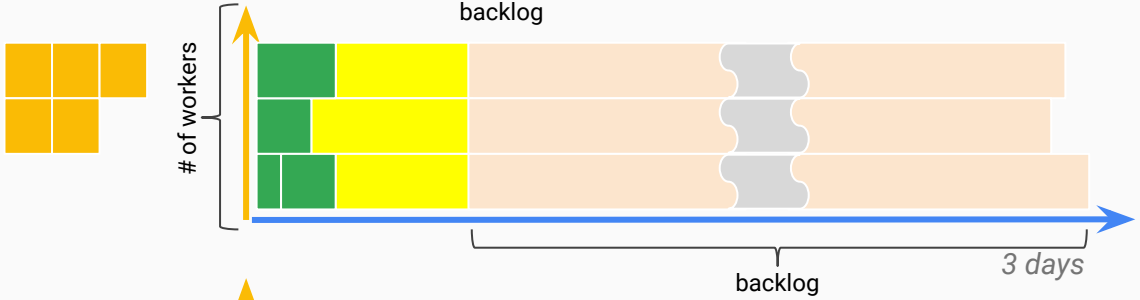


Cloud Dataflow: Autoscaling

Start off with **3** workers,
things are looking okay

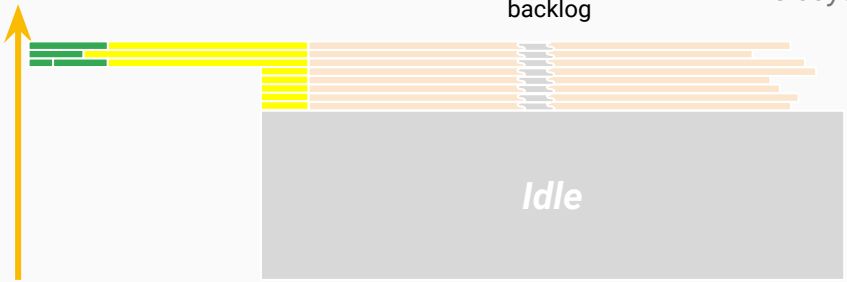


Re-estimation shows there's
orders of magnitude more work:
need **100** workers!



You have 100 workers
but you don't have 100 pieces of work!

...and that's really the most important part



Cloud Dataflow: why do people use it?

Usually **the goal** of big data efforts is to reduce the time required to answer questions for making faster, better decisions.

Examples of questions prospects want answered:

- “How many online sales did I make in the last hour due to advertising conversion?”
- “What was the average viewing time over the past seven days compared to last year?”
- “Which version of my web page do people like better?”
- “Which transactions look fraudulent?”

Cloud Dataflow **speeds up the rate at which questions can be answered** by:

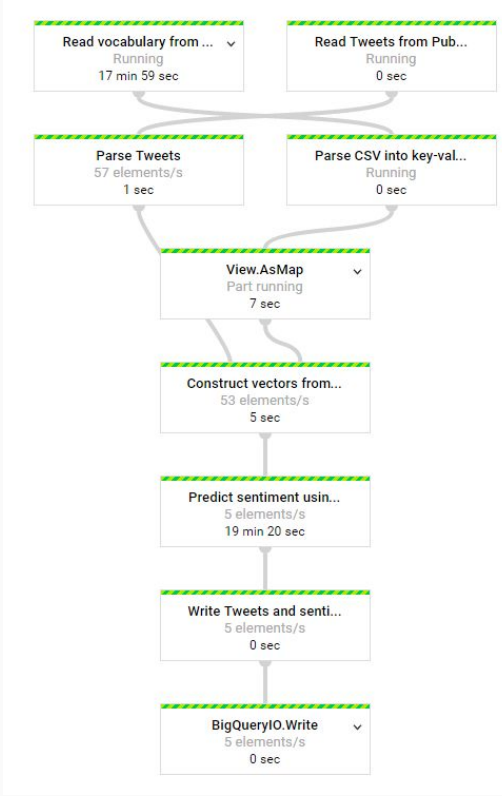
- Integrating data from multiple sources and preparing it for analysis.
- Analyzing event data streams using the Dataflow service.

How Cloud Dataflow works ?

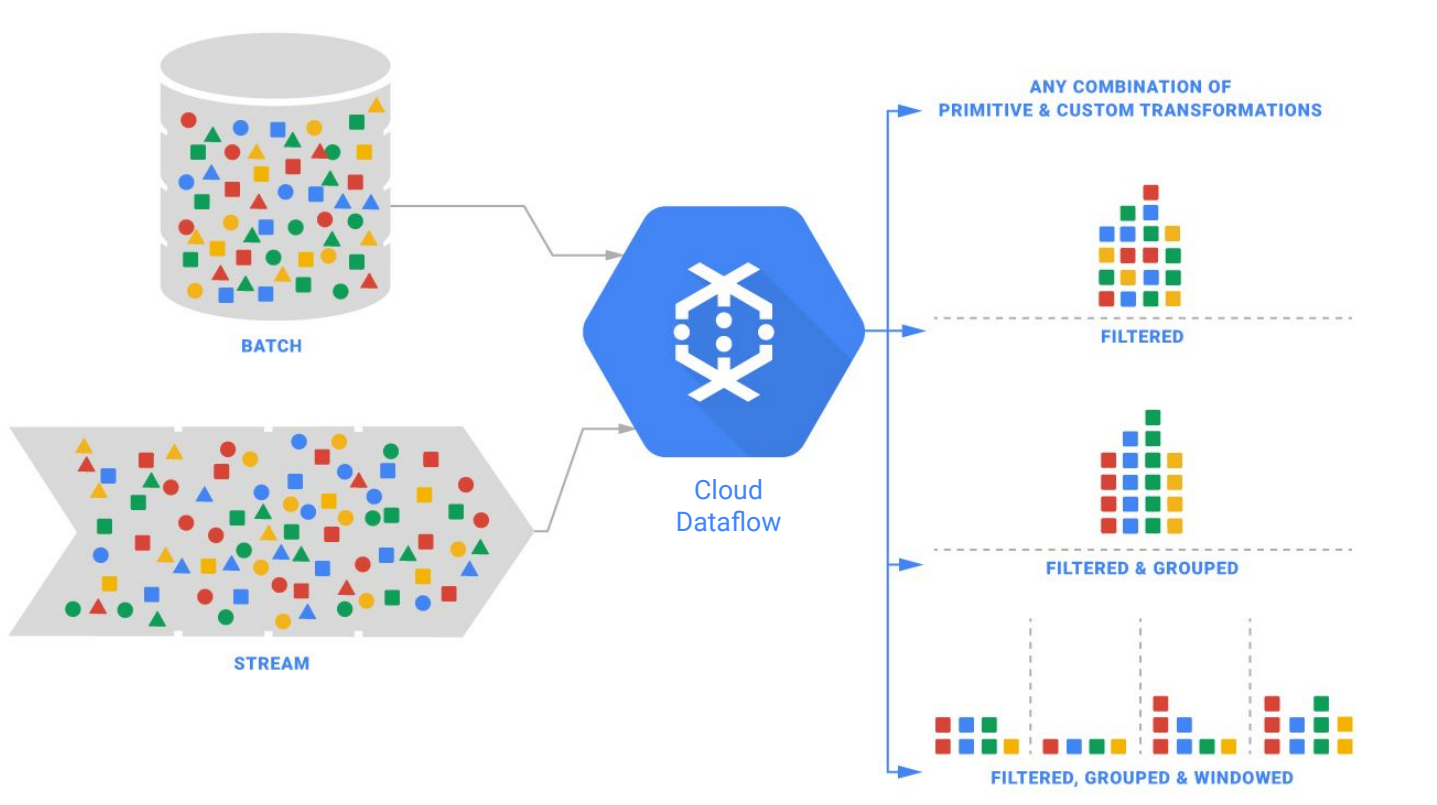
- Dataflow provides an easy way to create your data processing jobs. Each job is represented by a data processing **pipeline** that you create by writing a program.

- Each pipeline reads some input data, performs some transforms on that data to gain useful or actionable intelligence about it, and produces some resulting output data.

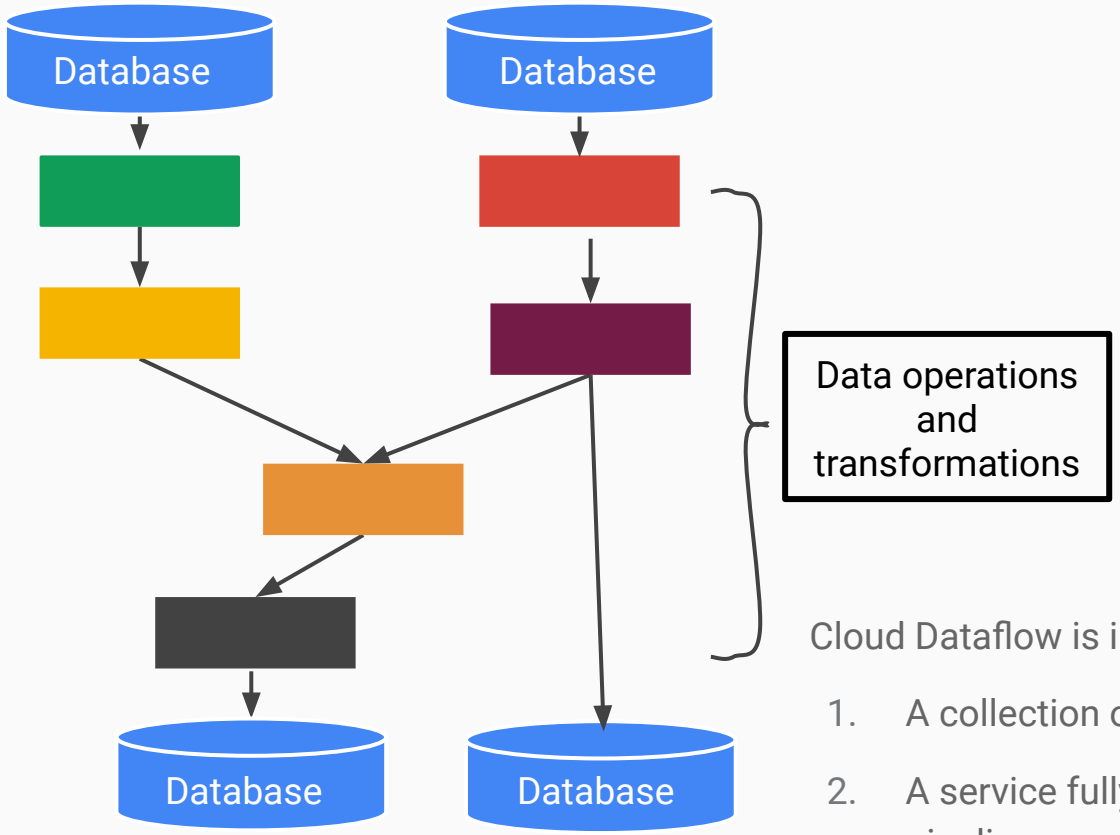
- A pipeline's transforms might include filtering, grouping, comparing, or joining data.



How Cloud Dataflow works ?



Dataflow: What are data pipelines?



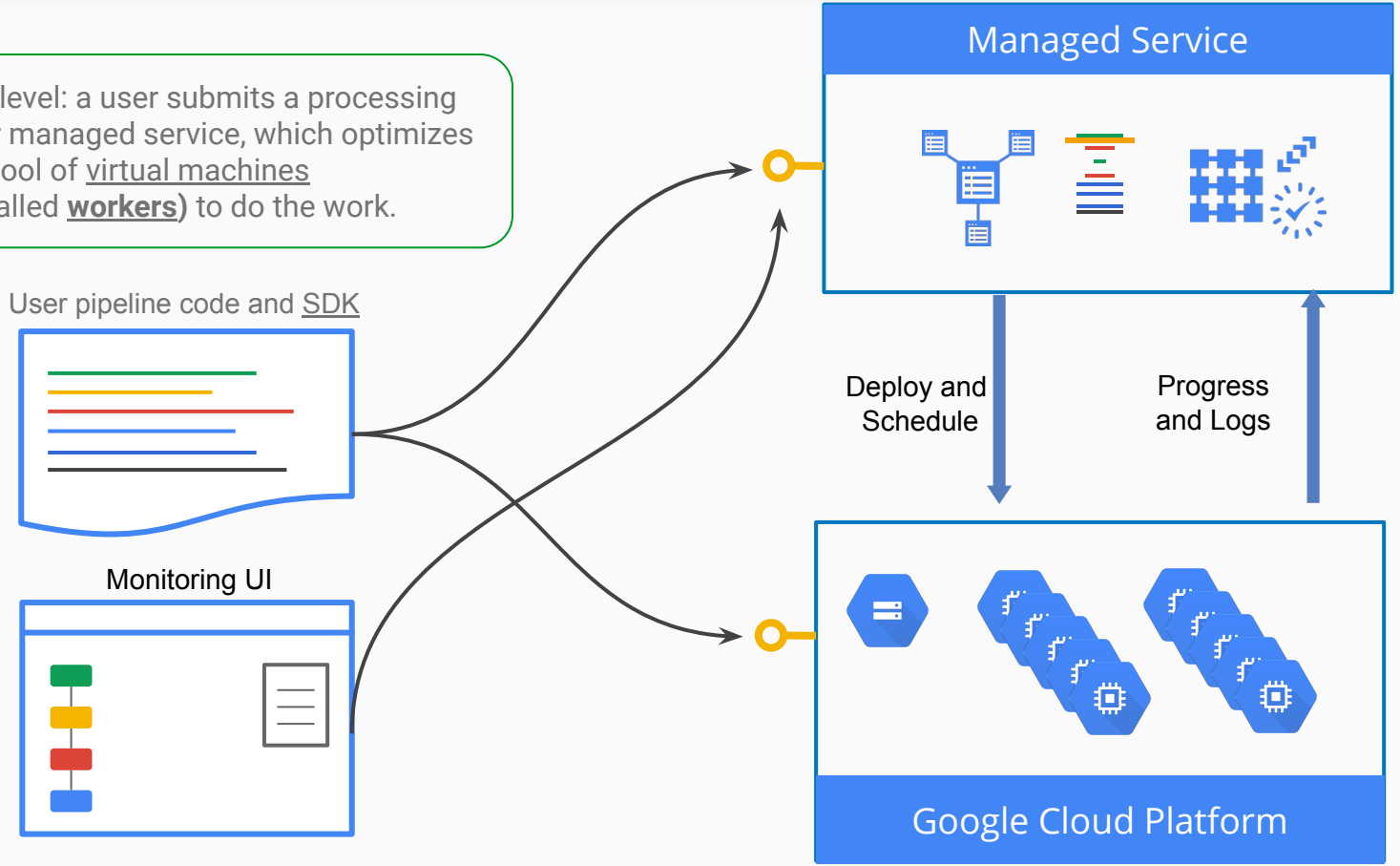
- A pipeline is a defined set of data processing transformations
- Optimized and executed as a unit
- Can include multiple inputs and multiple outputs
- Can perform many mathematical, logical, or transformation operations
- PCollections conceptually flows through the pipeline

Cloud Dataflow is implemented as two distinct elements:

1. A collection of software development kits.
2. A service fully managed by Google for running data pipelines.

Dataflow: Life of a Pipeline

At a very high level: a user submits a processing pipeline to our managed service, which optimizes it and runs a pool of virtual machines (sometimes called **workers**) to do the work.



