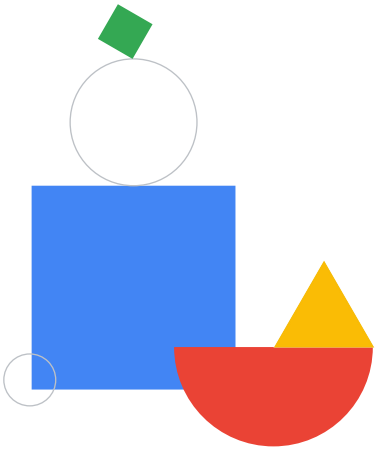
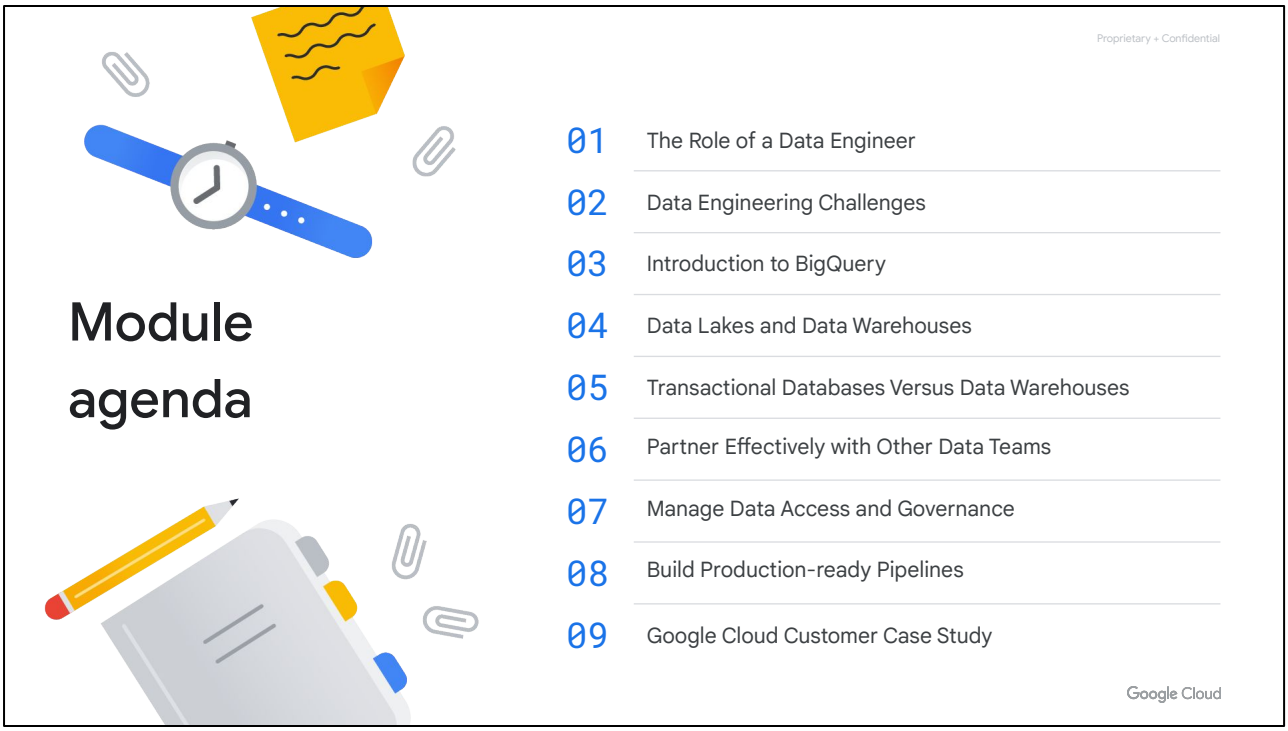
Google Cloud

Introduction to Data Engineering



In this **Introduction to Data Engineering** module, we'll describe the role of a data engineer and motivate the claim why data engineering should be done in the Cloud.



A data engineer is someone who builds data pipelines, and so we'll start by looking at what this means -- what kinds of pipelines a data engineer builds and their purpose.

We'll look at the challenges associated with the practice of data engineering and how many of those challenges are easier to address when you build your data pipelines in the cloud.

Next, we'll introduce you to BigQuery, Google Cloud's petabyte-scale serverless data warehouse.

Having defined what data lakes and data warehouses are, we'll then discuss these in more detail.

Data Engineers may be responsible for both the backend transactional database systems that support a company's applications and the data warehouses that support their analytic workloads. In this lesson, we'll explore the differences between databases and data warehouses and the Google Cloud solutions for each of these workloads.

Since a data warehouse also serves other teams, it is crucial to learn how to partner effectively with them.

As part of being an effective partner, your engineering team will be asked to set up data access policies and overall governance of how data is to be used and NOT used

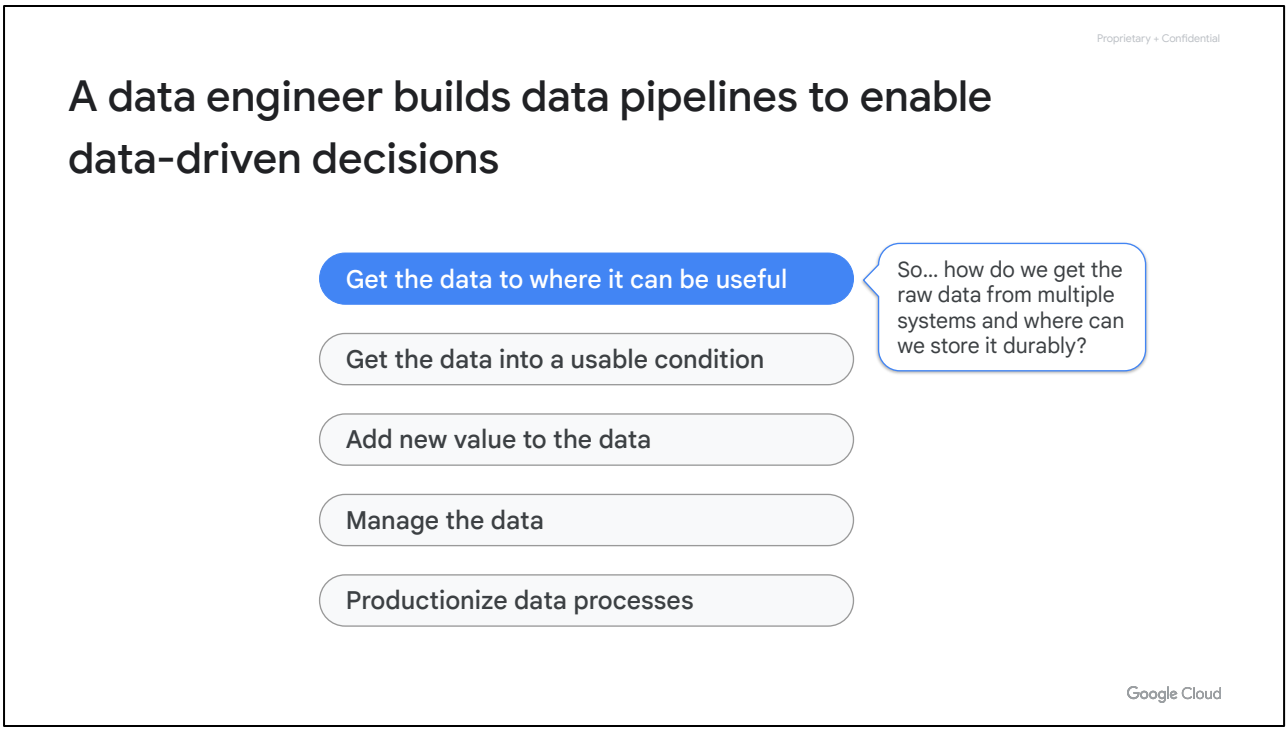
by your users. We'll discuss how to provide access to the data warehouse while keeping to data governance best practices.

We'll also discuss productionizing the whole operation and automating and monitoring as much of it as possible.

Finally, we'll look at a case study of how a Google Cloud customer solved a specific business problem, before you complete a hands-on lab where you will use BigQuery to analyze data.

The Role of a Data Engineer

Let’s start by exploring the role of a data engineer in a little more detail.

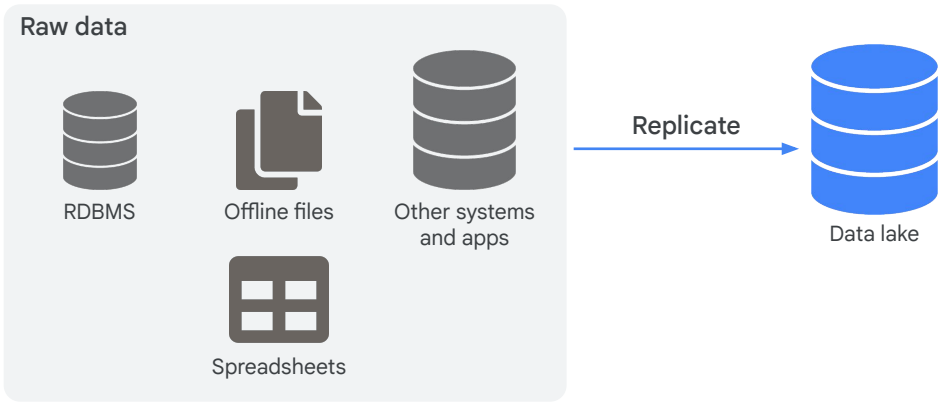


What does a data engineer do? A data engineer builds data pipelines.

Why does the data engineer build data pipelines? Because they want to get their data into a place, such as a dashboard or report or machine learning model, from where the business can make data-driven decisions.

The data has to be in a usable condition so that someone can use this data to make decisions. Many times, the raw data is by itself not very **useful**.

A data lake brings together data from across the enterprise into a single location



One term you will hear a lot when you do data engineering is the concept of a data lake.

A data lake brings together data from across the enterprise into a single location.

So, you might get the data from a relational database or from a spreadsheet, and store the raw data in a data lake.

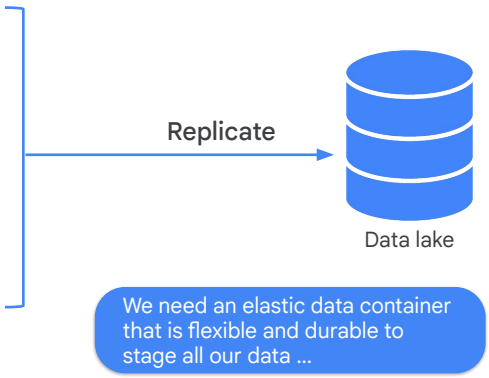
One option for this single location to store the raw data is to store it in a Cloud Storage bucket.

