Architecting database migration and replication using Striim

This document describes using Striim (https://striim.com) to architect database migration and replication. This document focuses on architectural concepts that relate to database migration and continuous replication. It is intended for database architects and database engineers as well as data owners who plan to migrate or replicate data sources to Google Cloud.

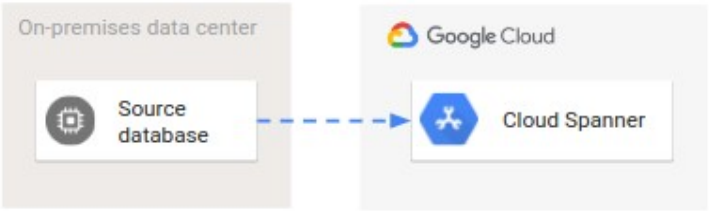
Introduction to database migration and replication

Many enterprises pursue a cloud-first application strategy by building business applications using cloud-based services and technologies. Enterprises compete by shortening time to market for their services and quickly responding to changes in customer demands and market conditions. Business applications that take full advantage of cloud-based computing from end to end are essential for long-lasting business growth.

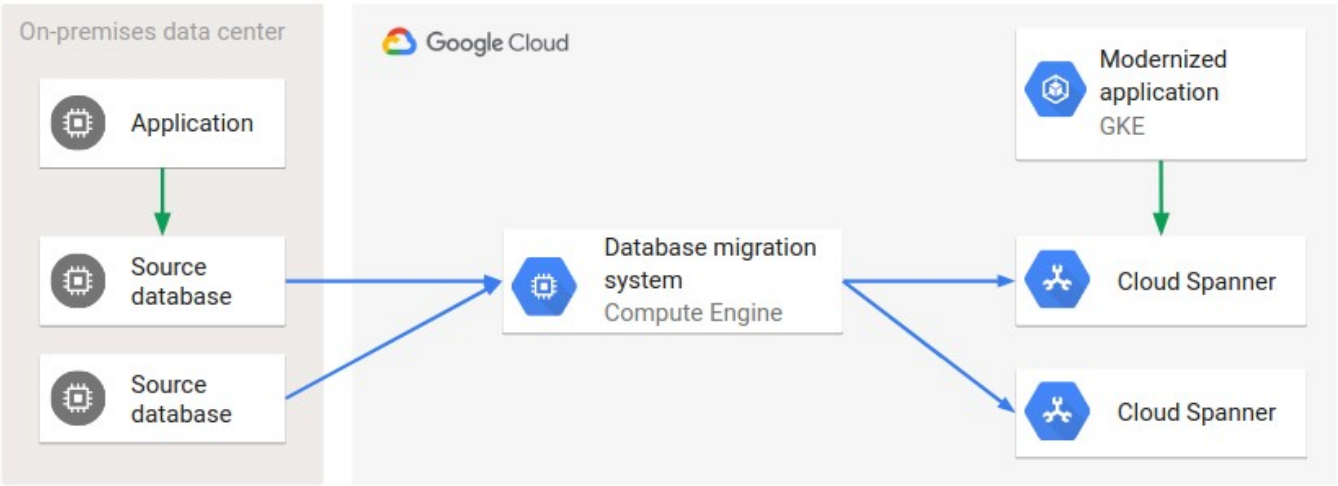
Legacy mission-critical applications introduce several challenges for a cloud migration: migrating the application tier, migrating the underlying database, and maintaining uninterrupted user access to applications during a migration.

Numerous migration products and services can assist with migrating the business application tier, but migrating the underlying databases that serve the business application is often the most difficult challenge.

In practice, a database migration to the cloud can take a few different forms. The following diagram shows the most basic use case—an on-premises database is migrated to a cloud-based database, in this case, Cloud Spanner.



The following diagram illustrates a more complex use case—moving several databases along with an application modernization (/solutions/application-modernization).



In this setup, two source databases are migrated to two target databases. One of the source databases is accessed by an application that accesses the corresponding target database in modernized form (implemented on Google Kubernetes Engine). The second source database also has clients (not shown in the diagram).

Although the database migration system is installed on Google Cloud, the migration system can be installed elsewhere. For example, the migration system might be required to run in the same location as the source database for security reasons.

Lift-and-shift and online database migrations

At a high level, the two approaches to database migration are a lift-and-shift migration and an online migration.

In a lift-and-shift migration, you typically export or back up a source database, move the exported or backup file to the cloud, and then import and restore the file to a target database instance running in the cloud. When the new target database is ready, business applications run in the cloud and access data.

A downside to the lift-and-shift approach is that it requires downtime for both the business application and database. This approach requires careful planning to migrate during low-activity periods. This approach also assumes that you can stop the business application during the migration and restart it after the database is restored in the cloud. Any testing of the application after the database is restored adds to the downtime. Furthermore, the lift-and-shift approach creates an intermediate copy of the database that needs to be secured, moved, stored, and eventually deleted. This aspect adds cost and management complexity. While the lift-and-shift approach might work for certain applications, the requirements of most business-critical applications do not tolerate these costs. For these reasons, an online database migration is a far better approach.