Dataflow Python code

```python
def run(argv=None):
    """Main entry point; defines and runs the pipeline."""

    parser = argparse.ArgumentParser()
    parser.add_argument('--input',
                        dest='input',
                        default='gs://table/table.csv',
                        help='Input file to process.')

    parser.add_argument('--output',
                        dest='output',
                        default='project:dataset.table01',
                        help='Output BigQuery table for results specified as: PROJECT:DATASET.TABLE
    known_args, pipeline_args = parser.parse_known_args(argv)

    # Run the pipeline (all operations are deferred until run() is called).

    p = beam.Pipeline(options=PipelineOptions(pipeline_args))

    (p
    # Read the text file[pattern] into a PCollection.
    | 'Read from a File' >> beam.io.ReadFromText(known_args.input,skip_header_lines=1)
    | 'String To BigQuery Row' >> beam.ParDo(FormatDoFn())
    # Write the output using a "Write" transform
    | 'Write to BigQuery' >> beam.io.WriteToBigQuery(
        known_args.output,
        schema=TABLE_SCHEMA,
        create_disposition=beam.io.BigQueryDisposition.CREATE_IF_NEEDED,
        write_disposition=beam.io.BigQueryDisposition.WRITE_APPEND)

    # Run the pipeline (all operations are deferred until run() is called).
    )
    p.run()
```

Dataflow technologies that customers value

Automated input "sharding"

Removing the requirement to pre-sort input data

Dynamic work rebalancing

Removing the investment in manually balancing the load between resources

Auto-scaling of work resources

Removing the investment in manually scaling resources to match needs

Unified programming model

Developers can express the computation needs regardless of batch or streaming data input

Logical monitoring

Provides developers a logical view of pipeline behavior vs. a control view

On-demand resourcing

All resources are provided on demand, providing nearly limitless resource scale

Why do customers value Dataflow?

Less overhead-one system

Dataflow reduces operational tuning and operation management overhead found in traditional batch or streaming processing systems. Dataflow fully manages resources in Google Cloud Platform on a per-job basis, including spinning up and shutting down workers and accessing Google Cloud Storage buckets for both I/O and temporary file staging.

Price vs. performance

Dataflow provides intelligent optimization (without developer intervention) of resources to provide optimum price and performance.

Dataflow solves this pain points

Lack of existing ETL solution for BigQuery

Attempting to adopt or extend a BigQuery implementation and they need a reliable, scalable way to move, cleanse, and load data into BigQuery.

Dataflow provides optimized ETL for BigQuery.

Hadoop MapReduce cluster at capacity

Have existing investment in on-premise or on-cloud Hadoop, but they're at capacity and can't or don't want to make further investments in hardware or development.

Dataflow provides on demand and nearly limitless elasticity to offload or build new workflows outside of Hadoop.

Struggling to launch streaming Spark cluster

Trying to get a real-time streaming processing application spun up.

Struggling with overhead of self deploying and managing a Spark cluster.

Dataflow provides a unified batch and streaming model that is fully managed.

Developers focus on development, not on operations.

Migrate existing MapReduce

Would like to migrate existing MapReduce or generic batch process to a real-time stream processing model.

Want to take a batch job; for example, process a log file every 24 hours—and migrate it to process a continuous stream of data and report on that data every 5 minutes.

Dataflow solves this pain points

Too much cost with Hadoop cluster

Typically will have 1 (if not 2) dedicated headcount just for cluster management. Dataflow effectively removes this cost, since the service is fully managed.

"Burstiness" of data rates or processing needs

Existing systems; for example, Hadoop cannot dynamically shift resources to meet incoming data rate changes.

This pain point is exacerbated by the fact that their incoming data rates are growing beyond their control, or the business is asking questions faster than they can respond.

Needs large-scale computation to develop metrics for management dashboards

Traditionally, these metrics would be built after data was loaded into their SQL database. This uses more resources, gets slower as data grows, and is troubled by data contention.

Dataflow provides computation language primitives, which can be run as a combination with traditional ETL processing.

Managing two systems: 1 - batch, 1- streaming

More advanced customers may have already implemented both a batch and streaming solution; however, they are then left with managing two different systems.

Dataflow provides a **unified model** for batch and stream processing.

Dataflow use cases

Batch data movement

Moving at rest data from one system to another, such as from Google Cloud Storage to BigQuery

Data reduction and enrichment

Reduce, compress, re-shape existing data into smaller, computed values, such as log files and geo tags

Continuous computation

Analyze real-time streaming inputs, such as click streams

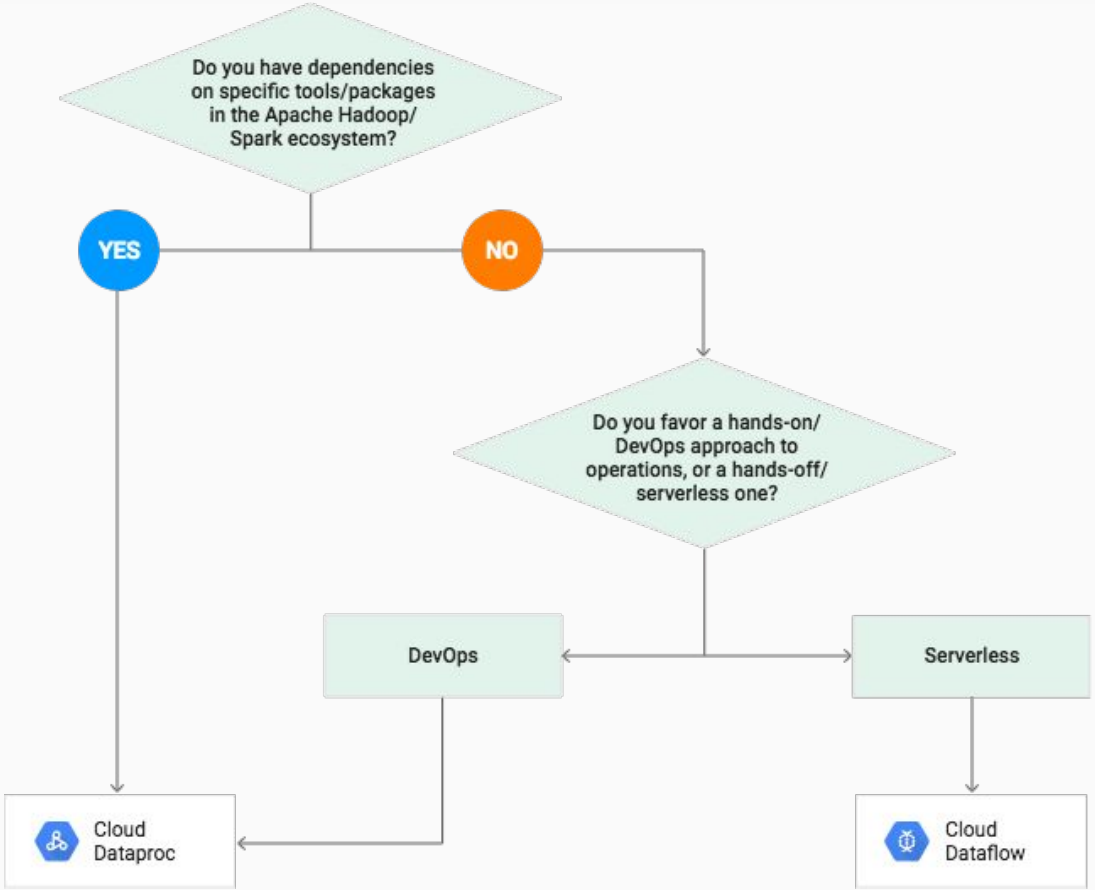
Continuous data movement

Real-time ETL over streaming inputs

Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Dataproc vs Dataflow



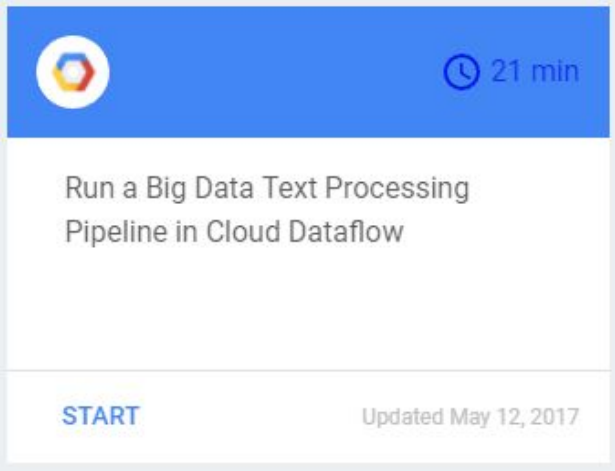
Dataproc vs Dataflow

Recommended Workloads

WORKLOADS	CLOUD DATAPROC	CLOUD DATAFLOW
Stream processing (ETL)		✓
Batch processing (ETL)	✓	✓
Iterative processing and notebooks	✓	
Machine learning with Spark ML	✓	
Preprocessing for machine learning		✓ (with Cloud ML Engine)

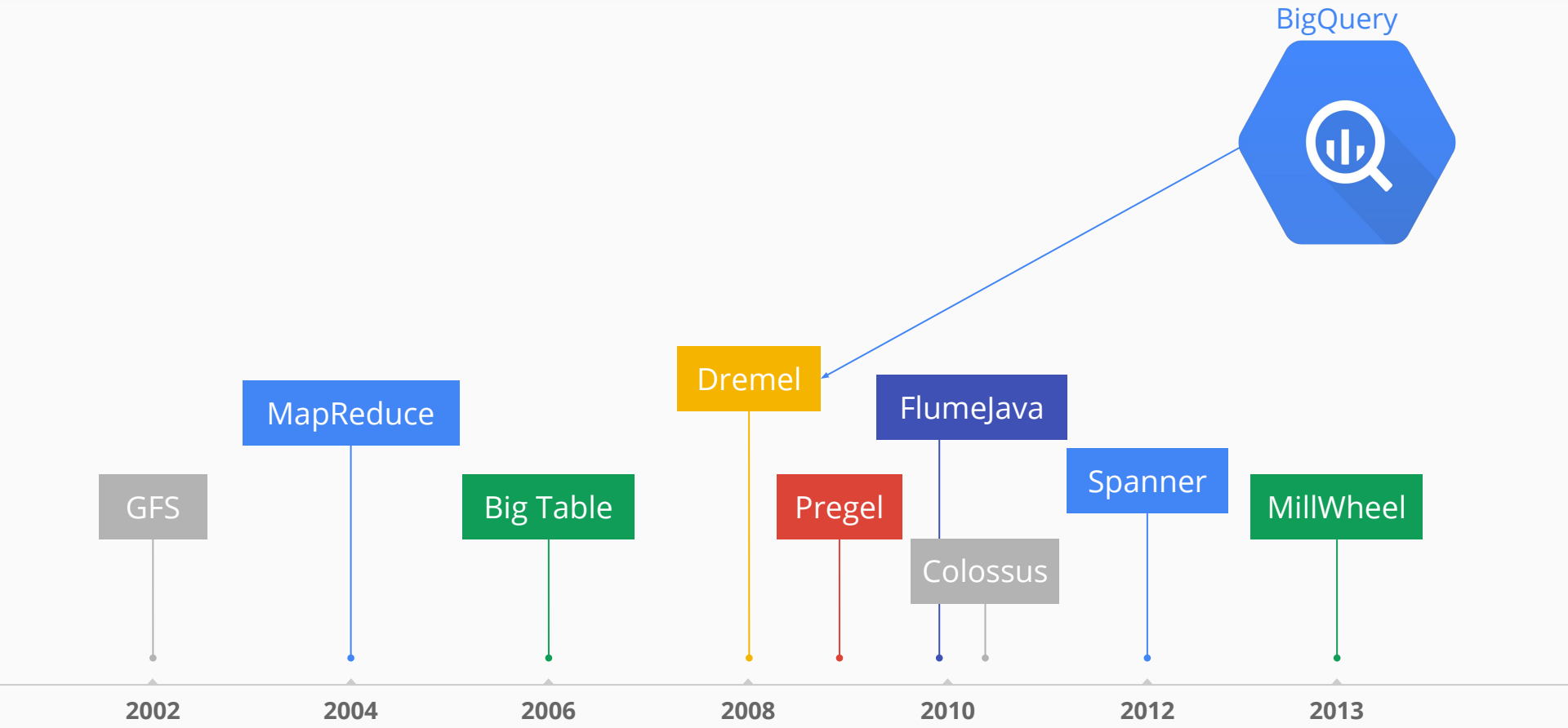
Ejercicio DataFlow

Inicio


21 min
Run a Big Data Text Processing Pipeline in Cloud Dataflow
START
Updated May 12, 2017

BigQuery

BigQuery - Analytics as a Service



BigQuery: 100% serverless data warehouse

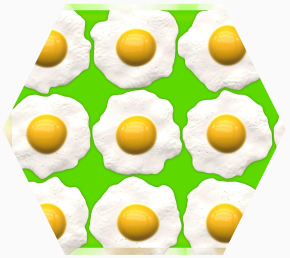
Google
BigQuery

- Google Cloud's Enterprise Data Warehouse for Analytics
- Petabyte-Scale and Fast Convenience of SQL
- Encrypted, Durable and Highly Available
- Fully Managed and Serverless

BigQuery: Is a great choice because

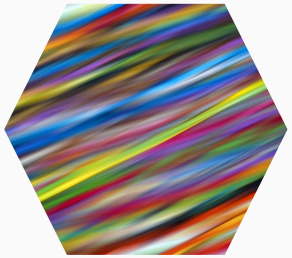
Near-real time analysis of massive datasets