What are the key considerations when deciding between data lake options? What do you think?

https://cloud.google.com/solutions/build-a-data-lake-on-gcp#cloud-storage-as-data-lake

Key considerations when building a data lake

1. Can your data lake handle all the types of data you have?
2. Can it scale to meet the demand?
3. Does it support high-throughput ingestion?
4. Is there fine-grained access control to objects?
5. Can other tools connect easily?

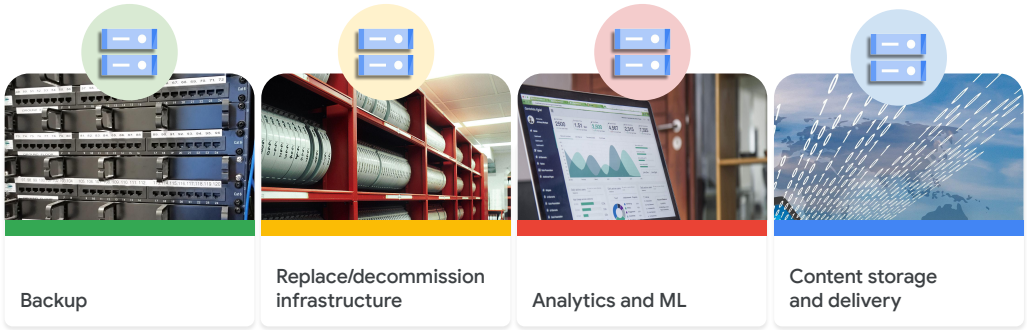


There are some considerations that you need to keep in mind as you build a data lake.

1. Does your data lake handle all the types of data you have? Can it all fit into a cloud storage bucket? If you have a RDBMS, you might need to put the data in Cloud SQL, a managed database, rather than cloud storage.
2. Can it elastically scale to meet the demand? As your data collected increases, will you run out of disk space? This is more a problem with on-premises systems than with cloud.
3. Does it support high-throughput ingestion? What is the network bandwidth? Do you have edge points of presence?
4. Is there fine-grained access control to objects? Do users need to seek within a file? Or is it enough to get a file as a whole? Cloud Storage is blob storage, so you might need to think about the granularity of what you store.
5. Can other tools connect easily? How do they access the store? Don't lose sight of the fact that the purpose of a data lake is to make data accessible for analytics.

https://cloud.google.com/solutions/build-a-data-lake-on-gcp#cloud-storage-as-data-lake

Cloud Storage is designed for 99.999999999% annual durability



Quickly create buckets with Cloud Shell
`gsutil mb gs://your-project-name`

We mentioned our first Google Cloud product, the Cloud Storage bucket, which is a good option for staging all of your raw data in one place before building transformation pipelines into your data warehouse.

Why choose Google Cloud storage? Commonly, businesses use Cloud Storage as a backup and archival utility for their businesses. Because of Google's many data center locations and high network availability, storing data in a Cloud Storage bucket is durable and performant.

For a data engineer, you will often use a cloud storage bucket as part of your data lake to store many different raw data files, such as CSV, JSON, or Avro. You could then load or query them directly from BigQuery as a data warehouse.

Later in the course, you'll create Cloud Shell buckets using the Cloud Console and command line like you see here. Other Google Cloud products and services can easily query and integrate with your bucket once you've got it setup and loaded with data.

What if your data is not usable in its original form?

Extract, **T**ransform, and **L**oad

Data processing

Dataproc Dataflow

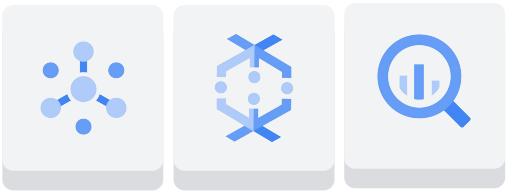
Speaking of loading data, what if your raw data needs additional processing?

You may need to extract the data from its original location, transform it, and then load it in.

One option is to carry out data processing. This is often done using Dataproc or Dataflow. We'll discuss using these products to carry out batch pipelines later in this course.

What if your data arrives continuously and endlessly?

Streaming data processing



But what if batch pipelines are not enough? What if you need real-time analytics on data that arrives continuously and endlessly?

In that case, you might receive the data in Pub/Sub, transform it using Dataflow and stream it into BigQuery.

We'll discuss streaming pipelines later in this course.

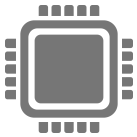
Data Engineering Challenges

Let's look at some of the challenges that a data engineer faces.

Common challenges encountered by data engineers

Access to data

Data accuracy and quality



Availability of computational resources

Query performance

As a data engineer, you will usually encounter a few problems when building data pipelines.

You might find it difficult to access the data that you need. You might find that the data, even after you **access it,** …doesn't have the **quality** that's required by the analytics or machine learning model.

You plan to build a model, and even if the data quality exists, you might find that the transformations require computational resources that **might not be available** to you.

And finally, you might run into challenges around **query performance** and being able to run all of the queries and all of the transformations that you need with the computational resources that you have.

Challenge: Consolidating disparate datasets, data formats, and manage access at scale

Access to data

Data accuracy and quality

Availability of computational resources

Query performance

Let’s take the first challenge of consolidating disparate datasets, data formats, and managing access at scale.

Getting insights across multiple datasets is difficult without a data lake

Data is scattered across Google Analytics 360, CRM, and Campaign Manager products, among other sources.

Customer and sales data is stored in a CRM system.

No common tool exists to analyze data and share results with the rest of the organization.

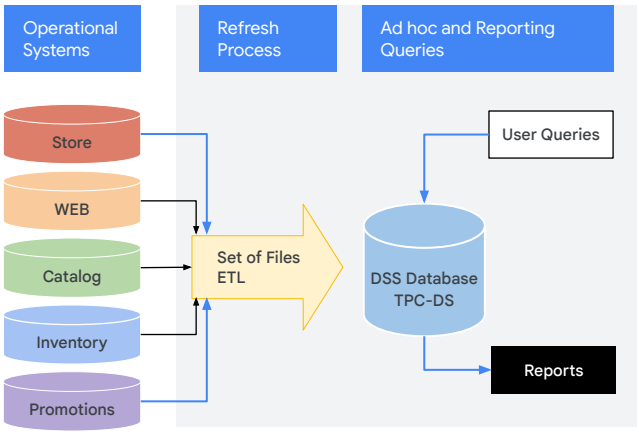
Some data is not in a queryable format.

For example, you want to compute the customer acquisition cost: how much does it cost in terms of marketing and promotions and discounts to acquire a customer?

That data might be scattered across a variety of marketing products and customer relationship management software, … and finding a tool that can analyze all of this data might be difficult because it might come from different organizations, different tools, and different schemas, and maybe some of that data is not even structured.

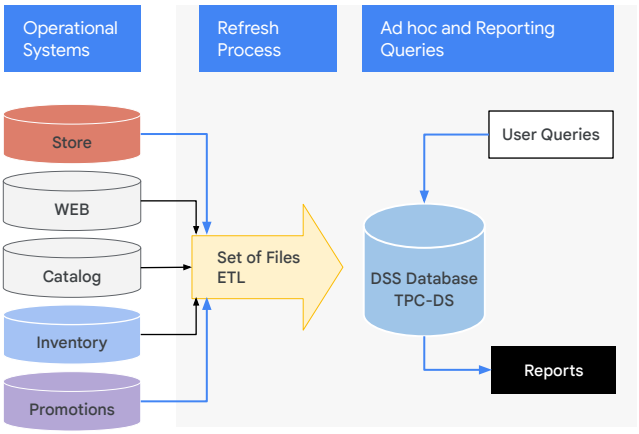
So in order to find something as essential to your business as how much getting a new customer costs, so that you can figure out what kind of discounts to offer to keep them from turning, you can't have your data exist in silos.

Data is often siloed in many upstream source systems



So what makes data access so difficult? Primarily, this is because data in many businesses is siloed by departments and each department creates its own transactional systems to support its own business processes.

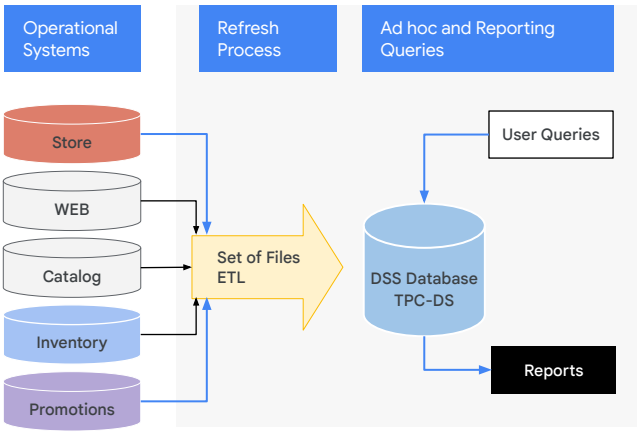
Data is often siloed in many upstream source systems



So, for example, you might have operational systems that correspond to store systems, have a different operational system maintained by your product warehouses that manages your inventory, and have a marketing department that manages all the promotions given that you need to do an analytics query on, such as, …

Data is often siloed in many upstream source systems

Example query:
Give me all the in-store promotions for recent orders and their **inventory levels.**

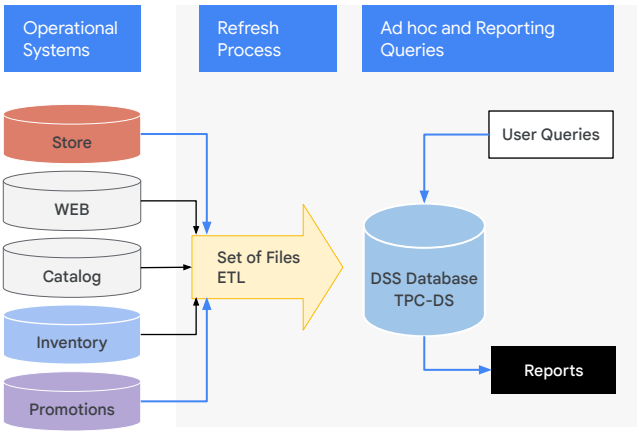


“Give me all the in-store promotions for recent orders and their inventory levels.”

