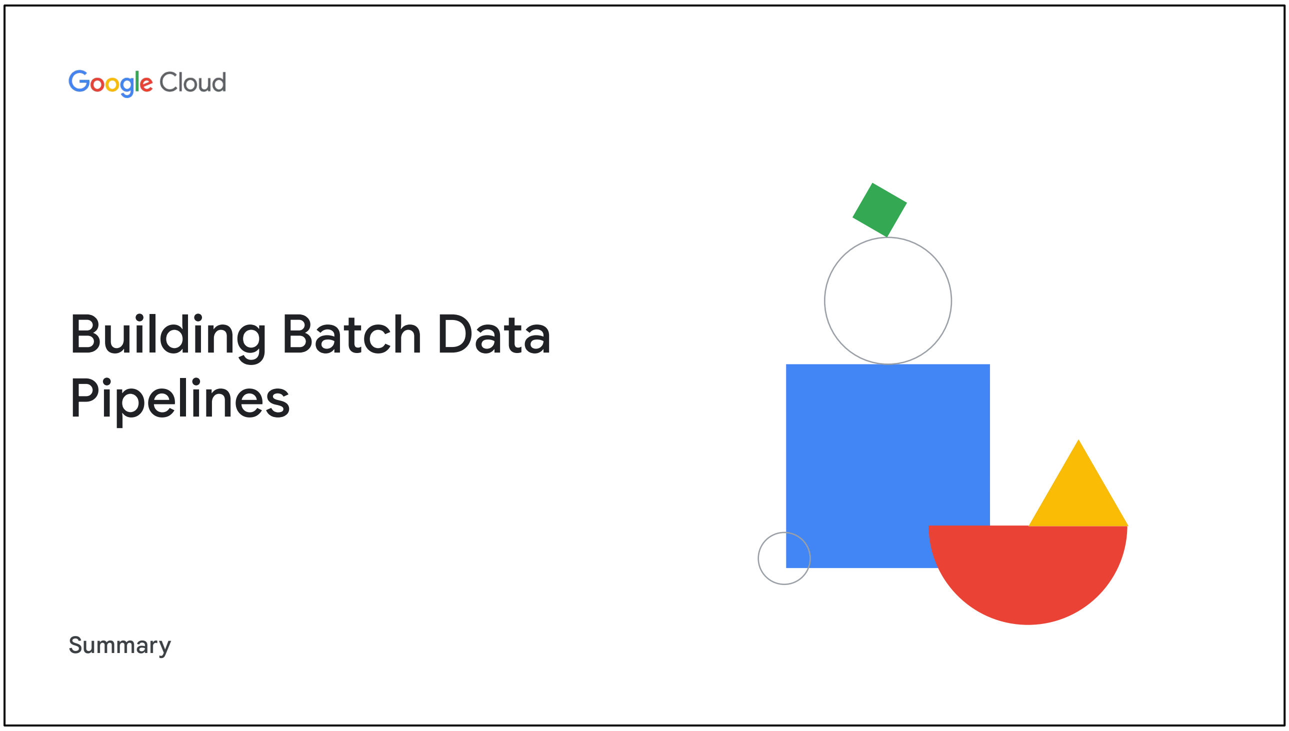
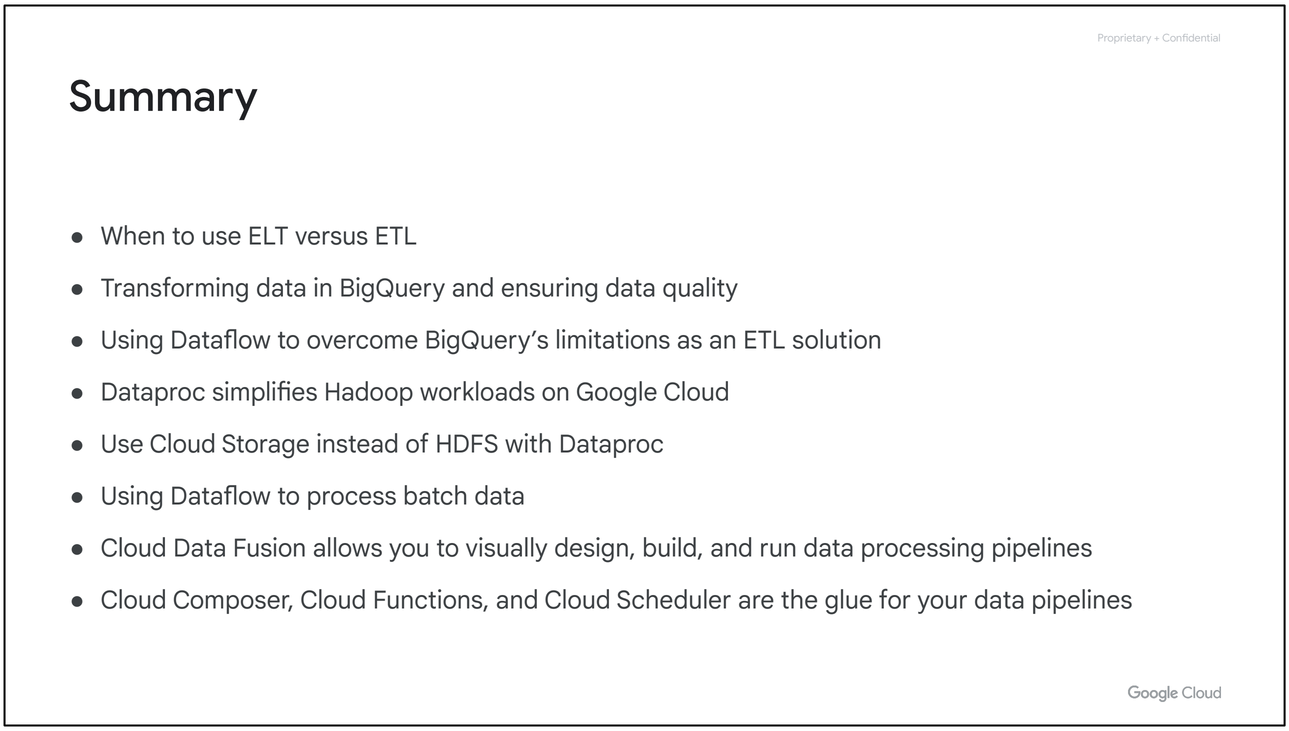