No-ops; Pay for use



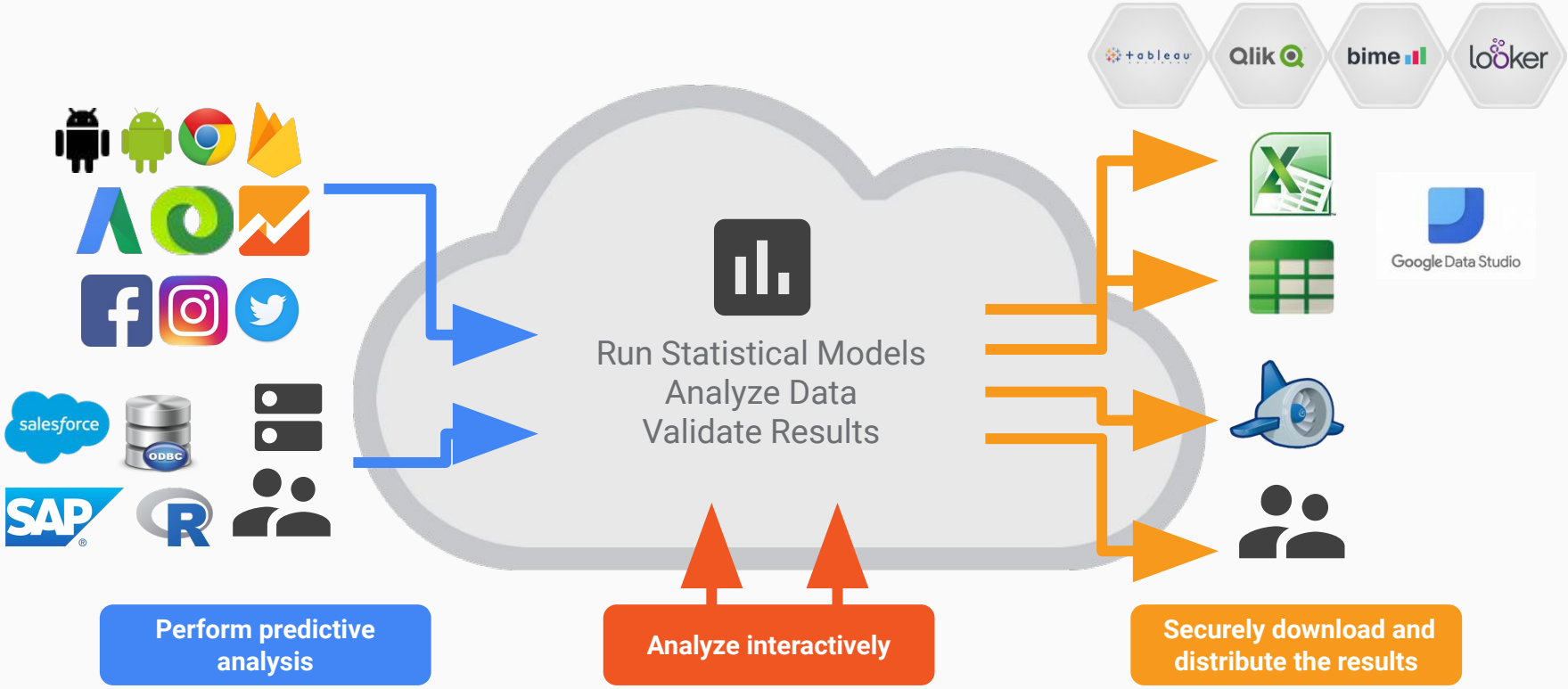
Durable (replicated), inexpensive storage

Immutable audit logs

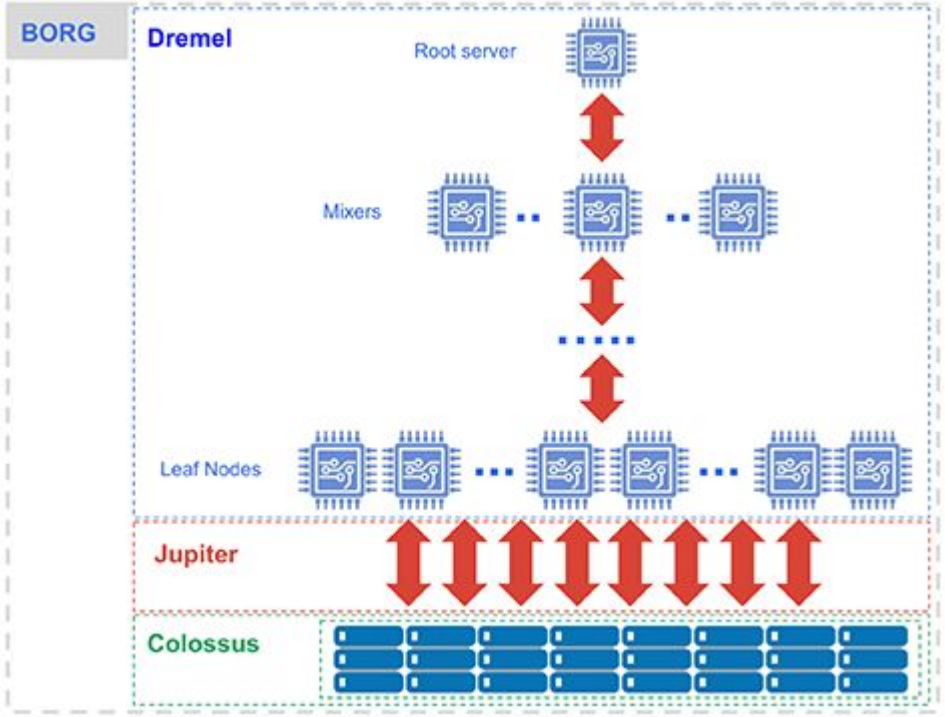


Mashing up different datasets to derive insights

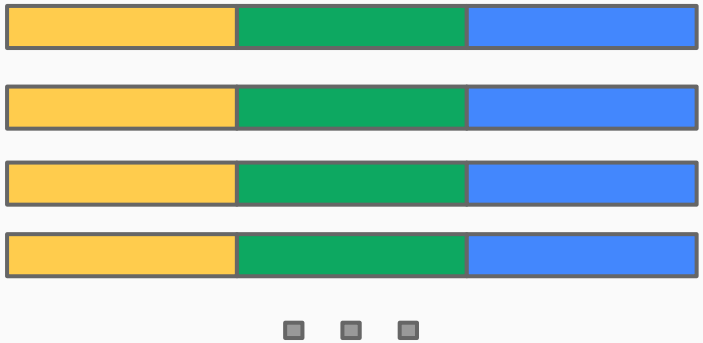
BigQuery - SQL meets Big Data



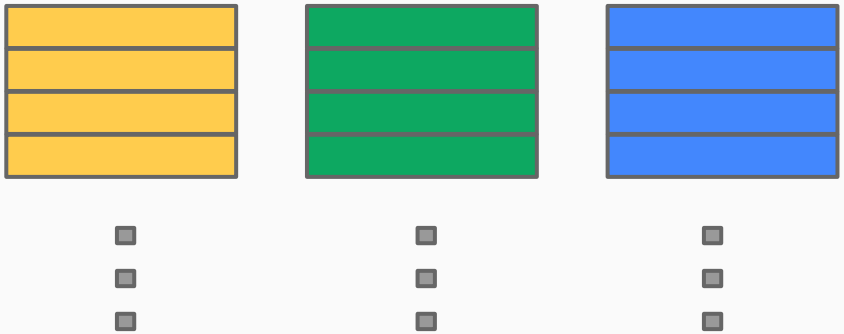
BigQuery - Arquitectura



BigQuery: Columnar Storage

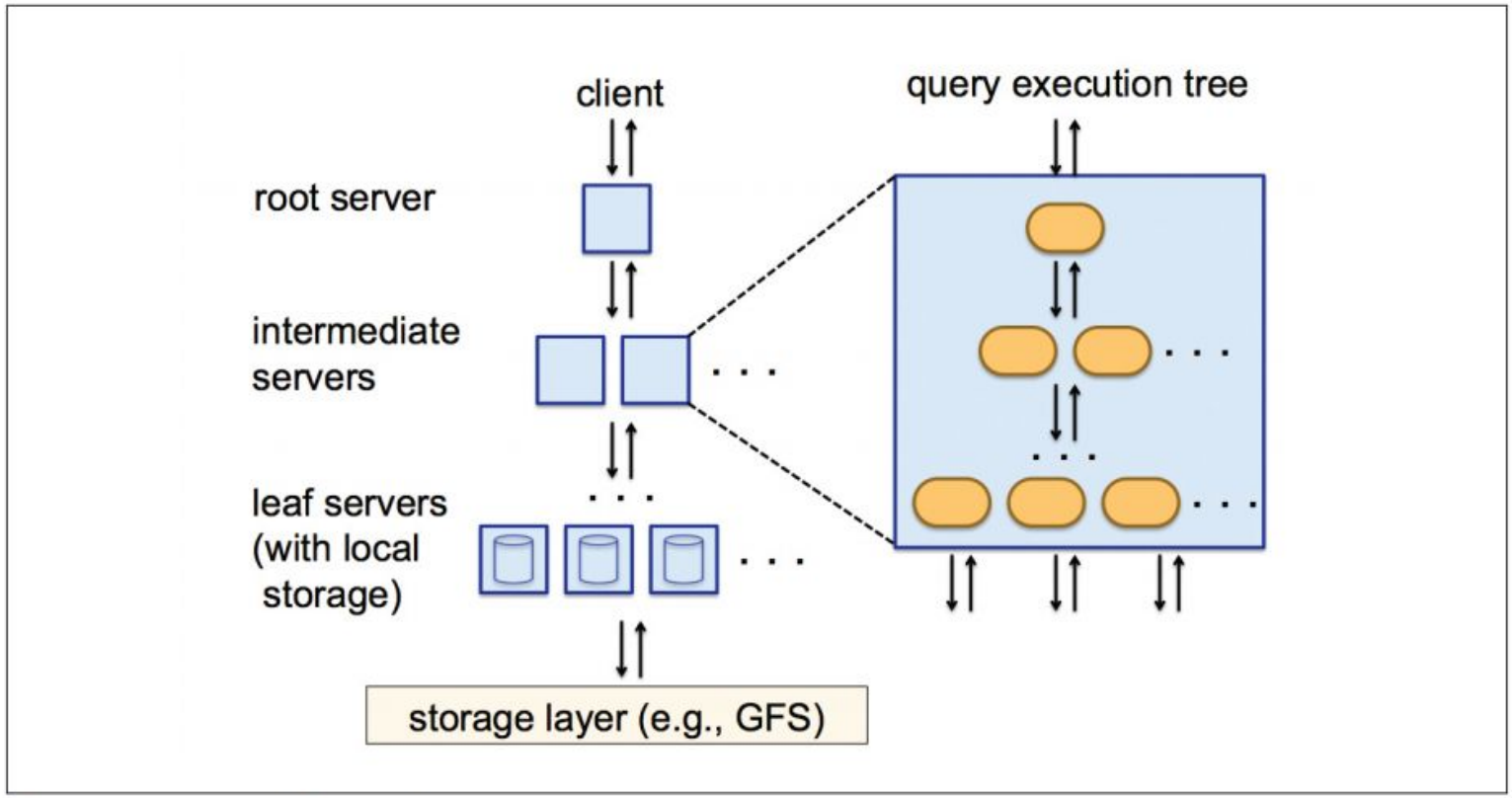


Record-Oriented Storage



Column-Oriented Storage

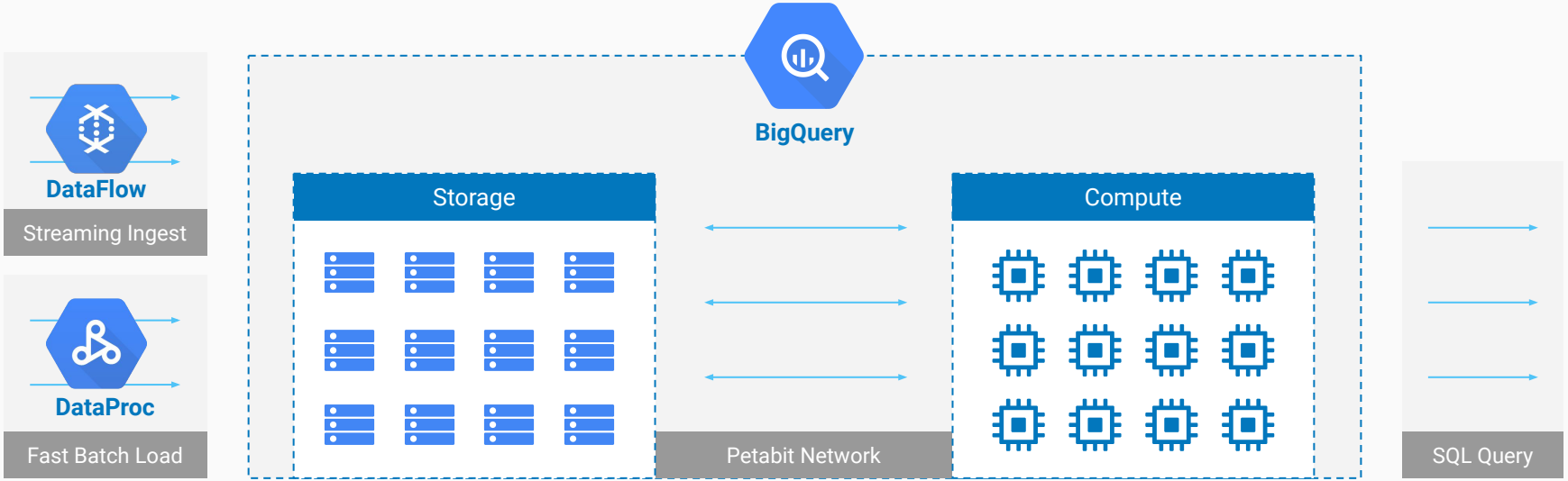
BigQuery - Arquitectura



Tree architecture of Dremel

BigQuery: Massive Parallel Processing Engine

BigQuery = **Massively Parallel Processing** query with the petabit network and thousands of servers



Datalab

Datalab

Google Datalab

- For data scientists
- Notebook interface
- Leverage existing Jupyter modules and knowledge
- Suitable to interactive data science and machine learning
- Closely integrated with BigQuery and CloudML

Why Choose Cloud Datalab?