Data is often siloed in many upstream source systems

Example query:
Give me all the in-store promotions for recent orders and their **inventory levels.**

Stored in a separate system and restricted access



You need to know how to combine data from the stores, from the promotions, and from the inventory levels, and because these are all stored in separate systems—some of which have restricted access—building an analytic system that uses all three of these data sets to answer an ad hoc query like this can be very difficult.

Challenge: Cleaning, formatting, and getting the data ready for useful business insights in a data warehouse

Access to data

Data accuracy and quality

Availability of computational resources

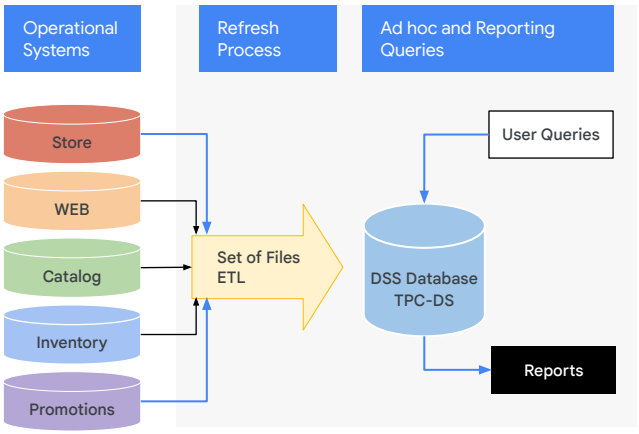
Query performance

The second challenge is that cleaning, formatting and getting the data ready for insights requires you to build ETL pipelines.

ETL pipelines are usually necessary to ensure data accuracy and quality. The cleaned and transformed data are typically stored, not in a data lake, but in a data warehouse.

A data warehouse is a consolidated place to store the data, and all the data are easily joinable and queryable. Unlike a data lake, where the data is in the raw format, in the data warehouse, the data is stored in a way that makes it efficient to query.

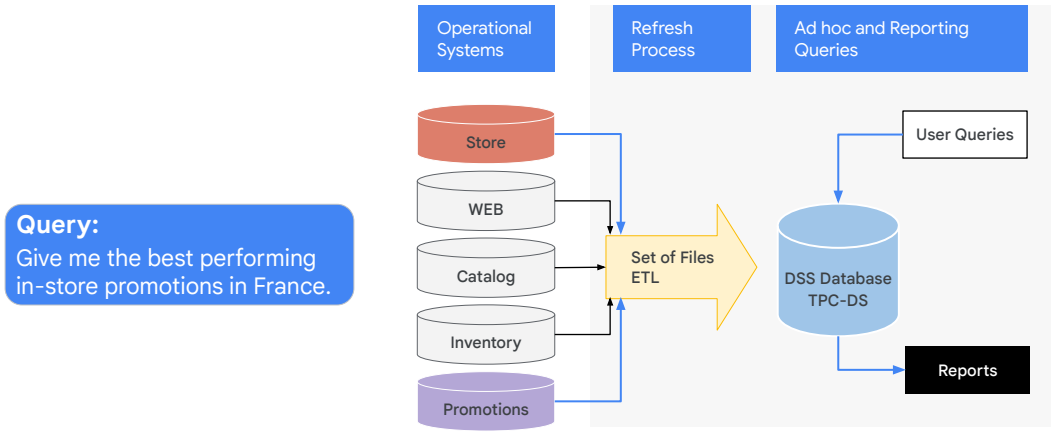
Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



Because data becomes useful only after you clean it up, you should assume that any raw data that you collect from source systems needs to be cleaned and transformed. And if you are transforming it, you might as well transform it into a format that makes it efficient to query. In other words, ETL the data and store it in a data warehouse.

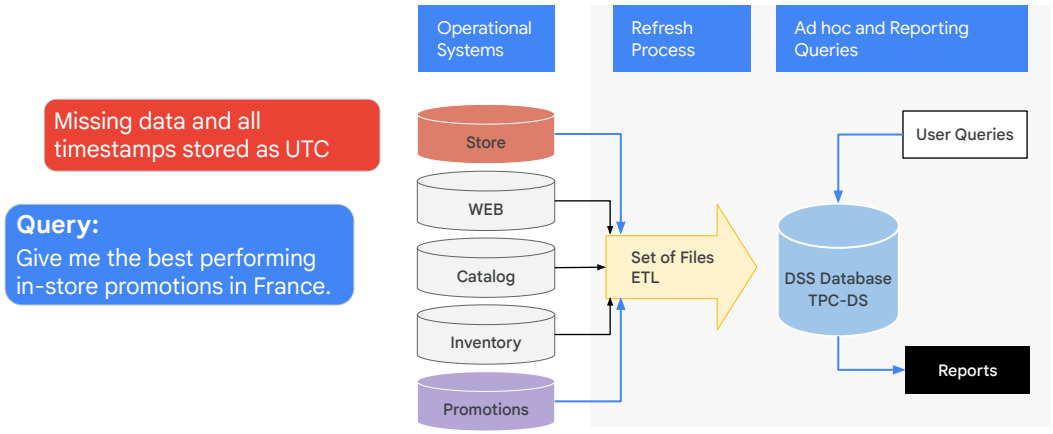
Let's say you are a retailer and you have to consolidate data from multiple source systems. Think about what the use case is.

Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



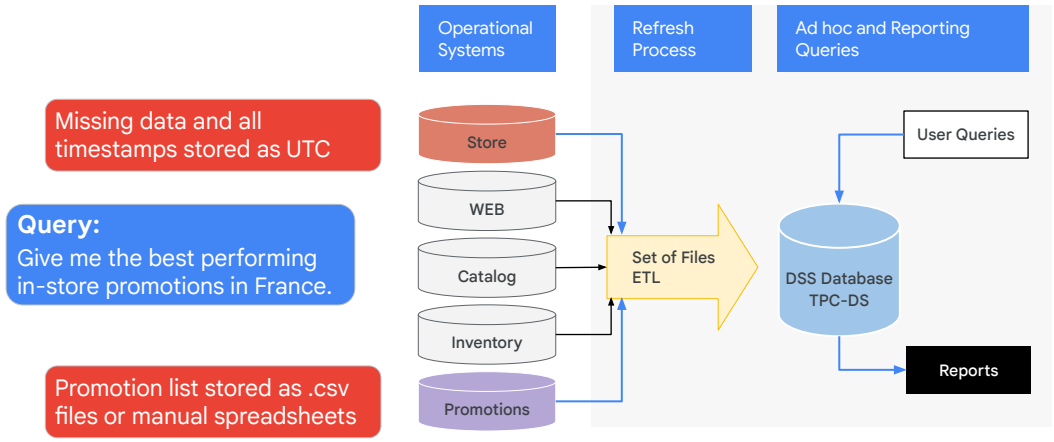
Suppose the use case is to get the best performing in-store promotions in France. You need to get the data from the stores and you have to get the data from the promotions. But perhaps the store data is missing information.

Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



Maybe some of the transactions are in cash, and for those, perhaps there is no information on who the customer is; or some transactions might be spread over multiple receipts, and you might need to combine those transactions because they come from the same customer. Or perhaps the time stamps of the products are stored in local time, whereas you have to spread across the globe, and so before you can do anything, you need to convert everything into UTC.

Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



Similarly, the promotions may not be stored in the transaction database at all. They might be just a text file that somebody loads on their web page, and it has a list of codes that are used by the web application to apply discounts.

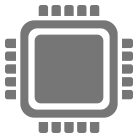
It can be extremely difficult to do a query like finding the best performing in-store promotions because the data has so many problems. Whenever you have data like this, you need to get the raw data and transform it into a form with which you can actually carry out the necessary analysis.

It is obviously best if you can do this sort of cleanup and consolidation just once, and store the resulting data to make further analysis easy. That's the point of a data warehouse.

Challenge: Ensuring you have the compute capacity to meet peak-demand for your team

Access to data

Data accuracy and quality

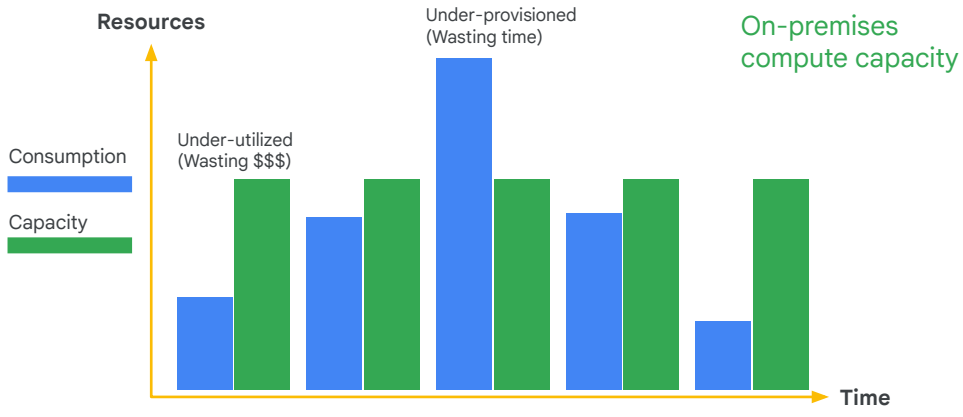


Availability of computational resources

Query performance

If you need to do so much consolidation and cleanup, a common problem that arises is where to carry out this compute. The availability of computation resources can be a challenge.

Challenge: Data Engineers need to manage server and cluster capacity if using on-premise



If you are on an on-premises system, data engineers will need to manage server and cluster capacity and make sure that enough capacity exists to carry out the ETL jobs.

The problem is that the compute needed by these ETL jobs is not constant over time. Very often, it varies week to week, and depending on factors like holidays and promotional sales. This means that when traffic is low, you are wasting money and when traffic is high, your jobs are taking way too long.

Challenge: Queries need to be optimized for performance (caching, parallel execution)

Access to data

Data accuracy and quality

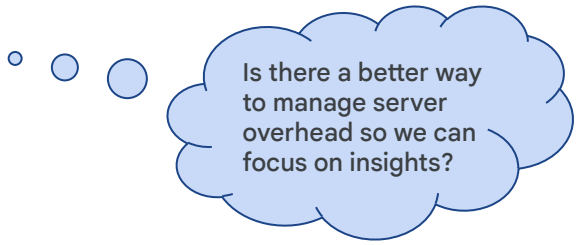
Availability of computational resources

Query performance

Once your data is in your data warehouse, you need to optimize the queries your users are running to make the most efficient use of your compute resources.