Google Cloud

Building Batch Data Pipelines

Summary

Let’s review some keys concepts we covered on building batch data pipelines.



Summary

- When to use ELT versus ETL
- Transforming data in BigQuery and ensuring data quality
- Using Dataflow to overcome BigQuery's limitations as an ETL solution
- Dataproc simplifies Hadoop workloads on Google Cloud
- Use Cloud Storage instead of HDFS with Dataproc
- Using Dataflow to process batch data
- Cloud Data Fusion allows you to visually design, build, and run data processing pipelines
- Cloud Composer, Cloud Functions, and Cloud Scheduler are the glue for your data pipelines

We covered the different methods of loading data into your data lakes and warehouses.

Remember ELT versus ETL and when to use each. ELT of extract, load, transform is a common pattern for when the transformations you want on the dataset are minor and then can be handled after you load the data. An example is loading the data into BigQuery first, and then doing SQL on the raw data and storing the transformed version as a new table. On the other hand, ETL or transforming before loading, is common when you have data that needs more complex transformations, or the sheer volume of data makes it better to do those transformations before loading. That's where you would want to use a batch pipeline like Dataflow.

Next, we discussed how to transform data in BigQuery using SQL, and you saw the common operations you can perform to ensure your structured data is ready for insights.

Lastly, you practiced building batch pipelines in the serverless way with Dataflow.