Simple & Easy to Use

Rich and simple toolkit at your fingertips for:

- Data exploration
- Data transformation
- Data analysis

Complete & Integrated

Integrated set of products:

- Cloud Storage
- BigQuery
- Cloud DataStore
- Cloud SQL

to bring data into a notebook

Productivity for All

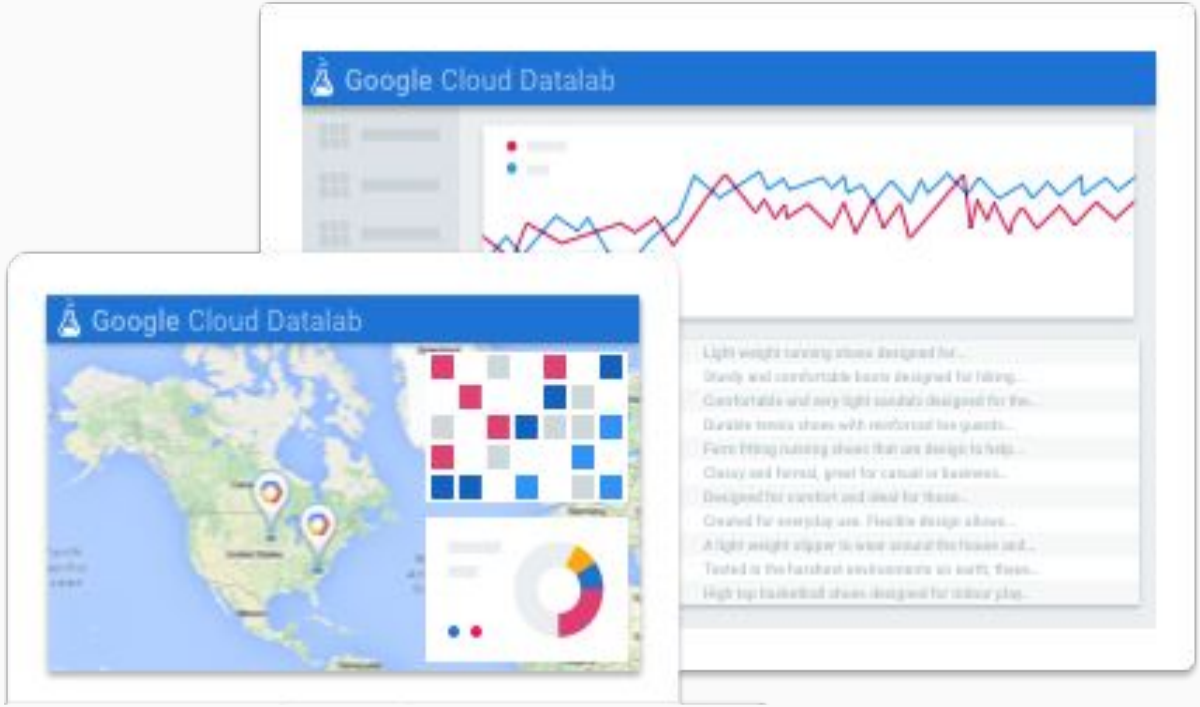
Get started easily and scale up to complex analyses:

- Analysts (SQL)
- Developers (Python)
- Data scientists (ML models)

What is Google Cloud Datalab?

Datalab is a tool that helps you:

1. Gain insight into raw data.
2. Present, collaborate, and publish data & insights in a fast, simple, and cost-effective way.
3. Use a large selection of open source libraries from the Jupyter and IPython ecosystem for analysis, visualization and machine learning.



What is Google Cloud Datalab?

- Datalab is a tool for data processing and analysis.
- Datalab provides a simple web interface for editing and running scripts using familiar languages, such as Python and SQL.
- These scripts are deployed within **notebook documents** or "**notebooks**."

Datalab: What is a notebook?

Notebooks are documents that contain:

- Computer code
- Human readable rich-text elements
- Equations
- Graphs
- Data visualizations

For humans, there are descriptions of data, analysis, code, and results, such as text, figures, and tables.

For computers, there are executable computer files that can be run to perform data analysis and samples of data.

Notebooks are easily shared with others.

Take a few minutes to explore a gallery of interesting notebooks and TensorFlow notebooks

Datalab: What is a notebook?

Google Cloud Datalab — Programming Language Correlation (autosaved)

Notebook ▾ | Add Code | Add Markdown | Delete | Move Up | Move Down | Run ▾ | Clear ▾ | Reset Session

Navigation | Help

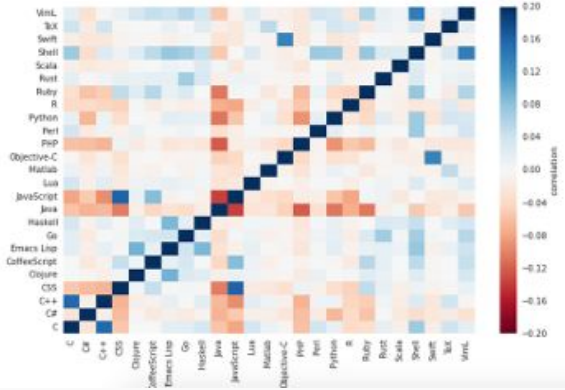
Notebook Outline

BigQuery, SQL + Python, Analysis + Visualization

BigQuery, SQL + Python, Analysis + Visualization

```
%%sql --module commits
SELECT user, language, pushes
FROM (SELECT timeline.actor AS user, timeline.repository_language AS language,
             COUNT(timeline.repository_language) AS pushes
FROM [githubarchive:github.timeline] AS timeline
JOIN (SELECT repository_language AS language, COUNT(repository_language) as pushes
      FROM [githubarchive:github.timeline]
      WHERE type = 'PushEvent' AND repository_language != '' AND
            PARSE_UTC_USEC(created_at) >= DATE_ADD(CURRENT_TIMESTAMP(), -1, 'YEAR')
      GROUP BY language
      ORDER BY pushes DESC
      LIMIT 25) AS languages
ON timeline.repository_language = languages.language
WHERE type = 'PushEvent' AND PARSE_UTC_USEC(created_at) >= DATE_ADD(CURRENT_TIMESTAMP(), -1, 'YEAR')
GROUP BY user, language)
WHERE ABS(HASH(user)) % 100 < 5
```

```
commits.to_dataframe().pivot(index='user', columns='language', values='pushes').fillna(0)
      .corr(method = 'spearman').plot()
```



Datalab - Features

Choose your language

Write code in multiple languages: Python, SQL and JavaScript.

Notebooks

Launch notebooks to explore, transform and process data on Google Cloud Platform or locally.

Fully Integrated

It leverages the power of Cloud Storage, BigQuery, Cloud DataStore and Cloud SQL for analyses.

Built on Jupyter

Built on IPython/Jupyter which already has a thriving ecosystem of modules and a huge knowledge base.

Datalab - Benefits

Increase Productivity

Increased productivity through interactive tools and availability of third party libraries.

Simplicity

Explore and analyze data with ad hoc queries and visualizations.

Collaboration

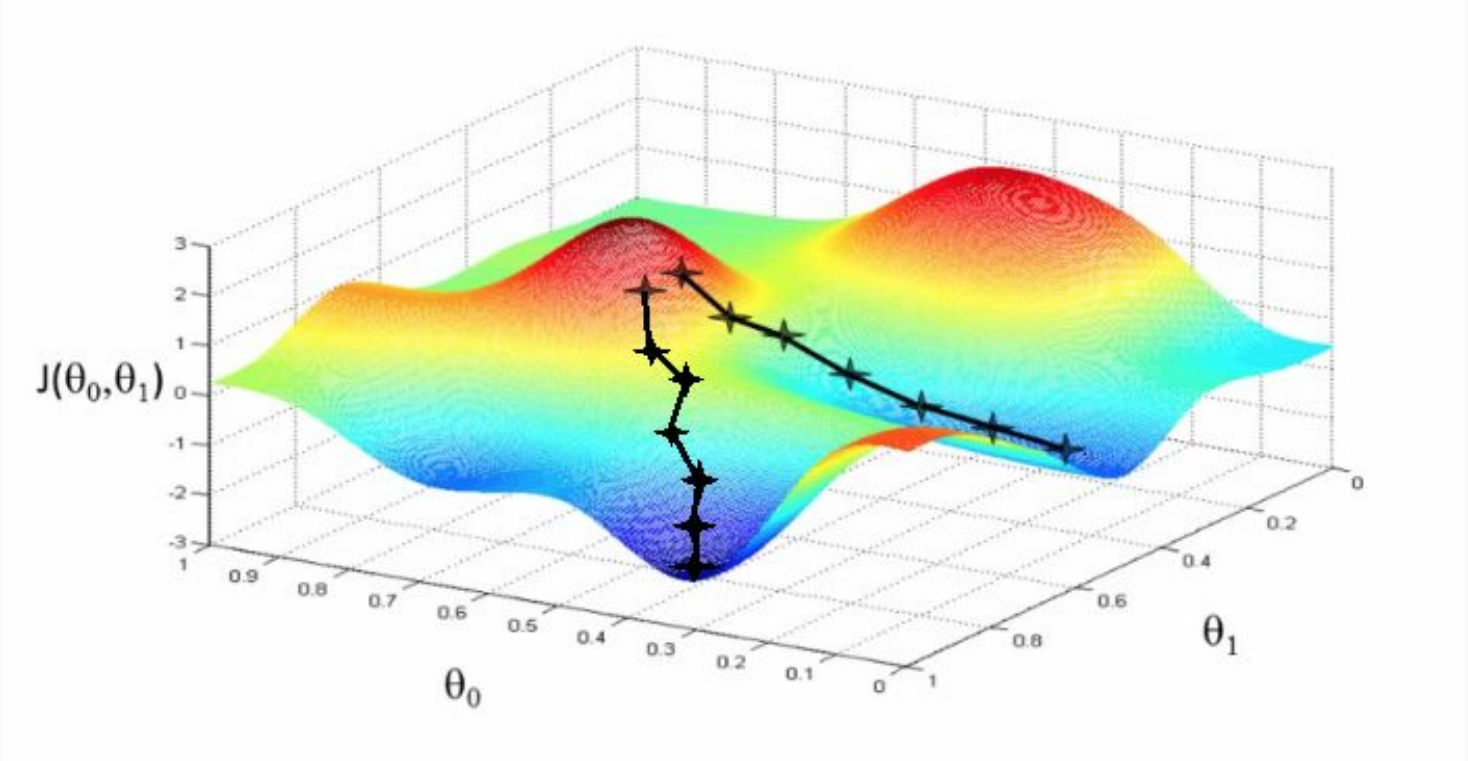
Explore, transform and process data collaboratively or publish data as reports, dashboards or APIs.

Reach

Makes Google's Big Data capabilities easier to use and therefore more accessible across the company.

What is a data visualization?

You'll find many data visualizations in notebooks. This example shows a *gradient descent* that adjusts the parameters gradually to reduce errors (the blue area).



How does Google Cloud Datalab provide value?

Datalab is commonly used for data cleansing and transformation, numerical simulation, statistical modeling, machine learning, data analysis and other data processing jobs. Examples:

- Exploring and analyzing data in Google BigQuery or Google Cloud Storage using standard libraries and SQL including user defined functions
- Getting started with TensorFlow and Google Cloud Machine Learning
- Visualizing query results using powerful visualization packages, such as Google Charts or Plotly or matplotlib
- Combining the power of Google's Cloud Machine Learning and Cloud Storage products in a single tool, eliminating the need for product integration
- It integrates with Google Cloud Platform for things such as authentication and code source control

Datalab is better than Jupyter

Datalab is based on Jupyter but adds considerable integration with Google Cloud Platform, specifically:

- Authentication and authorization
- Google Cloud Platform project integration
- Support for, Google BigQuery, Google Cloud Machine Learning, and Google Cloud Storage
- And Google provides a massive infrastructure most companies could not duplicate. We've taken care of the "plumbing" tasks .

Google Cloud Datalab provides business value

- If you employ data scientists, Datalab helps you be more productive by providing you an environment where you can experiment and analyze large datasets in a simple and powerful way. It simplifies data analysis and transformation.
- If you employ developers with Python and SQL skills, Datalab helps them get started and learn the ropes of data science and machine learning much faster.

Use case 1: get started with Big Data

User issue:
"As a Developer working on a small team, I'm often entrusted with the task of data analysis. I want a simple way to understand what I can do with the data and a roadmap for implementing the analyses. If tools and services connect the dots for me and show some interesting results, I can iteratively build a more useful data pipeline. I can be a part-time Data Analyst or Data Scientist, and help my company while building new skills."

Enabling technology

- A catalog of common plug-and-play analyses and visualizations
- Cloud project integration—easy way to launch notebooks and authentication
- Connectors for Google Cloud Storage, BigQuery, DataStore, and Cloud SQL to bring data into a notebook
- An end-to-end, tool and service-enabled story for going from logs and other semi-structured data to structured analyses and visualizations

Use case 2: analyze and visualize data

User issue:
“As a data analyst, I want to understand how data varies as a function of time, geolocation and other common variables. I want to be able to chart it to understand it and communicate the meaning to others on the team.”