Challenge: Managing query performance on-premise comes with added overhead

- Choosing a query engine.
- Continually patching and updating query engine software.
- Managing clusters and when to re-cluster.
- Optimize for concurrent queries and quota / demand between teams.



If you're managing an on-premise data analytics cluster, you will be responsible for choosing a query engine and installing the query engine software and keeping it up to date as well as provisioning any more servers for additional capacity.

Isn't there a better way to manage server overhead so we can focus on insights?

Introduction to BigQuery

There is a much better way to manage server overhead so we can focus on insights. It is to use a serverless data warehouse.

BigQuery is Google Cloud's petabyte-scale serverless data warehouse. You don't have to manage clusters. Just focus on insights.

BigQuery is Google's data warehouse solution

Data warehouse	Data mart	Data lake	Tables and views	Grants
BigQuery replaces a typical data warehouse hardware setup.	BigQuery organizes data tables into units called datasets.	BigQuery defines schemas and issues queries directly on external data sources.	Function the same way as in a traditional data warehouse.	IAM grants permission to perform specific actions.

The BigQuery service replaces the typical hardware setup for a traditional data warehouse. That is, it serves as a collective home for all analytical data in an organization.

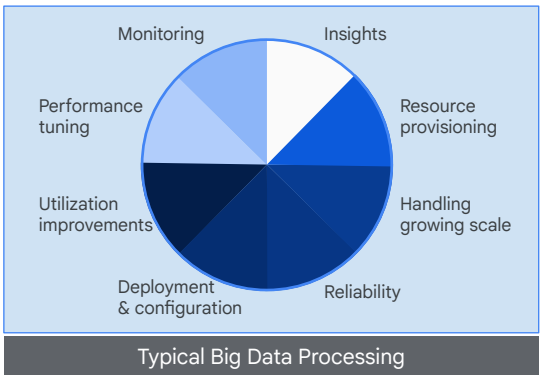
Datasets are collections of tables that can be divided along business lines or a given analytical domain. Each dataset is tied to a Google Cloud project.

A data lake might contain files in Cloud Storage or Google Drive or even transactional data from Cloud Bigtable. BigQuery can define a schema and issue queries directly on external data as federated data sources.

Database tables and views function the same way in BigQuery as they do in a traditional data warehouse, allowing BigQuery to support queries written in a standard SQL dialect which is ANSI: 2011 compliant.

Identity and Access Management is used to grant permission to perform specific actions in BigQuery. This replaces the SQL GRANT and REVOKE statements that are used to manage access permissions in traditional SQL databases.

Cloud allows data engineers to spend less time managing hardware and enabling scale. Let Google do that for you



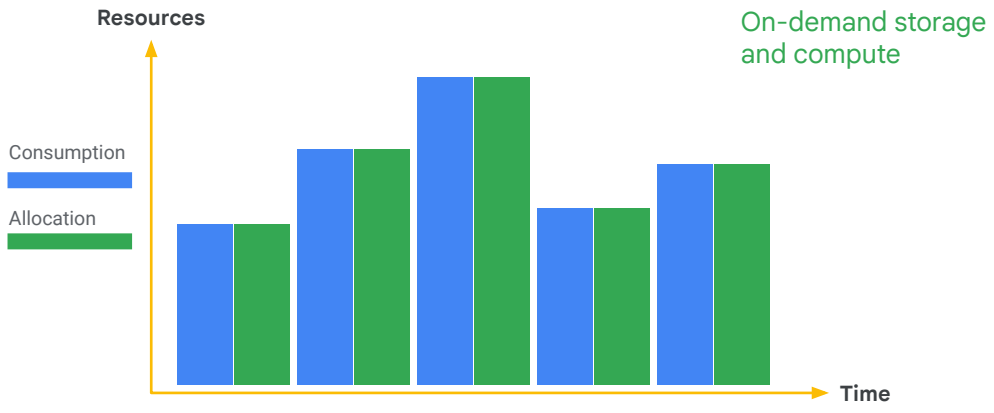
A key consideration behind agility is being able to do more with less, and it's important to make sure that you're not doing things that don't add value.

If you do work that is common across multiple industries, it's probably not something that your business wants to pay for.

The cloud lets you, the data engineer, spend less time managing hardware and more time doing things that are much more customized and specific to the business.

You don't have to be concerned about provisioning and reliability and utilization improvements in performance or tuning on the cloud, so you can spend all your time thinking about how to get better insights from your data.

You don't need to provision resources before using BigQuery



You don't need to provision resources before using BigQuery, unlike many RDBMS systems. BigQuery allocates storage and query resources dynamically based on your usage patterns.

Storage resources are allocated as you consume them and deallocated as you remove data or drop tables.

Query resources are allocated according to query type and complexity. Each query uses some number of slots, which are units of computation that comprise a certain amount of CPU and RAM.

Data Lakes and Data Warehouses

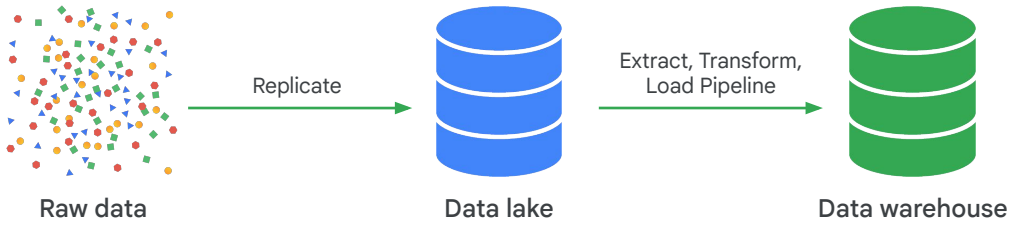
We've defined what a data lake is, and what a data warehouse is. Let's look at these in a bit more detail.

A data engineer gets data into a useable condition

- Get the data to where it can be useful
- **Get the data into a usable condition**
- Add new value to the data
- Manage the data
- Productionize data processes

Recall that we emphasized that the data has to be in a usable condition so that someone can use this data to make decisions. Many times, the raw data is by itself not **very useful.**

A data warehouse stores transformed data in a usable condition for business insights



What are the key considerations when deciding between data warehouse options?

We said that raw data gets replicated and stored in a data lake. In order to make the data usable, you will use Extract-Transform-Load or ETL pipelines to make the data usable, and store this more usable data in a data warehouse.

Let's consider what are the key considerations when deciding between data warehouse options ...

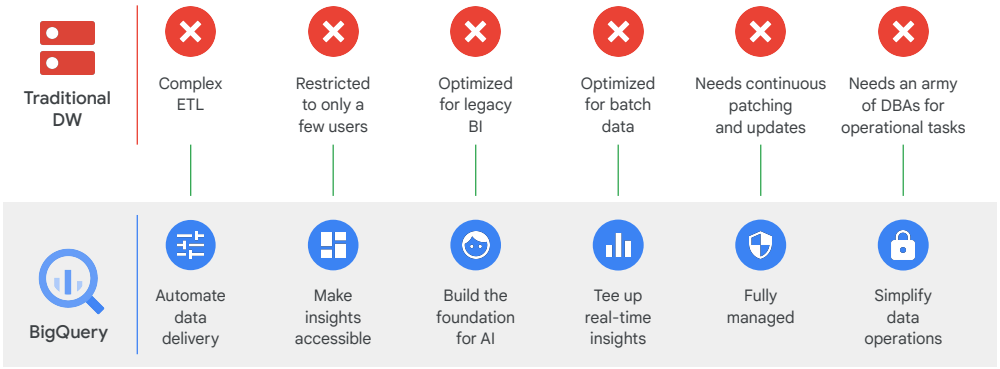
Considerations when choosing a data warehouse

- Can it serve as a sink for both batch and streaming data pipelines?
- Can the data warehouse scale to meet my needs?
- How is the data organized, cataloged, and access controlled?
- Is the warehouse designed for performance?
- What level of maintenance is required by our engineering team?

We need to ask ourselves these questions:

- The data warehouse is going to definitely serve as a sink — you're going to store data in it. But will it be fed by a batch pipeline or by a streaming pipeline? Does the warehouse need to be accurate up-to-the-minute? Or is it enough to load data into it once a day or once a week?
- Will the data warehouse scale to meet my needs? Many cluster-based data warehouses will set per-cluster concurrent query limits. Will those query limits cause a problem? Will the cluster size be large enough to store and traverse your data?
- How is the data organized, cataloged, and access controlled? Will you be able to share access to the data to all your stakeholders? What happens if they want to query the data? Who will pay for the querying?
- Is the warehouse designed for performance? Again, carefully consider concurrent query performance. And whether that performance comes out of the box, or whether you need to go around creating indexes and tuning the data warehouse.
- Finally, what level of maintenance is required by your engineering team?

BigQuery is a modern data warehouse that changes the conventional mode of data warehousing



Traditional data warehouses are hard to manage and operate. They were designed for a batch paradigm of data analytics and for operational reporting needs. The data in the data warehouse was meant to be used by only a few management folks for reporting purposes. BigQuery is a modern data warehouse that changes the conventional mode of data warehousing. Here we can see some of the key comparisons between a traditional data warehouse and BigQuery.

BigQuery provides mechanisms for automated data transfer and powers business applications using technology that teams already know and use, so everyone has access to data insights. You can create read-only shared data sources that both internal and external users can query, and make query results accessible for anyone through user-friendly tools such as Looker, Google Sheets, Tableau, or Google Data Studio.

BigQuery lays the foundation for AI. It's possible to train Tensorflow and Google Cloud Machine Learning models directly with data sets stored in BigQuery, and BigQuery ML can be used to build and train machine learning models with simple SQL. Another extended capability is BigQuery GIS, which allows organizations to analyze geographic data in BigQuery, essential to many critical business decisions that revolve around location data.

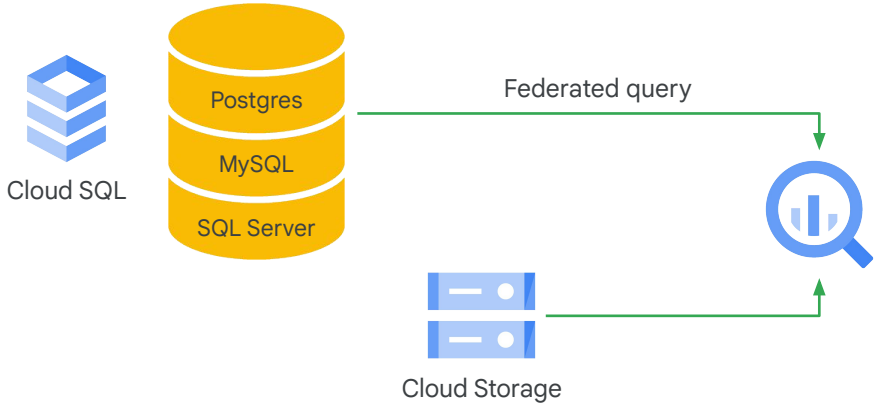
BigQuery also allows organizations to analyze business events real-time, as they unfold, by automatically ingesting data and making it immediately available to query in their data warehouse. This is supported by the ability of BigQuery to ingest up to

100,000 rows of data per second and for petabytes of data to be queried at lightning-fast speeds.