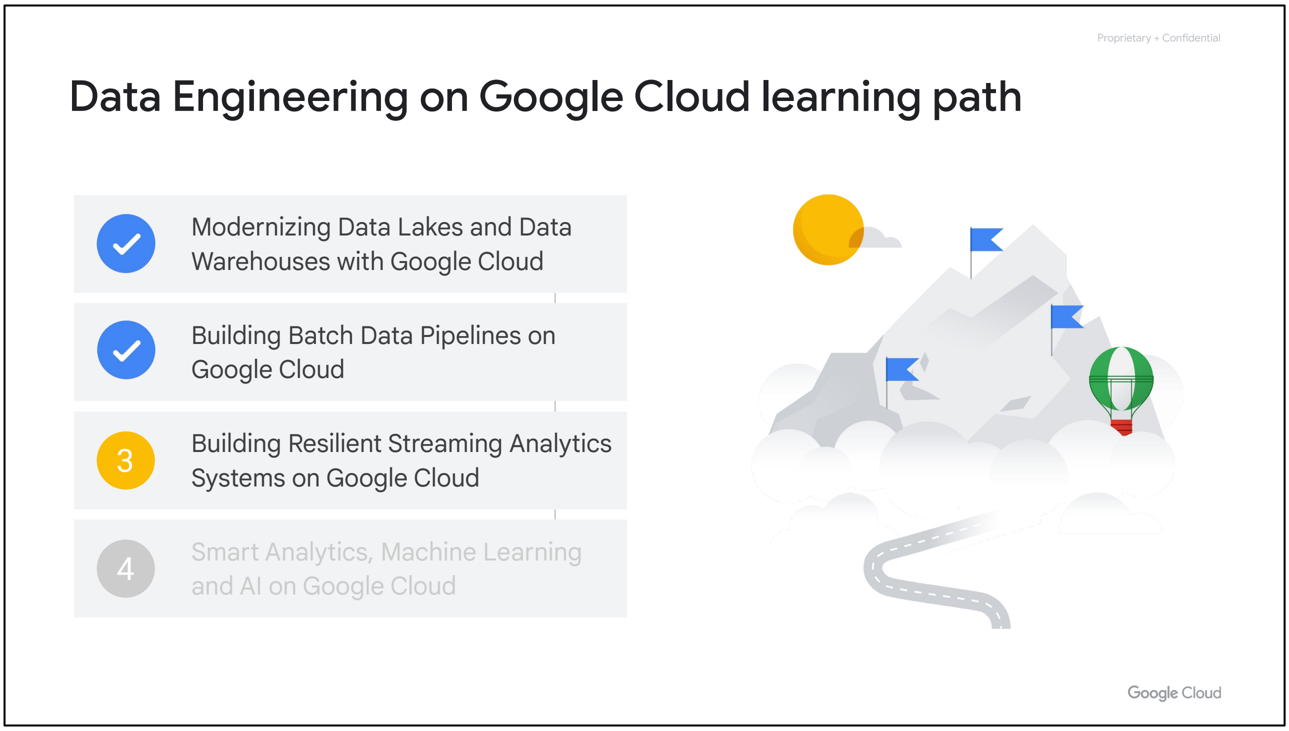
In our Hadoop module, we discussed Dataproc in detail. You learned that you can lift and shift your existing Hadoop workloads to the cloud with no code changes and they will just work.

However, once in the Cloud there were additional optimizations you could make, like using Cloud Storage for cluster storage, instead of HDFS for greater efficiency and cost savings.

We then used Dataflow to build batch data pipelines using Dataflow Templates and by writing them ourselves. Recall that the fundamental unit of logical data in the pipeline is the PCollection, which stands for a parallel collection. Dataflow will automatically split up your dataset into many pieces and farm out the processing across as many worker VMs as it needs to complete the job. Dataflow is a serverless application, which means you will have some control over the maximum number of workers, the actual processing work, and auto-scaling of workers up and down as the demand requires.

In our module on managing data pipelines with Google Cloud, we discussed two new tools: Cloud Data Fusion and Cloud Composer. Recall that Cloud Data Fusion allows data analysts and ETL developers to wrangle data and build pipelines in a visual way. The technology then builds and executes the pipelines on a runner. With Cloud Data Fusion, you also get access to the lineage of each data field, which is the series of any transformation logic that happened before and after the field reaches your dataset.

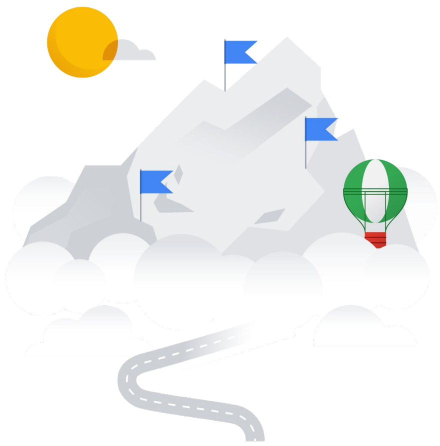
The second tool we covered is Cloud Composer as a workflow orchestrator. Cloud Composer is managed Apache Airflow and allows you to, at a high level, command Google Cloud services in a DAG to perform complex operations. These DAGs can be user scheduled or event-driven with Cloud Functions. Recall the example where when a new CSV was uploaded to our Cloud Storage bucket, a Cloud Function was triggered to start a Dataflow pipeline for processing, and then sync the data into BigQuery.



Proprietary + Confidential

Data Engineering on Google Cloud learning path

- ✓ Modernizing Data Lakes and Data Warehouses with Google Cloud
- ✓ Building Batch Data Pipelines on Google Cloud
- 3 Building Resilient Streaming Analytics Systems on Google Cloud
- 4 Smart Analytics, Machine Learning and AI on Google Cloud



Google Cloud

We've completed **Building Batch Data Pipelines on Google Cloud**.

Building Resilient Streaming Analytics Systems on Google Cloud is the third part of the Data Engineering on Google Cloud course, which we will cover on day 3.