Enabling technologies

- Rich visualization library support
- Extensible visualization support for third party and custom visualization
- Built-in support for time-series and geospatial analysis and rendering
- Visualizing user segmentation and conversion funnels

Use case 3: build machine learning models

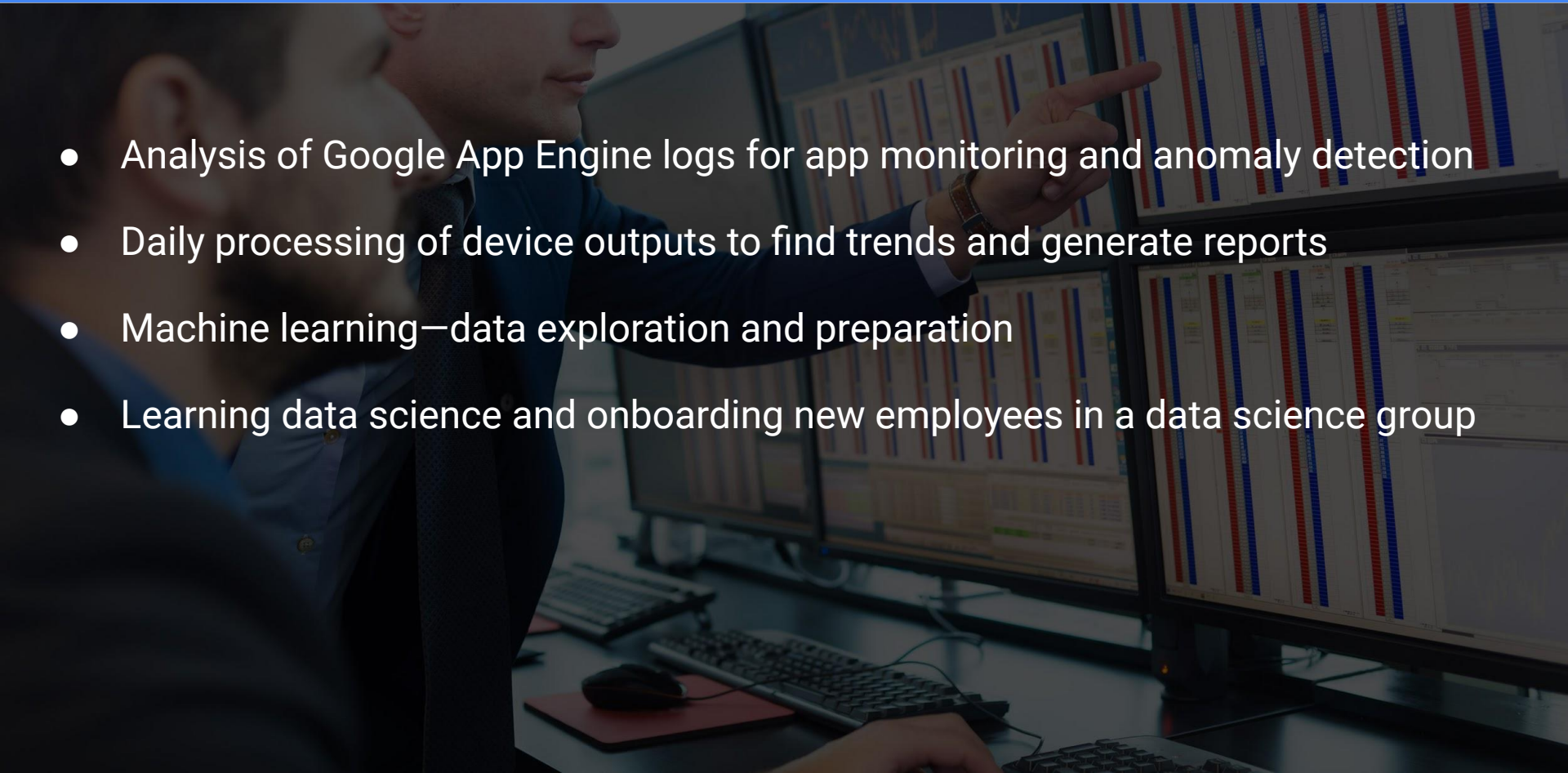
User issue

"As a developer or data analyst, I want access to capabilities that have traditionally required Machine Learning experts. I want to get a jump on building basic prediction models and use text analysis on user-generated content such as reviews and comments without starting from scratch or sitting down with a text book."

Enabling technology

- A library of learning algorithms
- An accessible toolkit or service for building models based on training data and evaluating models based on new data
- Interactive experience for model evaluation and iterative refinement
- Deployment of the model as a service

What interesting work are customers doing?



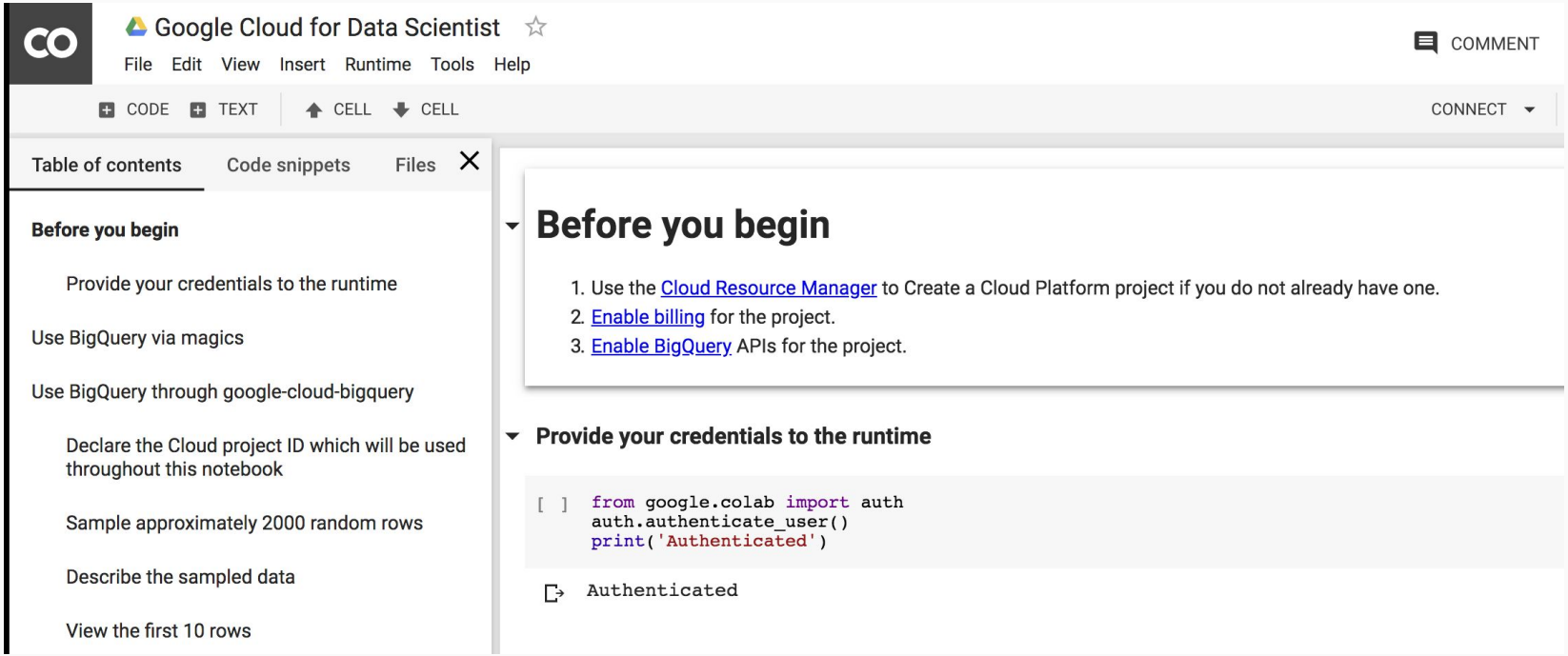
- Analysis of Google App Engine logs for app monitoring and anomaly detection
- Daily processing of device outputs to find trends and generate reports
- Machine learning—data exploration and preparation
- Learning data science and onboarding new employees in a data science group

Datalab - Creating from command line

```
INSTANCE_NAME=datalab-vm
ZONE=europe-west1

datalab create $INSTANCE_NAME --zone $ZONE

datalab connect $INSTANCE_NAME


# memory-intensive machine
datalab create $INSTANCE_NAME --zone $ZONE \
--machine-type n1-highmem-8


# with GPUs
datalab create $INSTANCE_NAME --zone $ZONE \
--accelerator type=nvidia-tesla-k80,count=1
```

```
datalab create  [-h]
      [--image-name IMAGE_NAME]
      [--disk-name DISK_NAME]
      [--disk-size-gb DISK_SIZE_GB]
      [--machine-type MACHINE_TYPE]
      [--no-connect] [--no-backups] [--no-create-repository]
      [--log-level] [--for-user FOR_USER]
      [--service-account SERVICE_ACCOUNT]
      [--port PORT] [--max-reconnects MAX_RECONNECTS]
      [--ssh-log-level] [--no-launch-browser]
      [--project PROJECT] [--quiet] [--verbosity]
      [--zone ZONE]
      NAME
```

Datalab - Colab



Composer

Cloud Composer

Managed Apache Airflow to make workflow creation and management easy, powerful, and consistent.

Composer - Benefits

How Cloud Composer helps you

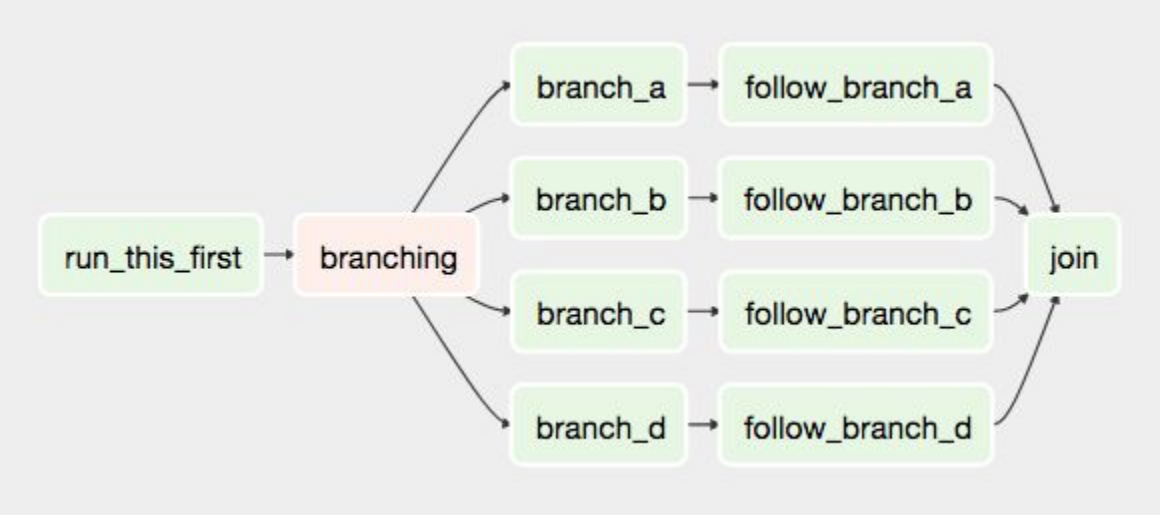
- **Comprehensive GCP integration:** Orchestrate your full GCP pipeline through Cloud Composer
- **Hybrid and multi-cloud environments:** Connect your pipeline and break down silos through a single orchestration service
- **Easy workflow orchestration:** Get moving quickly by coding your workflows using simple Python
- **Open source at its core:** Ensure freedom from lock-in through Apache Airflow's open source portability

Key Cloud Composer features

- Client tooling including the Google Developer Console and Cloud SDK
- Easy and controlled access to the Airflow web UI through Cloud Identity-Aware Proxy
- Streamlined Airflow runtime and environment configuration, such as plugin support
- Stackdriver logging and monitoring
- Identity access management (IAM)
- Simplified DAG (workflow) management
- Python (PyPi) package management

Composer - Airflow DAGs

DAG = Directed Acyclic Graph(Vertex, Edge) where
V = Task | DAG (SubDagOperator)
E = Dependency (e.g., v1 >> v2 : v1 is the upstream dependency of v2)



Understanding Airflow DAGs

Airflow DAGs are Python files with 5 main sections

1. Imports
2. Arguments
3. Setup
4. Tasks
5. Ordering

Imports

Import Python dependencies needed for the workflow.

```
import datetime
from airflow import DAG
from airflow import models
from airflow.contrib.operators import bigquery_operator
from airflow.utils import trigger_rule
```

Arguments

Define default and DAG-specific arguments. Can utilize environment variables and Jinja templating.

```
default_dag_args = {
    # Setting start date as yesterday starts the DAG
    # immediately when it is detected in the Cloud
    # Storage bucket.
    'start_date': yesterday,
    'email_on_failure': False,
    'email_on_retry': False,
    # If a task fails, retry it once after waiting
    # at least 5 minutes
    'retries': 1,
    'retry_delay': datetime.timedelta(minutes=5),
}

bq_dataset_name = 'airflow_bq_dataset_{{ ds_nodash }}'
bq_githib_table_id = bq_dataset_name + '.github_commits'
output_file = 'gs://my_bucket/github_commits.csv'
```

Setup

Give the DAG a name, configure the schedule, and set the DAG settings.

```
dag = DAG(
      'demo_dag',
      # Continue to run DAG once per day
      schedule_interval = datetime.timedelta(days = 1),
      default_args = default_dag_args
)
```

Tasks

Tasks that do two things:

1. Run a BigQuery query that exports the results to a new dataset
2. Exports the new dataset to Cloud Storage

```
# Perform query of Airflow GitHub commits
bq_airflow_commits_query =
bigquery_operator.BigQueryOperator(
      task_id = 'bq_airflow_commits_query',
      bql = """      SELECT commit, subject, message
      FROM [bigquery-public-data:github_repos.commits]
      WHERE repo_name contains 'airflow'
       """,
       destination_dataset_table = bq_githib_table_id
)

# Export query result to Cloud Storage
export_commits_to_gcs =
bigquery_to_gcs.BigQueryToCloudStorageOperator(
      task_id = 'export_airflow_commits_to_gcs',
      source_project_dataset_table = bq_githib_table_id,
      destination_cloud_storage_uris = [output_file],
      export_format = 'CSV'
)
```

Ordering

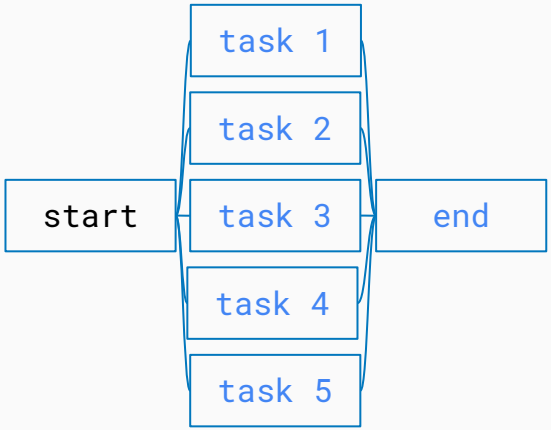
Set the order the tasks should be executed in. Ordering can be simple (all at once, in serial) or complex (graph.)

```
# Define task ordering
bq_airflow_commits_query >> export_commits_to_gcs
```

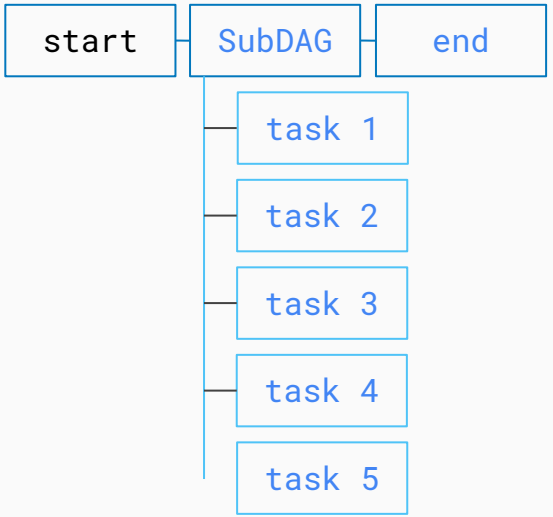
SubDAGs

SubDAGs can be used to simplify and separate DAGs into clean and logical parts.