Due to Google's fully managed, serverless infrastructure and globally available network, BigQuery eliminates the work associated with provisioning and maintaining a traditional data warehousing infrastructure.

BigQuery also simplifies data operations through the use of Identity and Access Management to control user access to resources, creating roles and groups and assigning permissions for running jobs and queries in a project, and also providing automatic data backup and replications.

You can simplify Data Warehouse ETL pipelines with external connections to Cloud Storage and Cloud SQL



Even though we talked about getting data into BigQuery by running ETL pipelines, there is another option.

That is to treat BigQuery as a query engine and allow it to query the data in-place. For example, you can use BigQuery to directly query database data in Cloud SQL — that is, managed relational databases like PostgreSQL, MySQL and SQL Server.

You can also use BigQuery to directly query files on Cloud Storage as long as these files are in formats like CSV or Parquet.

The real power comes when you can leave your data in place and still join it against other data in the data warehouse.

Let’s take a look.

Transactional Databases Versus Data Warehouses

Data engineers may be responsible for both the back end transactional database systems that support your company's applications AND the data warehouses that support your analytic workloads. In this lesson, you'll explore the differences between databases and data warehouses and the Google Cloud solutions for each workload.

Cloud SQL is fully managed SQL Server, Postgres, or MySQL for your Relational Database (transactional RDBMS)

Why not simply use Cloud SQL for reporting workflows?

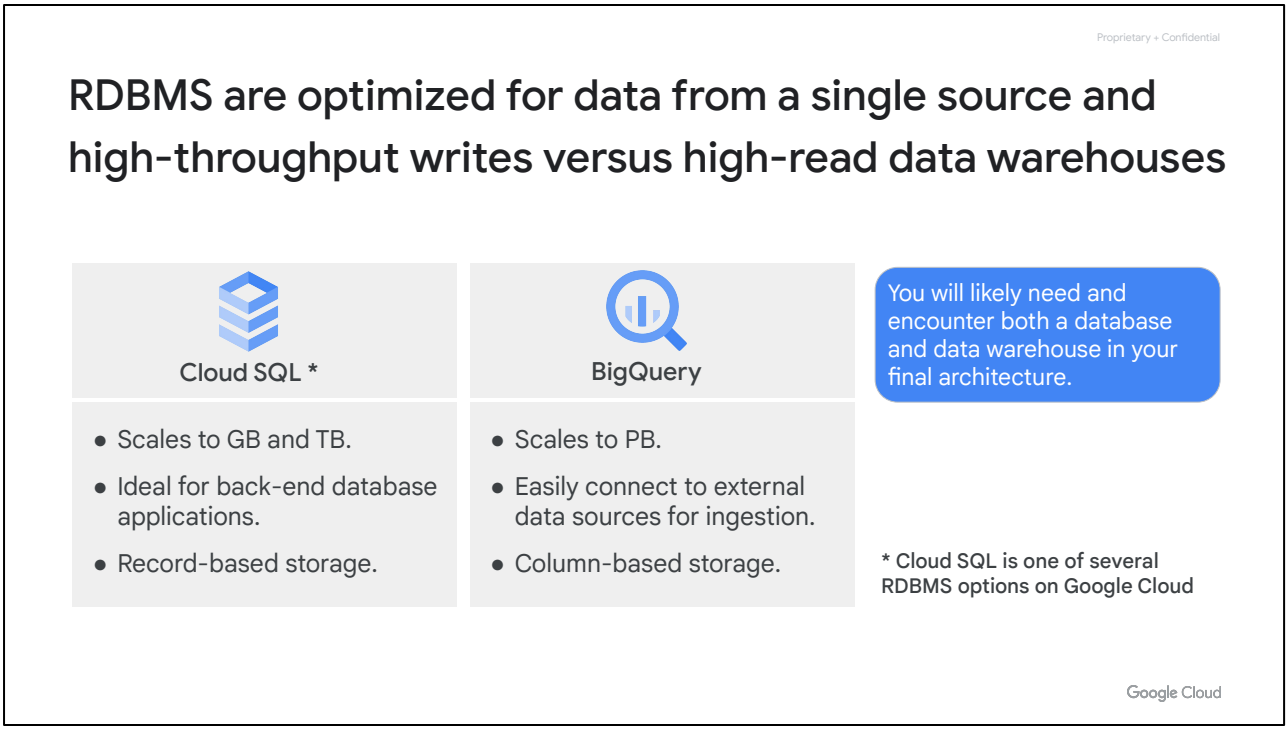
- Automatic encryption
- 30 TB storage capacity
- 60,000 IOPS (read/write per second)
- Auto-scale and auto backup

If you have SQL Server, MySQL, or PostgreSQL as your relational database, you can migrate it to Cloud SQL which is Google Cloud's fully managed relational database solution.

Cloud SQL delivers high performance and scalability with up to 30 TB of storage capacity, 60,000 IOPS, and 416 GB of RAM per instance. You can take advantage of storage auto-scale to handle growing database needs with zero downtime.

One question you might get asked is: "Why not use simply use Cloud SQL for reporting workflows? You can run SQL directly on the database right?"

https://cloud.google.com/products/databases/



This is a great question and will be answered in greater detail in the **Building a Data Warehouse** module. Google Cloud has several options for RDBMS's including Spanner and Cloud SQL.

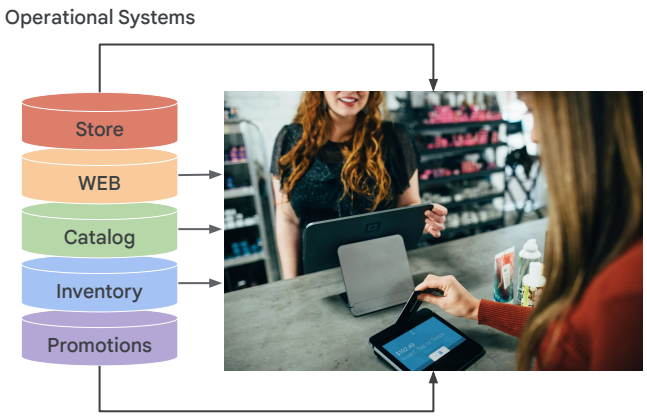
When considering Cloud SQL, be aware that Cloud SQL is optimized to be a database for transactions (or writes) and BigQuery is a data warehouse optimized for reporting workloads (mostly reads).

The fundamental architecture of these data storage options is quite different. Cloud SQL databases are RECORD-based storage -- meaning the entire record must be opened on disk, even if you just selected a single column in your query.

BigQuery is COLUMN-based storage, which as you might guess, allows for really wide reporting schemas since you can simply read individual columns out from disk.

Relational database management systems (RDBMS) are critical for managing new transactions

RDBMS are optimized for high throughput **WRITES** to **RECORDS**.



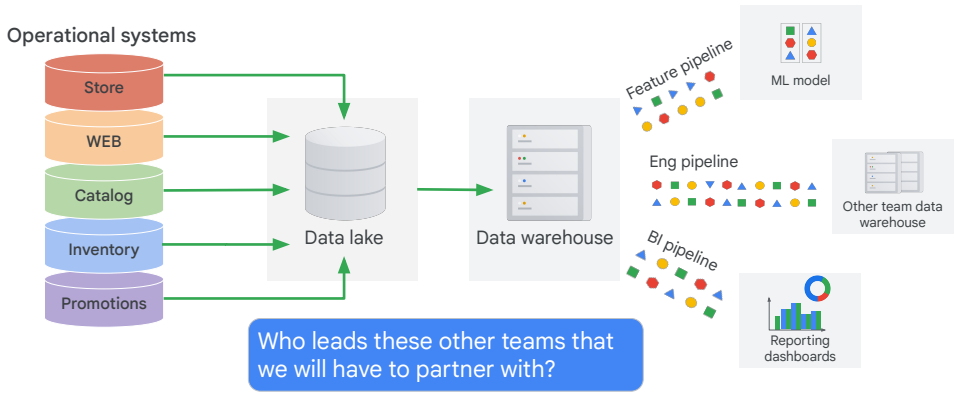
This isn't to say RDBMS' aren't as performant as Data Warehouses -- they serve two different purposes. RDBMS helps your business manage new transactions.

Take this point-of-sale terminal at a storefront. Each order and product is likely written out as new records in a relational database somewhere. This database may store all of the orders received from their website, all of the products listed in the catalog, or the number of items in their inventory.

This is so that when an existing order is changed, it can be quickly updated in the database. Transactional systems allow for a single row in a database table to be modified in a consistent way. They also are built on certain relational database principles, like referential integrity, to guard against cases like a customer ordering a product that doesn't exist in the product table.

So where does all this raw data end up in our data lake and data warehouse discussion? What's the complete picture?

The complete picture: Source data comes into the data lake, is processed into the data warehouse and made available for insights



Here it is. Our operational systems, like our relational databases that store online orders, inventory, and promotions, are our raw data sources on the left. Note that this isn't exhaustive -- you could have other source systems that are manual like CSV files or spreadsheets too.

These upstream data sources get gathered together in a single consolidated location in our **data lake** which is designed for durability and high availability.

Once in the data lake, the data often needs to be processed via transformations that then output the data into our **data warehouse** where it is ready for use by downstream teams.

