A customer had this interesting business requirement...

The overall business requirement was to migrate to Cloud, an on-premise reporting solution aimed to produce daily reports to regulators.

The on-premises solution was coded using a 'SQL-like language' and it was run on a Hadoop cluster using Spark/MapReduce (leveraging proprietary third-party software).

The client wanted to optimize the target cloud solution to:

- Minimize changes to their processes and codebase.
- Leverage PaaS and native cloud solution (i.e., avoid third-party software).
- Significantly improve performance (from hours to minutes).

We mapped that to technical requirements like this…

Business Requirements

- Minimum changes to their processes and codebase
- Leverage PaaS and native cloud solution (i.e., avoid third-party software)
- Significantly improve performance (from hours to minutes)

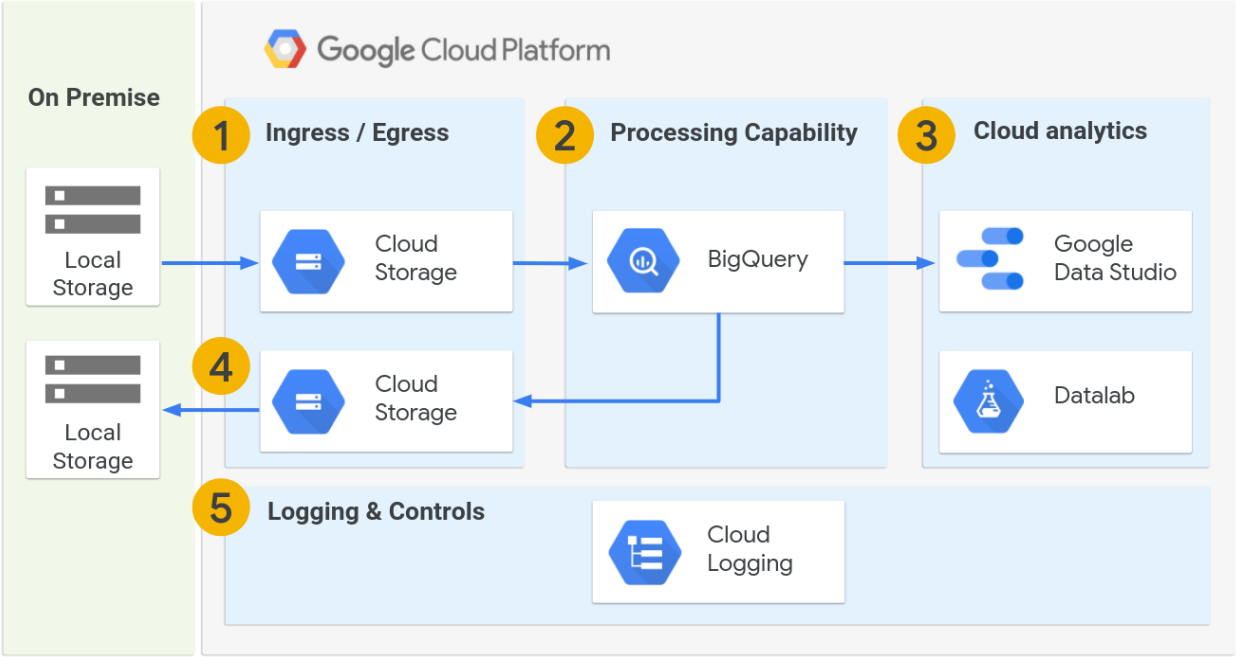
Technical Requirements

- Programmatically convert 'SQL-like' into ANSI SQL
- No changes to source/target input structures
- Automated 'black box' regression testing
- Minimize number of systems/interfaces, aim for full execution in BigQuery (i.e., remove the need for a Hadoop cluster)
- Analyze and (manually) optimize the SQL code in BigQuery for performance

The financial reporting applications run either daily or monthly. So we knew that all we needed to do was set up a pipeline to handle their requirements and then run the pipeline as required by their applications.

And this is how we implemented that technical requirement.

- Source data is ingested in Cloud Storage (no changes to sourcing)
- Output reports are generated entirely in BigQuery
- Cloud Analytics allow users to review the reports
- Data is egressed on-premises (no changes to target structures → no changes downstream)
- Logs and controls out of the box in Cloud Logging



We ended up with a fairly simple solution. The most notable part is what is not in the picture -- and that is Hadoop. Recall that the customer's application that was performing the processing was on Hadoop and some MapReduce. We were able to port that data out of the Hadoop cluster to Cloud Storage. Once it was in Cloud Storage, we found that the processing that was being done was simple enough that we were able to implement it in the processing front-end of BigQuery. This created the Liquidity report they wanted. And we were able to use this for analytics in Google Data Studio and for some analytics processing in Datalab.

The final reports are pushed first into Cloud Storage and then back into the on premise storage. This was important because it meant no changes in their business process were needed.