One DAG, many tasks



One DAG with SubDAG



Composer - Steps

Composer - Operators

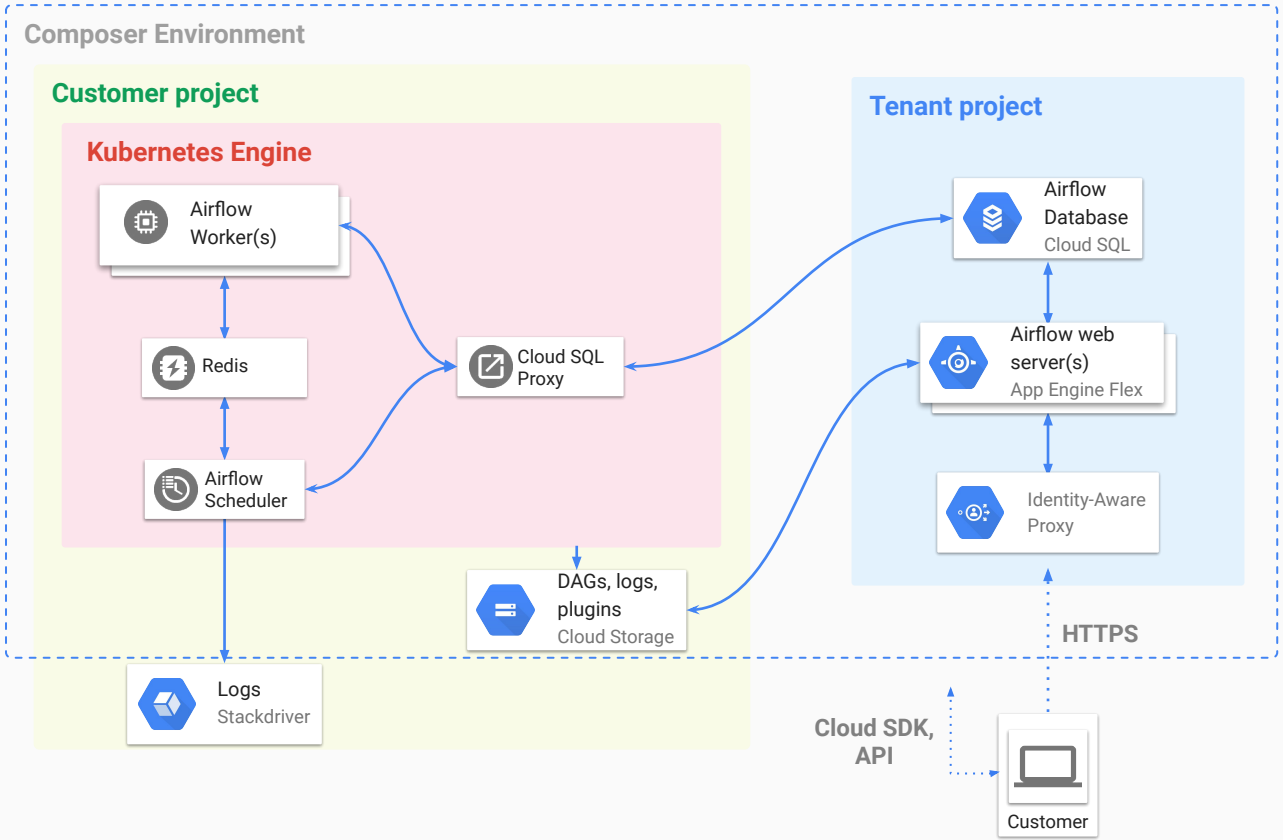
An operator describes a single task in a workflow.

Types:

- Action operator that performs or tells the system to perform an action
- Transfer operator that moves data from one system to another
- Sensor are a special type of operators that keeps running until a certain criteria is met (e.g. availability of data).

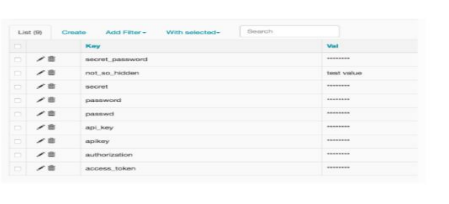
Airflow already supports BigQuery, Dataflow, Dataproc, Datastore, Cloud Storage, Pub/Sub and Cloud ML engine.

Composer - Architecture

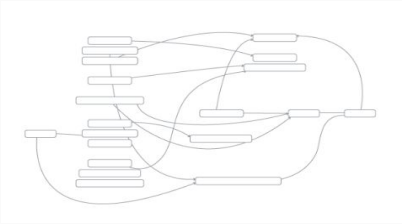


Composer - DAG Views on Airflow UI

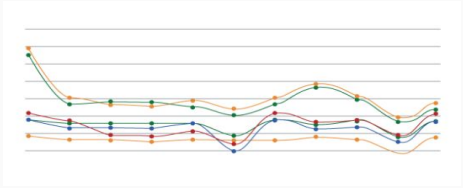
Tree views show a DAG across time, aiding in identifying blockers for late pipelines.



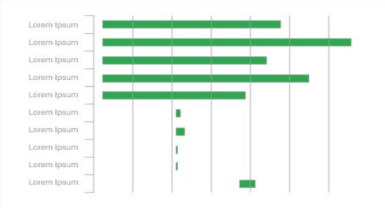
Variable views allow you to inspect the key-value pair of variables used in jobs



Graph views give a comprehensive look at dependencies and status of a work-flow.



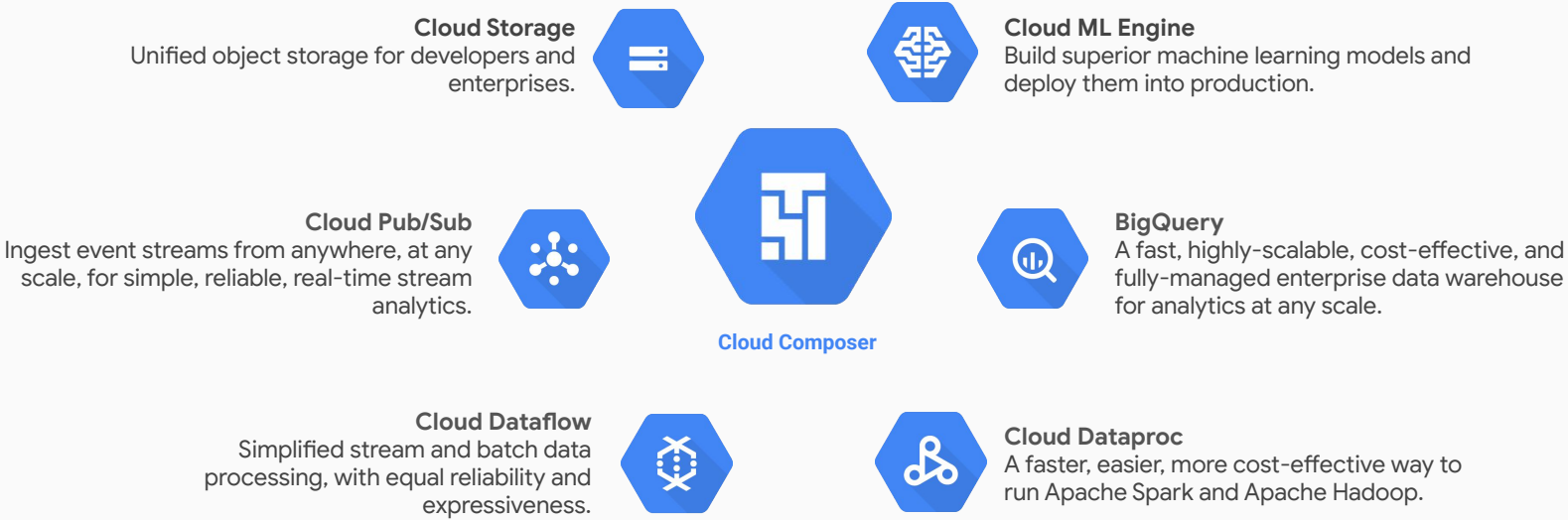
Gantt views allow you to analyze task duration to identify bottlenecks



Gantt views allow you to analyze task duration to spot bottlenecks

Code views give you the clearest view of what is happening within your pipeline

Composer - Integrations

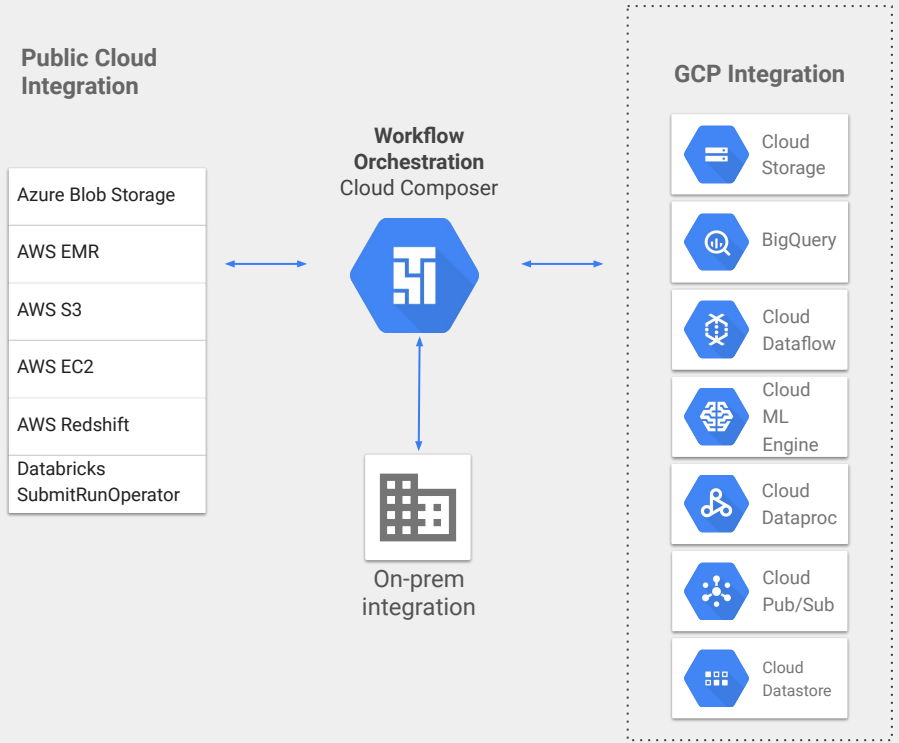


Composer - Connectors

Broad community of development
Diverse needs of the OSS community ensure that connectors are built to a variety of services, with more added on an ongoing basis

Easy-to-use experience
Drop do-it-yourself orchestration between services that hamper future flexibility by using established connectors

Connect data across environments
Take advantage of what you feel are best in breed solutions from a mix of your on-premises and cloud environments



Cloud Composer Data Workflow Orchestration

End-to-End GCP Integration

Author full workflows for GCP with a single tool, Integration with BigQuery, Dataflow, Dataproc, Datastore, Cloud Storage, Pub/Sub, Cloud ML Engine

Hybrid and Multi-Cloud Environments

Break down data silos and unite data sources in one tool, Connect data pipeline between different clouds and different on-premises tools

Easy Workflow Orchestration

Author workflows in Python, One-click deployment and auto-synchronization, Rich library of connectors, Easy troubleshooting and increased reliability through graph-based (DAG) UI

Open Source at its Core

Built upon Apache Airflow, Freedom from lock-in, Community continuing contributions for non-GCP platforms GCP contributing back our developments

Data Fusion

Cloud Data Fusion

Cloud Data Fusion is a fully-managed, cloud native, enterprise data integration service for quickly building and managing data pipelines.

Data Fusion - CDAP

Cloud Data Fusion is Google's cloud native manage service for enterprise data integration powered by CDAP.

CDAP

cdap.io

Google acquired Cask - May 14th 2018

CDAP is an 100% open-source framework to build data analytics applications for on-premise and cloud.