Here are three quick examples of other teams that often build pipelines on our data warehouse:

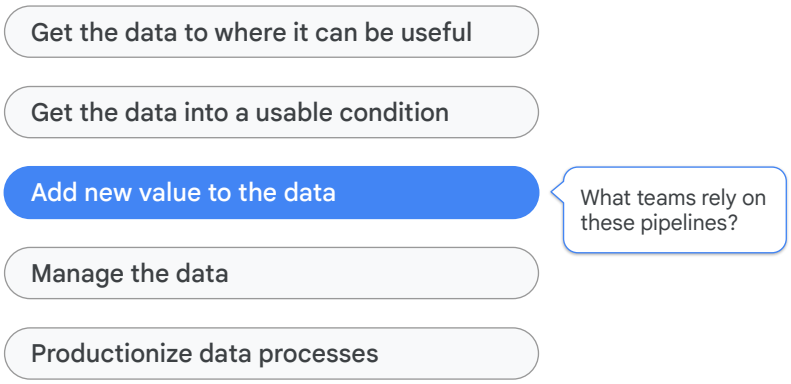
- An **ML team** may build a pipeline to get features for their models, …
- An **engineering team** may be using our data as part of their data warehouse, …
- And a **BI team** may want to build dashboards using some of our data.

So who works on these teams and how do they partner with our data engineering team?

Partner Effectively with Other Data Teams

Since a data warehouse also serves other teams, it is crucial to learn how to partner effectively with them.

A data engineer builds data pipelines to enable data-driven decisions



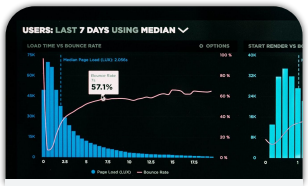
Remember that once you've got data where it can be useful and it's in a usable condition we need to add new value to the data through analytics and machine learning. What teams might rely on our data?

Many teams rely on partnerships with data engineering to get value out of their data

How might each of these teams rely on data engineering?

ML Engineer

Data Analyst



Data Engineer

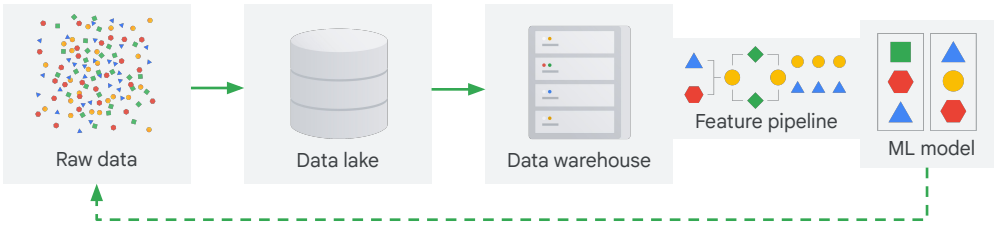
There are many data teams that rely on your data warehouse and partnerships with data engineering to build and maintain new data pipelines.

The three most common clients are:

- The machine learning engineer
- The data or BI analyst, and
- Other data engineers

Let's examine how each of these roles interacts with your new data warehouse and how data engineers can best partner with them.

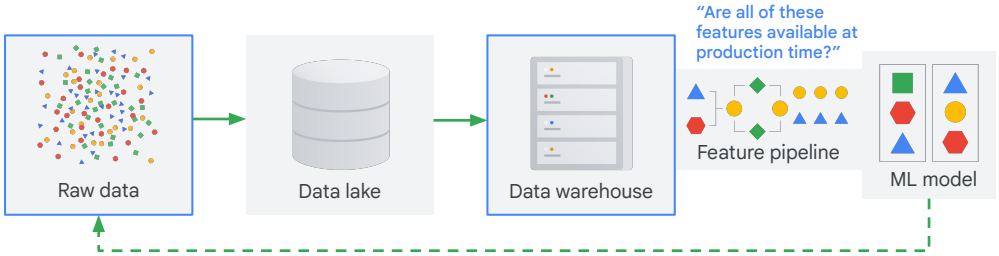
Machine learning teams need data engineers to help them capture new features in a stable pipeline



As you'll see in our course on machine learning, an ML team's models rely on having lots of high quality input data to create, train, test, evaluate, and serve their models. They will often partner with data engineering teams to build pipelines and datasets for use in their models.

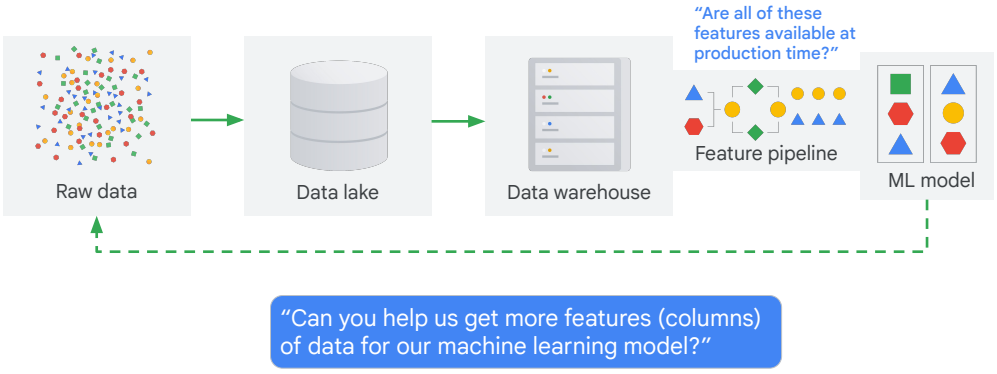
Two common questions you may get asked are …

Machine learning teams need data engineers to help them capture new features in a stable pipeline



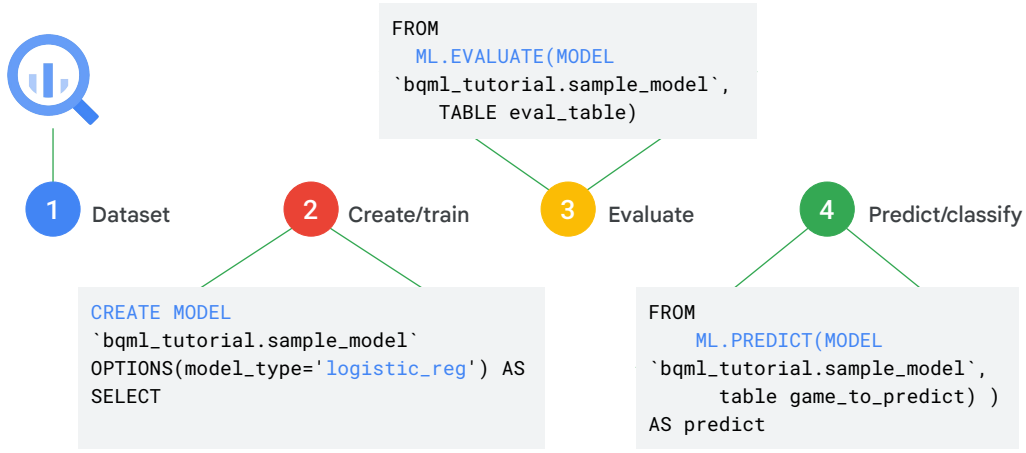
"How long does it take for a transaction to make it from raw data all the way into the data warehouse"? They're asking this because any data that they train their models on must also be available at prediction-time as well. If there is a long delay in collecting and aggregating the raw data it will impact the ML team's ability to create useful models.

Machine learning teams need data engineers to help them capture new features in a stable pipeline



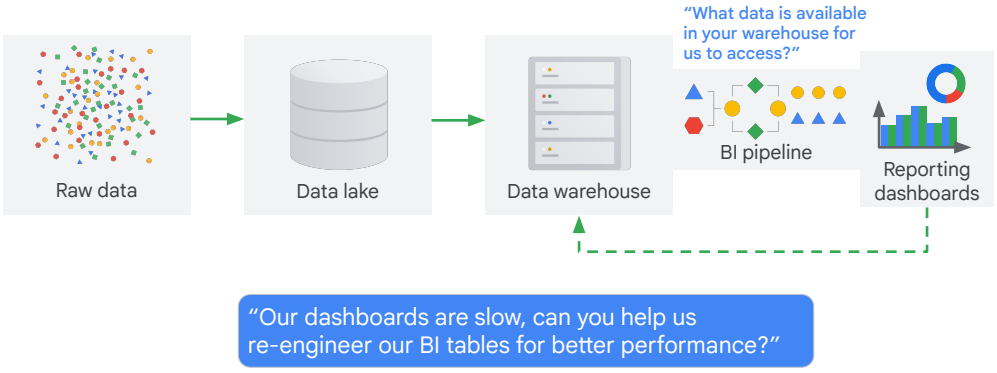
A second question that you will definitely get asked is how difficult it would be to add more columns or rows of data into certain datasets. Again, the ML team relies on teasing out relationships between the columns of data and having a rich history to train models on. You will earn the trust of your partner ML teams by making your datasets easily discoverable, documented, and available to ML teams to experiment on quickly.

Add value: Machine learning directly in BigQuery



A unique feature of BigQuery is that you can create high-performing machine learning models directly in BigQuery using just SQL by using Bigquery ML. Here is the actual model code to CREATE a model, EVALUATE it, and then MAKE predictions. You'll see this again in our lectures on machine learning later on.

Data analysis and business intelligence teams rely on data engineering to showcase the latest insights

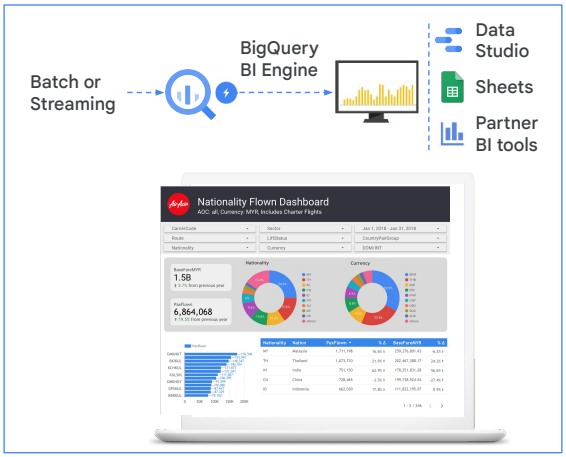


Other critical stakeholders are your business intelligence and data analyst teams that rely on good clean data to query for insights and build dashboards.

These teams need datasets that have clearly defined schema definitions, the ability to quickly preview rows, and the performance to scale to many concurrent dashboard users.

Add value: BI Engine for dashboard performance

- No need to manage OLAP cubes or separate BI servers for dashboard performance.
- Natively integrates with BigQuery streaming for real-time data refresh.
- Column oriented in-memory BI execution engine.