The online migration approach aims for minimal impact on database performance and users of the business applications. The online approach lets you continuously migrate your databases to the cloud, keeping both the source and target databases fully synchronized for months, or even years, while you optimize and test the business application for the new cloud-based database.

Database migration versus database replication

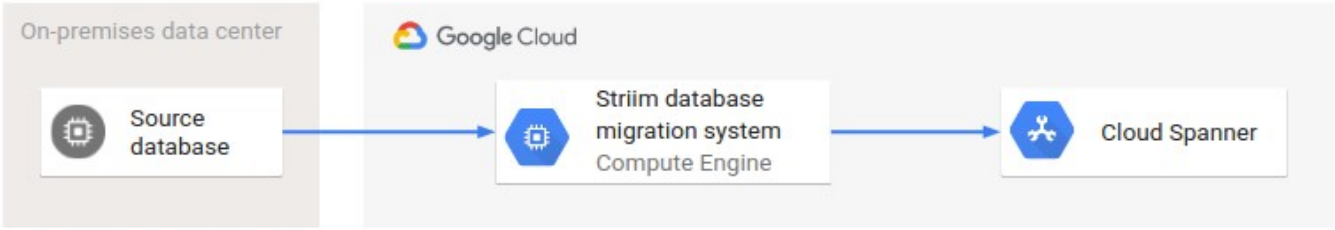
In a database migration to the cloud, the source database is retired when the migration is complete because the cloud database is now the production instance.

In some use cases, the original database instance might continue to run while downstream processing occurs on Google Cloud. In this case, the source database is replicated in Google Cloud. For example, you might have an application that accesses a database that depends on technology that cannot be moved to Google Cloud. At the same time, Google Cloud technology is used for functionality like analysis—for example, replicating to BigQuery. Another example is where a database that contains personally identifiable information (PII) must reside in an on-premises environment while downstream processing that uses obfuscated PII information occurs in Google Cloud.

In these use cases, the original database must be continuously replicated to the operational or analytical target, such as Spanner or BigQuery. The source database is not shut down as it is with a database migration.

Online database migration and replications for heterogeneous databases are typically complex, involving months or even years of hand coding and integrating various services. A more modern and efficient approach to online database migrations is the use of a database migration and integration systems such as Striim (http://www.striim.com).

The following diagram shows a basic deployment architecture of the Striim data integration and intelligence platform with one source and one target database.



Database migration and replication with Striim

Striim, a Google technology partner providing database migration technology, simplifies online migrations by using a drag-and-drop interface to set up continuous data movement between heterogeneous databases. For such migrations to Google Cloud, Striim offers a non-intrusive, streaming extract, transform, and load (ETL) platform that is quicker to deploy and easier to iterate than other solutions.

With non-intrusive change data capture (CDC), Striim extracts real-time data without slowing down the source transactional databases, thus enabling cloud database migrations that have zero downtime and minimal risk. While the data is in flight, Striim can filter, transform, aggregate, and enrich data through SQL queries (https://www.striim.com/blog/2019/01/streaming-sql/) by using a comprehensive streaming analytics engine.

By continuously replicating data between on-premises and Google Cloud databases, Striim supports online replication where mission-critical applications continue running without incurring downtime. With built-in continuous delivery validation, high availability, and failover, Striim also minimizes the risk of data loss.

Use cases for database migration and replication

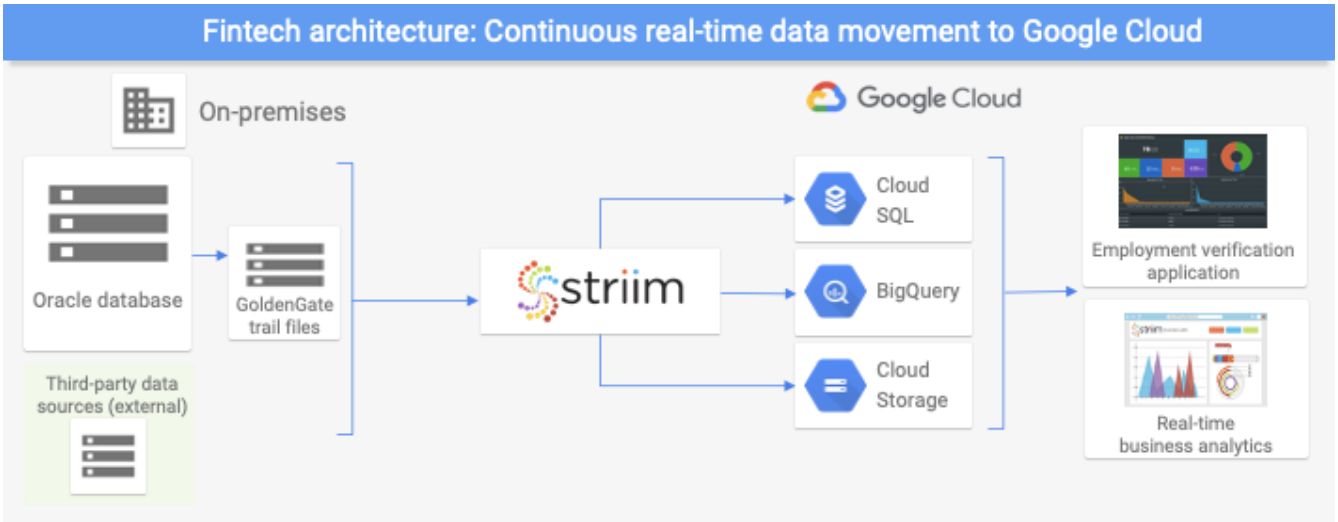
Online database migrations and continuous replication can be applied to a range of industry-specific and technological (industry-neutral) problems. For instance, an industry-specific use might be a financial service that leverages cloud data services and real-time data. A more technology-focused use case might be migrations of reporting databases or phased migrations of operational databases. Striim enables these use cases by providing continuous data pipelines that deliver data in the right format at the right time.