API | Runtime | Pre-built Accelerators | Library | Apache 2.0 License | In production at scale for +4.5 years / Dev +7 years

Developer, Data scientist and Business Analyst

1

Need to cleanse, match, de-dupe, blend, transform, partition, transfer, standardize, automate, & monitor

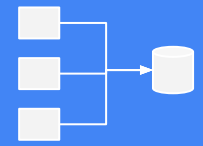
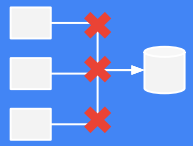
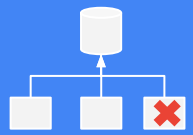
2

Use Cloud Data Fusion to visually build integration pipeline, test, debug and deploy

3

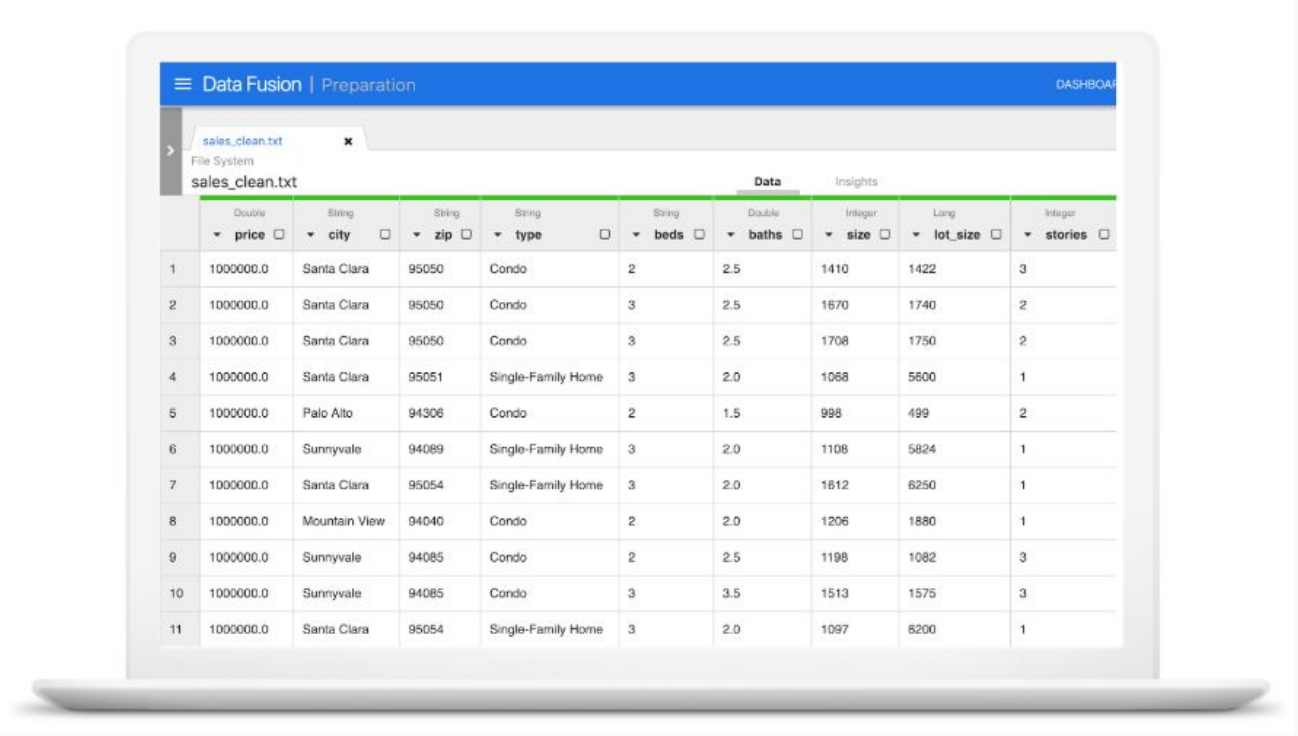
Run it at scale on GCP, operationalize (monitor, report) pipelines, inspect rich integration metadata

Data Fusion

Business Imperatives	Improve Decisions & Regulatory Compliance	Modernize Business and Reduce IT costs	Acquire and Merge	Increase Business Profitability	Extend or Migrate or both to Cloud
IT Initiatives	Business Intelligence / Analytics / AI	Legacy Retirement	Data and Application Consolidation	Customer, Supplier, Product Hubs	iPaaS, SaaS
Data Fusion Use cases	 Data Warehouse	 Data Migration	 Data Consolidation	 Master Data Management	 Data Consistency
Data Services					

Data Fusion - Benefits

- Integrate with any data
- Increase productivity
- Reduce complexity
- Increase flexibility



	price	city	zip	type	beds	baths	size	lot_size	stories
1	1000000.0	Santa Clara	95050	Condo	2	2.5	1410	1422	3
2	1000000.0	Santa Clara	95050	Condo	3	2.5	1670	1740	2
3	1000000.0	Santa Clara	95050	Condo	3	2.5	1708	1750	2
4	1000000.0	Santa Clara	95051	Single-Family Home	3	2.0	1068	5600	1
5	1000000.0	Palo Alto	94306	Condo	2	1.5	998	499	2
6	1000000.0	Sunnyvale	94089	Single-Family Home	3	2.0	1108	5824	1
7	1000000.0	Santa Clara	95054	Single-Family Home	3	2.0	1612	6250	1
8	1000000.0	Mountain View	94040	Condo	2	2.0	1206	1880	1
9	1000000.0	Sunnyvale	94085	Condo	2	2.5	1198	1082	3
10	1000000.0	Sunnyvale	94085	Condo	3	3.5	1513	1575	3
11	1000000.0	Santa Clara	95054	Single-Family Home	3	2.0	1097	6200	1

Data Fusion - Development

Rich graphical interface

100+ plugins - connectors, transforms & actions

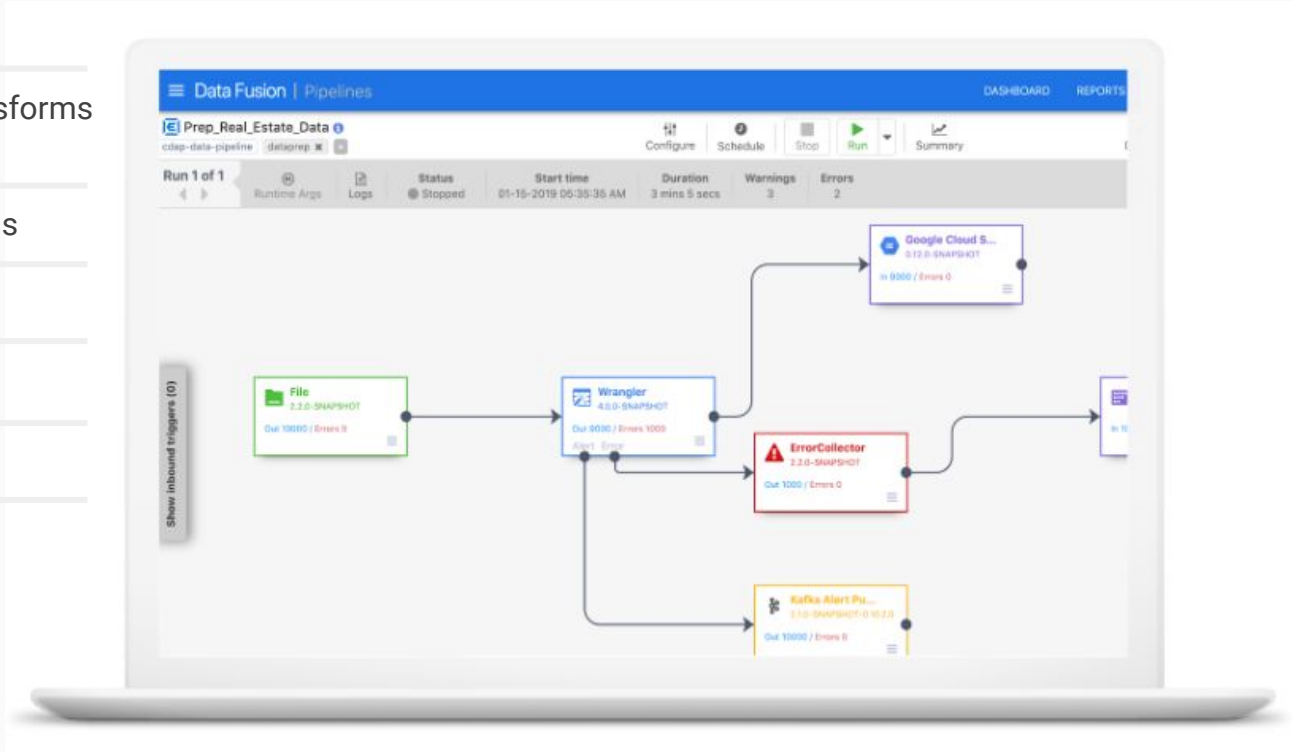
Code free visual transformations

1000+ transforms, data quality

Test and debug pipeline

Pre-built pipelines

Developer SDK



Data Fusion - Execution & Management

Cloud Dataproc - Batch & Realtime

Schedules and Triggers

Control and data flow pipelines support

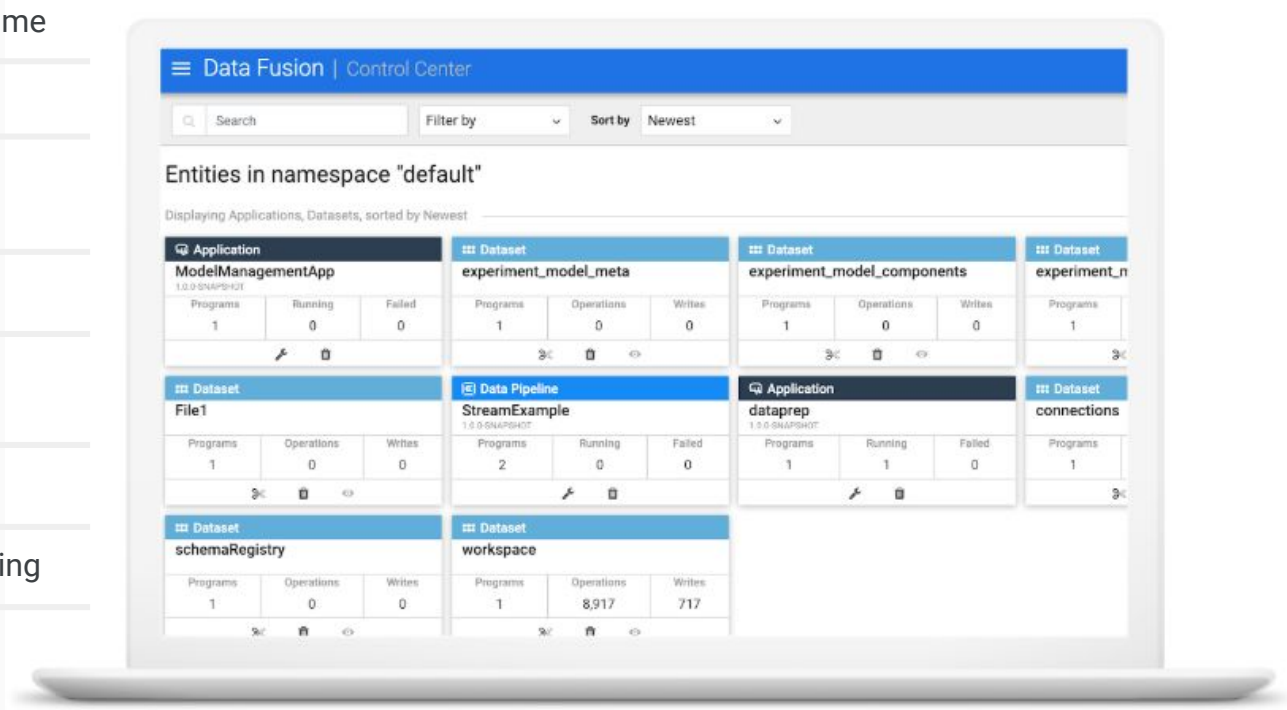
Dashboard and Reports

Execute existing **Spark** and **MapReduce** jobs

Private IP support

Service Networking - VPC Peering

REST API and CLI



Data Fusion - Integration Metadata

Tags and Properties support

Pipeline

Dataset

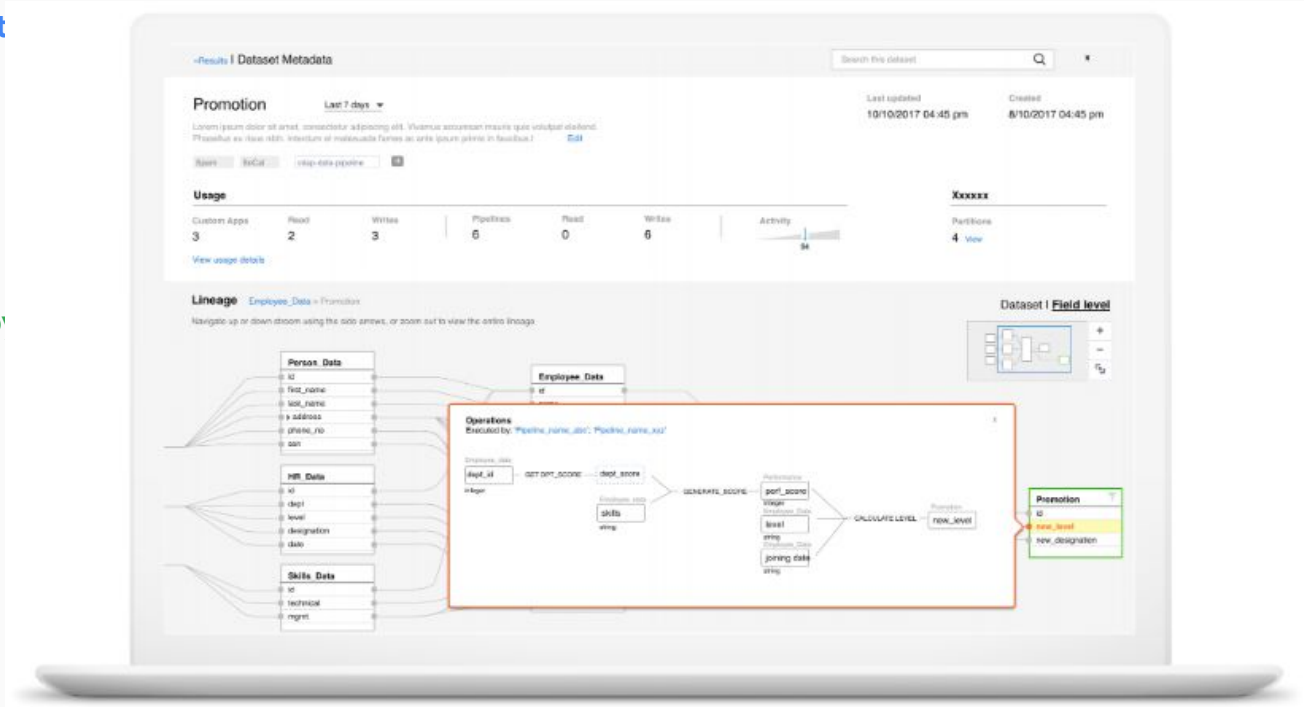
Schema

Search integrated entities b

Keyword

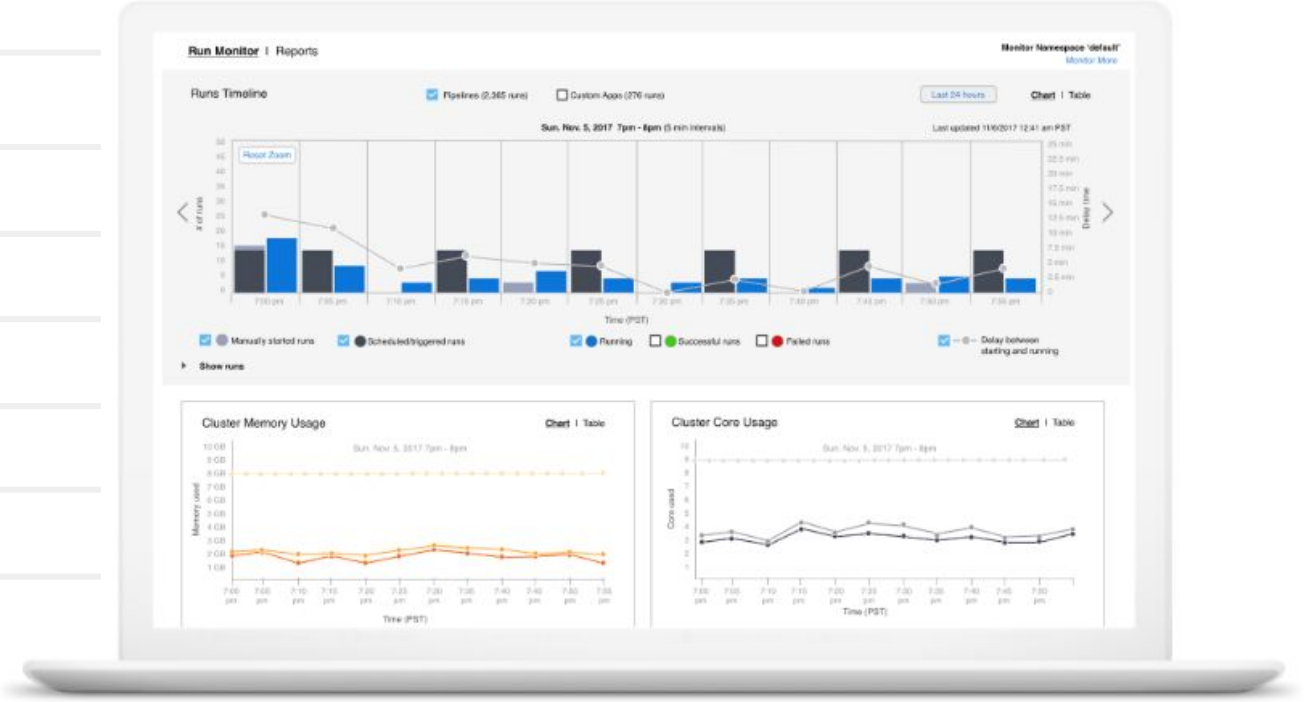
Schema name and type

Dataset level and Field level Lineage



Data Fusion - Extensible

- Pipeline templatization
- Conditional Pipeline Triggers
- Plugin Management
- Plugin templatization
- Plugin UI Widget
- Custom Provisioners
- Custom Compute Profiles
- Hub Integration