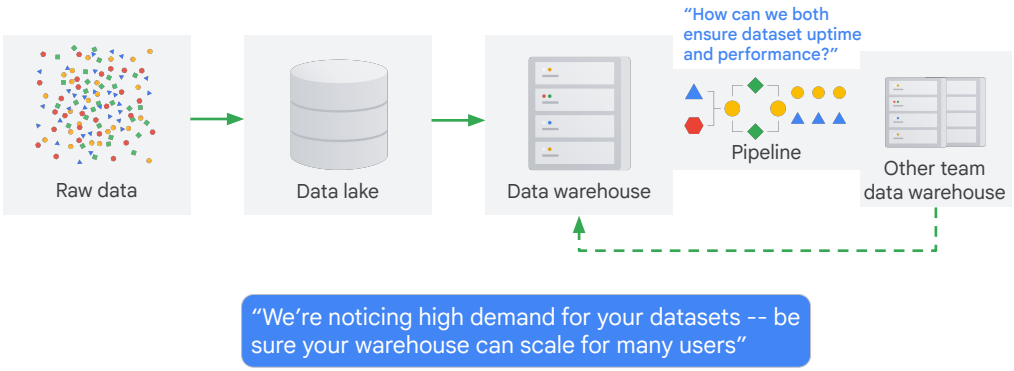


One of the Google Cloud products that helps manage the performance of dashboards is BigQuery BI Engine. BI Engine is a fast, in-memory analysis service that is built directly into BigQuery and available to speed up your business intelligence applications.

Historically, BI teams would have to build, manage, and optimize their own BI servers and OLAP cubes to support reporting applications. Now, with BI Engine, you can get sub-second query response time on your BigQuery datasets without having to create your own cubes. BI Engine is built on top of the same BigQuery storage and compute architecture and servers as a fast in-memory intelligent caching service that maintains state.

https://www.youtube.com/watch?v=TqIrIcmqPgo

Other data engineering teams may rely on your pipelines being timely and error free

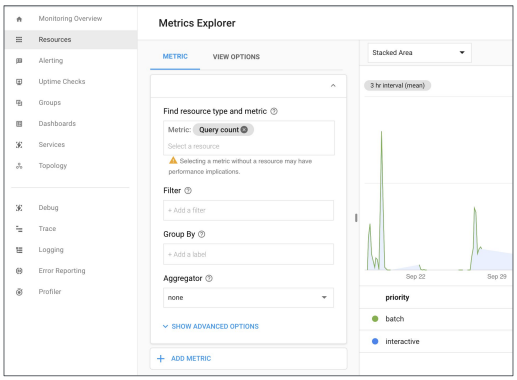


One last group of stakeholders are other data engineers that rely on the uptime and performance of your data warehouse and pipelines for their downstream data lakes and data warehouses.

They will often ask "How can you ensure that the data pipeline we depend on will always be available when we need it?"

Or, "We are noticing high demand for certain really popular datasets. How can you monitor and scale the health of your entire data ecosystem?"

Add value: Cloud Monitoring for performance



- View in-flight and completed queries.
- Create alerts and send notifications.
- Track spending on BigQuery resources.
- Use Cloud Audit Logs to view actual job information (who executed, what query was ran).

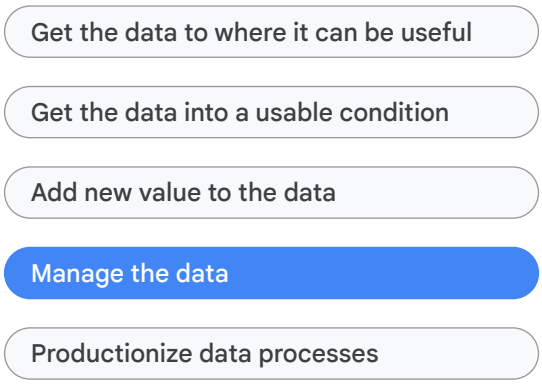
- One popular way is to use the built-in Cloud Monitoring of all resources on Google Cloud.
- Since Google Cloud Storage and BigQuery are resources, you can set up alerts and notifications for metrics like “Query Count” or “Bytes of Data Processed” so you can better track usage and performance.
- Another two reasons why Cloud Monitoring is used is for tracking spending of all the different resources used and what the billing trends are for your team or organization.
- And lastly, you can use the Cloud Audit Logs to view actual query job information to see granular level details about which queries were executed and by whom. This is useful if you have sensitive datasets that you need to monitor closely. A topic we will discuss more next.

https://cloud.google.com/bigquery/docs/monitoring

Manage Data Access and Governance

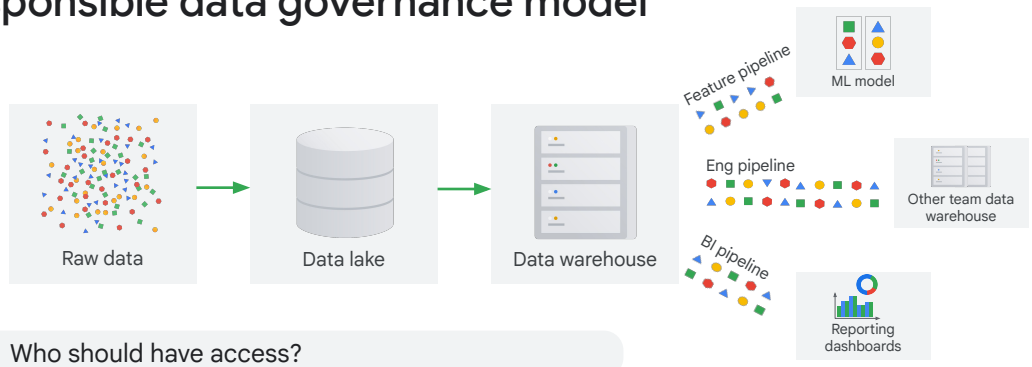
As part of being an effective partner, your engineering team will be asked to set up data access policies and overall governance of how data is to be used and NOT used by your users.

A data engineer manages data access and governance



This is what we mean when we say a data engineer must manage the data. This includes critical topics, such as privacy and security. What are some key considerations when managing certain datasets?

Data engineering must set and communicate a responsible data governance model

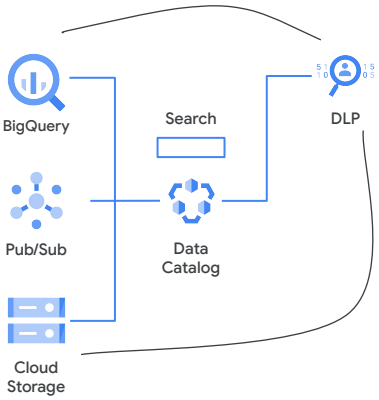


- Who should have access?
- How is PII handled?
- How can we educate end-users on our data catalog?

Clearly communicating a data governance model for:

- Who should and should not have access?
- How is Personally Identifiable Information (like phone numbers or email addresses) handled?
- And even more basic tasks like how will our end users discover the different datasets we have for analysis?

Cloud Data Catalog is a managed data discovery + Data Loss Prevention API for guarding PII



Data Catalog

Simplify data discovery at any scale:
Fully managed metadata management service with no infrastructure to set up or manage.

Unified view of all datasets:
Central and secure data catalog across Google Cloud with metadata capture and tagging.

Data governance foundation:
Security compliance with access level controls along with Cloud Data Loss Prevention integration for handling sensitive data.

One solution for data governance is the Cloud Data Catalog and the Data Loss Prevention API.

- The Data Catalog makes all the metadata about your datasets available to search for your users. You group datasets together with tags, flag certain columns as sensitive, etc.
- Why is this useful? If you have many different datasets with many different tables -- which different users have different access levels to -- the Data Catalog provides a single unified user experience for discovering those datasets quickly. No more hunting for specific table names in SQL first.
- Often used in conjunction with Data Catalog is the Cloud Data Loss Prevention API, or DLP API, which helps you better understand and manage sensitive data. It provides fast, scalable classification and redaction for sensitive data elements like credit card numbers, names, social security numbers, US and selected international identifier numbers, phone numbers, and Google Cloud credentials.

Build Production-ready Pipelines

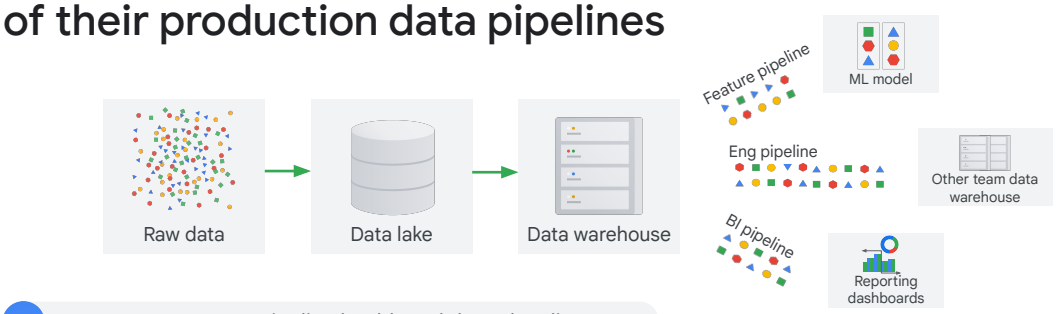
Once your data lakes and data warehouses are set up and your governance policy is in place, it's time to productionalize the whole operation and automate and monitor as much of it as we can.

A data engineer builds production data pipelines to enable data-driven decisions

- Get the data to where it can be useful
- Get the data into a usable condition
- Add new value to the data
- Manage the data
- **Productionize data processes**

That's what we mean when we say productionalize the data process. It has to be an end-to-end and scalable data processing system.

Data engineering owns the health and future of their production data pipelines



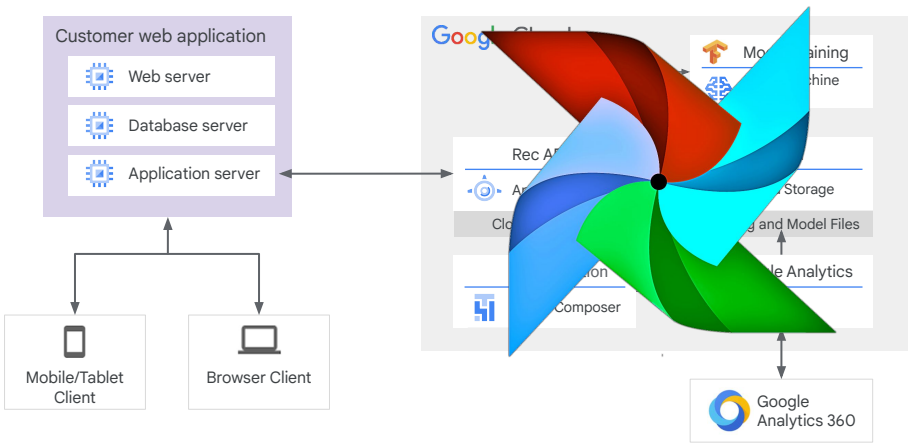
- How can we ensure pipeline health and data cleanliness?
- How do we productionalize these pipelines to minimize maintenance and maximize uptime?
- How do we respond and adapt to changing schemas and business needs?
- Are we using the latest data engineering tools and best practices?