Data Catalog

Data Catalog is a fully managed and highly scalable data discovery and metadata management service

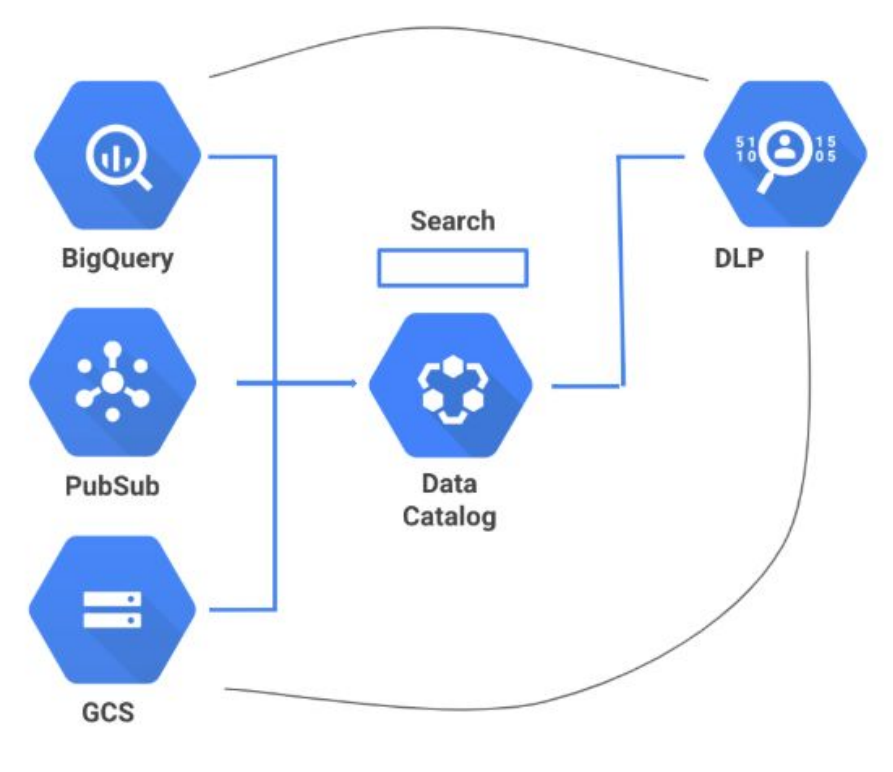
Organizations faced with a wealth of data spread across disjoint systems need an **effective solution for data discovery**

Offers **unified data discovery** of all data assets, spread across multiple projects and systems

Empowers users to **annotate business metadata** in a collaborative manner

Provides the **foundation** for data **governance**

Data Catalog feature highlights



1. Provides a simple search interface for data discovery
2. Supports UI and API for all metadata operations
3. Supports business metadata through schematized tags
4. Auto-ingests technical metadata from GCP data assets
5. Enforces ACL controls on metadata
6. Auto-tags PII data through DLP integration

Data Catalog - Architecture

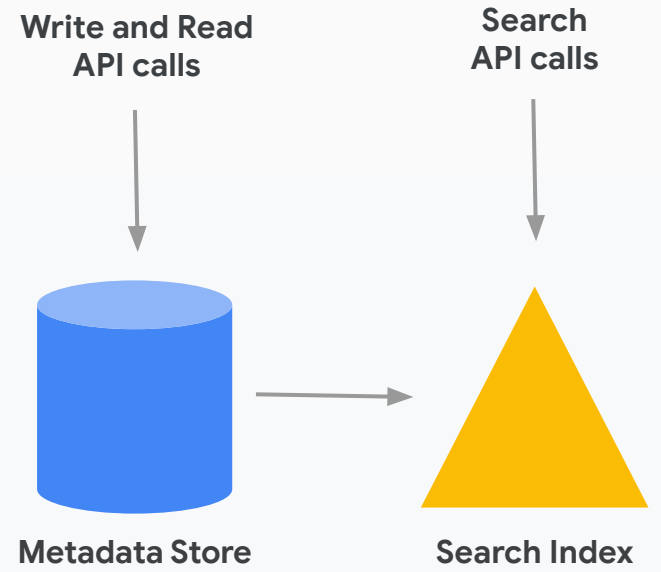
Data Catalog leverages the architecture and innovation of Google's internal metadata management service that is in wide use for many years.

Spanner, the globally distributed, strongly consistent database for storing all metadata entries

Real-time and batch syncers for auto-ingestion of technical metadata

Google search index with built-in ACL checks for data discovery -

Leverages the same technology that powers **Gmail** and **Google Drive**



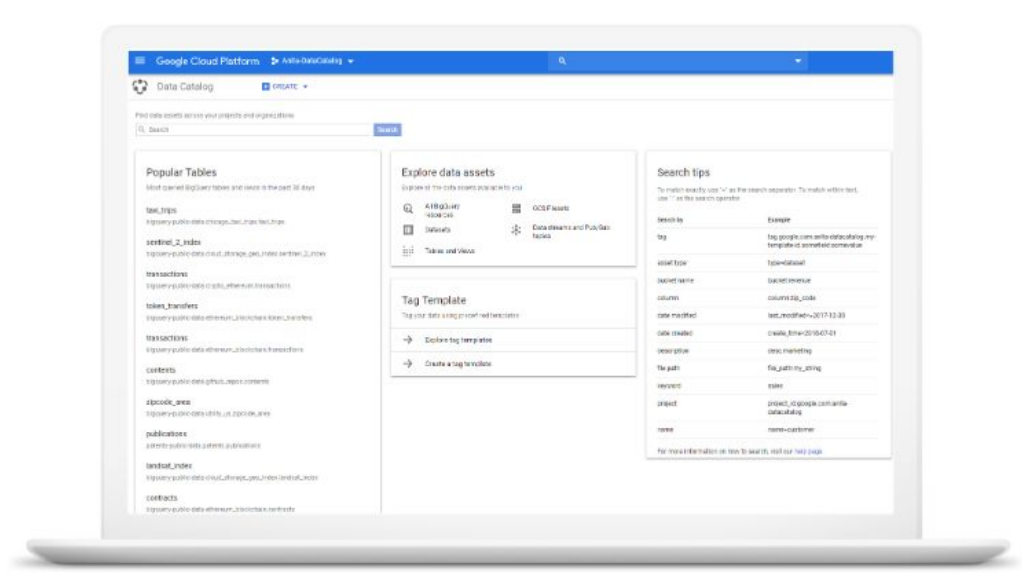
Data Catalog - Data Discovery

UI enables search, read, and write access **to all metadata**

Simple keyword search interface enables both, business and technical users

Facet search enables power users

type:view
column:keyword
tag:approved_for_use: true
tag:data_classification:confidential
tag:has_pii:true
tag:type_pii:ssn



Programmatic access through API

Read, Write, and Search API for full metadata access

API powers bulk metadata updates

- Beta API with Python, Java, and Node.js language libraries

API enables Enterprise Applications and Custom Frontends

- GOJEK and others

```
# Create Tag Template
template = datacatalog_v1beta1.types.TagTemplate()
template.display_name = 'A test for v1beta1 using only primitive types'

template.fields['a-boolean'].display_name = 'BOOL field'
template.fields['a-boolean'].type.primitive_type = \
    datacatalog_v1beta1.enums.FieldType.PrimitiveType.BOOL.value

template.fields['a-double'].display_name = 'DOUBLE field'
template.fields['a-double'].type.primitive_type = \
    datacatalog_v1beta1.enums.FieldType.PrimitiveType.DOUBLE.value

template.fields['a-string'].display_name = 'STRING field'
template.fields['a-string'].type.primitive_type = \
    datacatalog_v1beta1.enums.FieldType.PrimitiveType.STRING.value

template.fields['a-timestamp'].display_name = 'TIMESTAMP field'
template.fields['a-timestamp'].type.primitive_type = \
    datacatalog_v1beta1.enums.FieldType.PrimitiveType.TIMESTAMP.value

template.fields['an-enum'].display_name = 'ENUM field'
template.fields['an-enum'].type.enum_type.allowed_values.append('VALUE_A')
template.fields['an-enum'].type.enum_type.allowed_values.append('VALUE_B')

client.create_tag_template(
    parent=client.project_path('YOUR_PROJECT_ID'),
    tag_template_id='YOUR_TEMPLATE_ID',
    tag_template=template)

# Create Tags
# ================================================================
tag = datacatalog_v1beta1.types.Tag()
tag.template = client.tag_template_path('YOUR_PROJECT_ID', 'YOUR_TEMPLATE_ID')
tag.fields['a-boolean'].bool_value = True
tag.fields['a-double'].double_value = 3.1415
tag.fields['a-string'].string_value = 'tag-string'
tag.fields['a-timestamp'].timestamp_value.FromJsonString('2019-03-29T15:45:00-07:00')

bq_dataset_entry_id = 'YOUR_BQ_DATASET_ENTRY_ID'
client.create_tag(parent=client.entry_path('YOUR_PROJECT_ID', 'global', '@bigquery', bq_dataset_entry_id), tag=tag)
```

Technical and Business Metadata

Technical Metadata (auto-ingested from data source)

- Table names, column names
- Table descriptions, column descriptions
- Date created, date modified

Business Metadata (user provided / inferred)

- Table has PII
- Data quality owner
- ‘Delete by’ date
- ‘Retain till’ date
- Business logic used for computing a column
- Data quality score

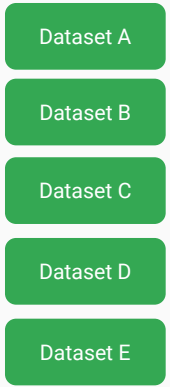
Metadata access governed by IAM

User-1 permissions

- read access to all datasets in BQ

User-1 view in Data Catalog

- Discover all datasets
- Read access to all datasets

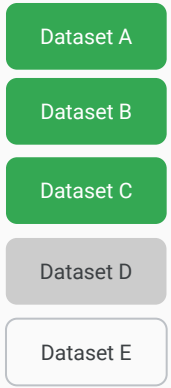


User-2 permissions

- read access to A, B, and C
- metadata-read only access to dataset D
- no read or metadata-read access to dataset E

User-2 view in Data Catalog

- Can discover datasets A, B, C, and D
- Discovers dataset D but needs to request read access to dataset D
- No metadata exfiltration for dataset E



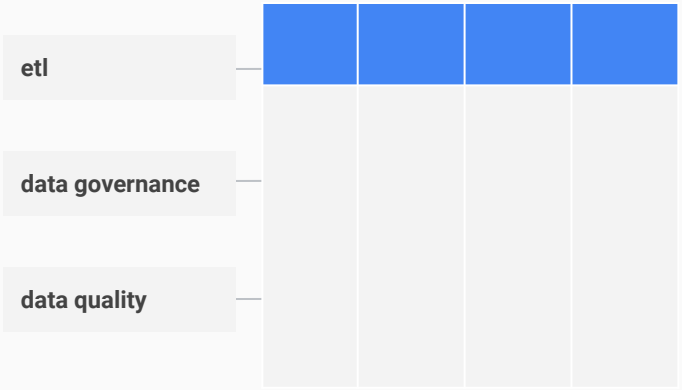
ACL controls on Business Metadata

Data governor permissions

- access to all tags

Data Governor view in Data Catalog

- can discover all tags

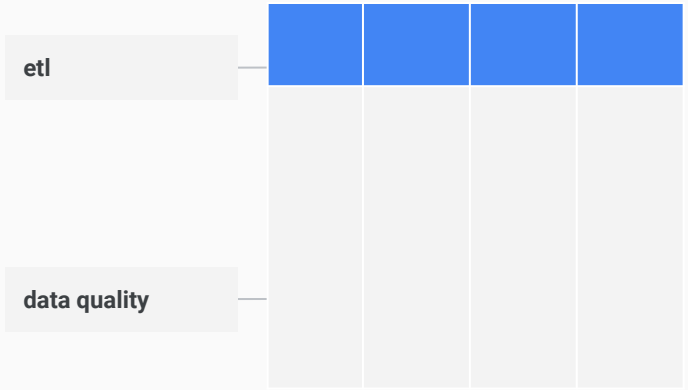


Data Analyst permissions

- no access to ‘data governance’ tags

Data Analyst view in Data Catalog

- can discover all accessible tags



¿ Preguntas ?

Ismael Yuste
linkedin.com/in/ismaelyuste/
@IsmaelYuste