Your data engineering team is responsible for the health of the plumbing -- that is the pipelines -- and ensuring that the data is available and up-to-date for analytic and ML workloads.

Common questions that you should ask at this phase are:

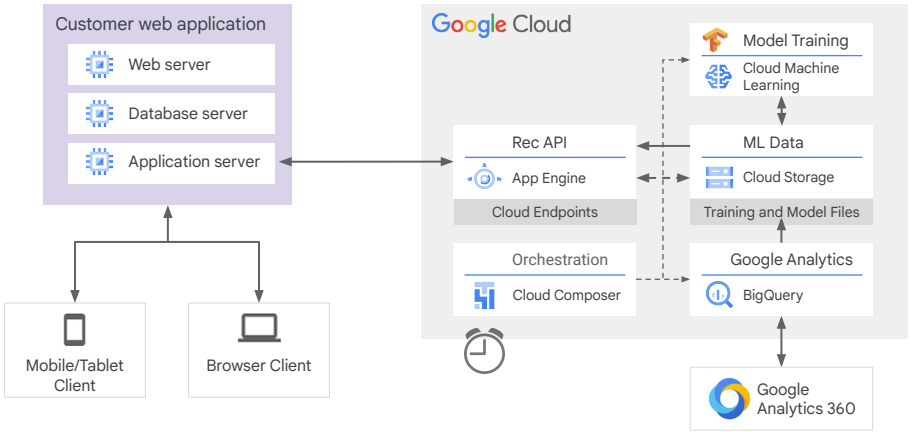
- How can we ensure pipeline health and data cleanliness?
- How do we productionalize these pipelines to minimize maintenance and maximize uptime?
- How do we respond and adapt to changing schemas and business needs?
- And are we using the latest data engineering tools and best practices?

Cloud Composer (managed Apache Airflow) is used to orchestrate production workflows



One common workflow orchestration tool used by enterprises is Apache Airflow.

Cloud Composer (managed Apache Airflow) is used to orchestrate production workflows



Google Cloud has a fully-managed version of Airflow called Cloud Composer.

Cloud Composer helps your data engineering team orchestrate all the pieces to the data engineering puzzle that we have discussed so far (and even more that you haven't come across yet!).

For example, when a new CSV file gets dropped into Cloud Storage, you can automatically have that trigger an event that kicks off a data processing workflow and puts that data directly into your data warehouse.

The power of this tool comes from the fact that Google Cloud big data products and services have API endpoints that you can call.

A Cloud Composer job can then run every night or every hour and kick-off your entire pipeline from raw data, to the data lake, and into the data warehouse for you.

We'll discuss workflow orchestration in greater detail in later modules and you'll do a lab on Cloud Composer as well.

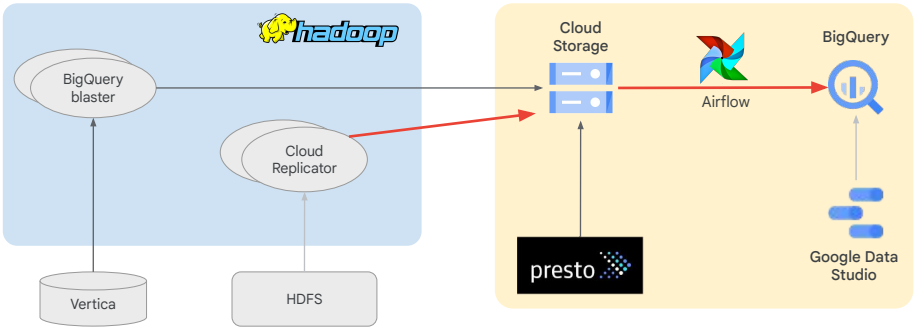
Google Cloud Customer Case Study

We have looked at a lot of different aspects of what a data engineer has to do. Let's look at a case study of how a Google Cloud customer solves a specific business problem. That will help tie all these different aspects together.

Twitter democratized data analysis using BigQuery

"We believe that users with a wide range of technical skills should be able to discover data and have access to SQL-based analysis and visualization tools that perform well"

-- Twitter



Twitter has large amounts of data, and they also have high-powered sales teams and marketing teams, which for a long time did not have access to the data and couldn't use that data for carrying out the analysis that they wanted to be able to do.

Much of the data was stored on Hadoop clusters that were completely over-taxed. So Twitter replicated some of that data from HDFS on to Cloud Storage, loaded it into BigQuery, and provided BigQuery to the rest of the organization. These were some of the most frequently requested data sets within Twitter, and they discovered that, with ready access to the data, many people who were not data analysts are now analyzing data and making better decisions as a result.

For more information, a link to the blog post is available in the PDF version of this content under Course Resources.

https://blog.twitter.com/engineering/en_us/topics/infrastructure/2019/democratizing-data-analysis-with-google-bigquery.html

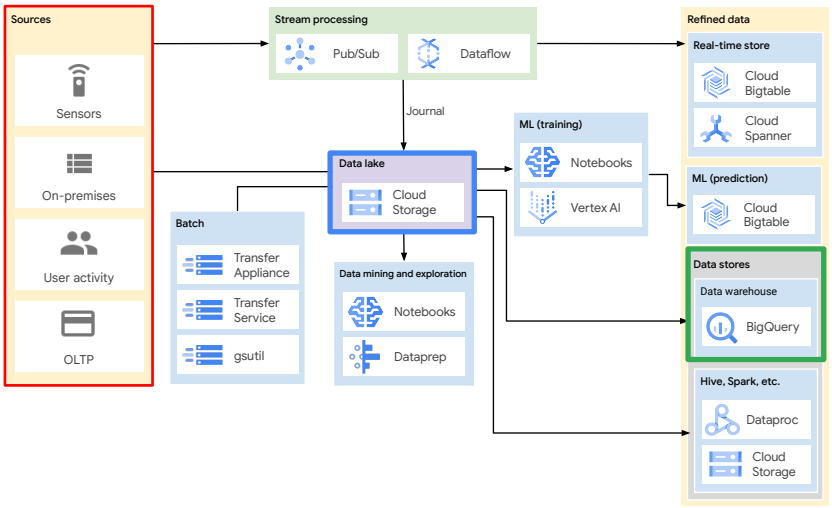
Recap

- Data sources
- Data lakes
- Data warehouses
- Google Cloud solutions for Data Engineering

Let’s summarize the major topics we covered so far in this introduction.

Concept review

Data sources feed into a Data Lake and are processed into your Data Warehouse for analysis.



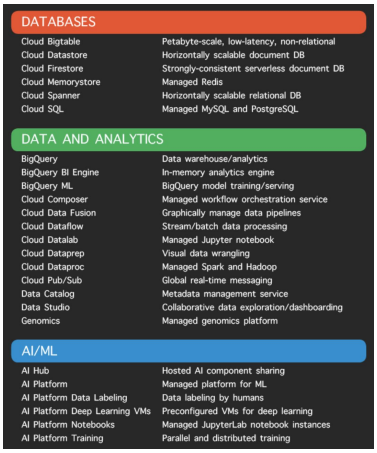
- Recall that your data sources are your upstream systems like RDBMS' and other raw data that comes from your business in different formats.
- Data lakes -- your consolidated location for raw data that is durable and highly available. In this example our data lake is Cloud Storage.
- And data warehouses which are the end result of preprocessing the raw data in your data lake and getting it ready for analytic and ML workloads.

You'll notice a lot of other Google Cloud product icons here like batch and streaming data into your lake and running ML on your data. We'll cover those topics in detail later in this course.

Here's a useful guide for "Google Cloud products in 4 words or less"

https://github.com/gregsramblings/google-cloud-4-words

Updated continually By Greg Wilson - Google DevRel



DATABASES	
Cloud Bigtable	Petabyte-scale, low-latency, non-relational
Cloud Datastore	Horizontally scalable document DB
Cloud Firestore	Strongly-consistent serverless document DB
Cloud Memorystore	Managed Redis
Cloud Spanner	Horizontally scalable relational DB
Cloud SQL	Managed MySQL and PostgreSQL

DATA AND ANALYTICS	
BigQuery	Data warehouse/analytics
BigQuery BI Engine	In-memory analytics engine
BigQuery ML	BigQuery model training/serving
Cloud Composer	Managed workflow orchestration service
Cloud Data Fusion	Graphically manage data pipelines
Cloud Dataflow	Stream/batch data processing
Cloud Datalab	Managed Jupyter notebook
Cloud Dataprep	Visual data wrangling
Cloud Dataproc	Managed Spark and Hadoop
Cloud Pub/Sub	Global real-time messaging
Data Catalog	Metadata management service
Data Studio	Collaborative data exploration/dashboarding
Genomics	Managed genomics platform

AI/ML	
AI Hub	Hosted AI component sharing
AI Platform	Managed platform for ML
AI Platform Data Labeling	Data labeling by humans
AI Platform Deep Learning VMs	Preconfigured VMs for deep learning
AI Platform Notebooks	Managed JupyterLab notebook instances
AI Platform Training	Parallel and distributed training

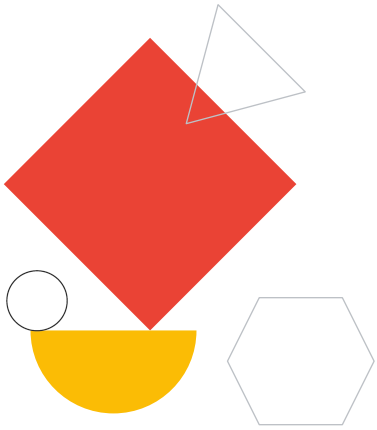
Google Cloud

A useful cheatsheet to bookmark as a reference is the "Google Cloud Products in 4 words or less" which is actively maintained on GitHub by our Google Developer Relations team. It's also a great way to stay on top of new products and services that come out just by following the GitHub commits!

https://github.com/gregsramblings/google-cloud-4-words

Lab Intro

Using BigQuery to do Analysis



Now it's time to practice analyzing data with BigQuery in your lab. In this lab, you will execute interactive queries in the BigQuery console, and then combine and run analytics on multiple datasets.