This section discusses using Striim to migrate databases for one industry-specific use case and two technological use cases.

Financial technology

Financial technology (fintech) companies require fast and accurate data. For instance, banking customers want to see transactions and new balances posted to their accounts immediately. Borrowers and lenders want to save time with loan originations. Fintech companies can use Striim to address numerous business requirements such as these.

Consider an automated employee and income verification service that accelerates loan originations. In this use case, Striim integrates real-time data from various on-premises sources to Google Cloud. This automation reduces delays in the employment verification process. The following diagram illustrates the architecture for this use case.

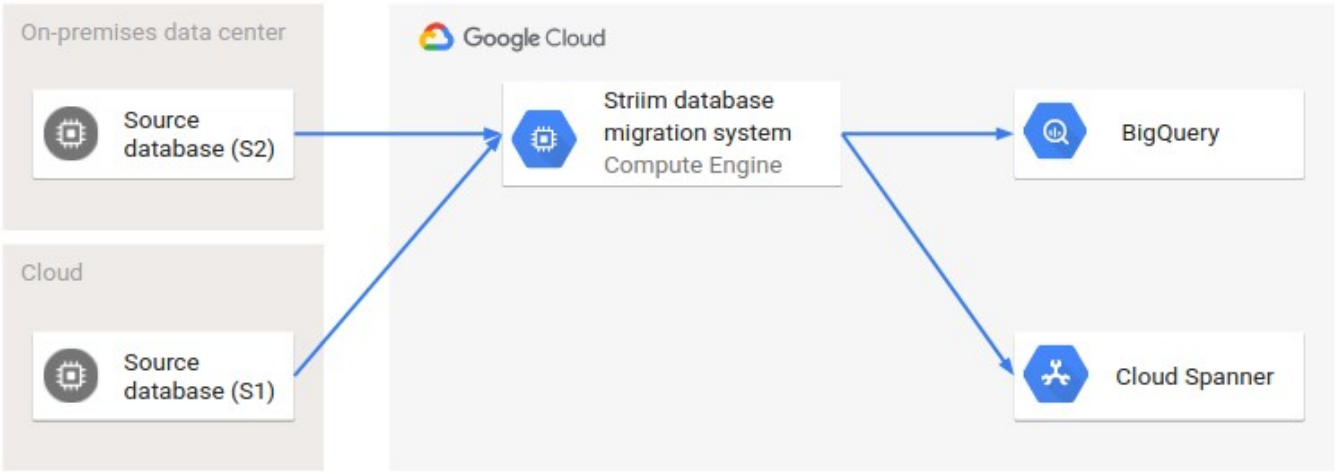


This architecture leverages Striim's real-time streaming data integration. Striim continuously collects data from various external and internal sources—for example, payroll from processors such as ADP, and employee and demographic data from other providers. Striim then feeds the data to various Google Cloud services. Striim continuously ingests data from relational databases by using log-based CDC, accessing Oracle® GoldenGate trail files where applicable, and even from flat files. Striim delivers the aggregated and transformed data continuously to Cloud Storage to power business applications and services built for the cloud.

Offloading reporting

For some technological use cases, a distinction is made between transactional workloads and analytical workloads. Due to the increased performance hit on your source transactional database and increased latency in returning analytical queries, it doesn't make sense to run analytical queries on transactional databases.

To address this distinction, you might continuously deliver a subset of your transactional data to Google Cloud to run further analysis and reporting in BigQuery (/bigquery). The following diagram shows an example architecture.



With the ability to transform and normalize data in streaming data pipelines, Striim can integrate a selected subset of the source transactional data to analytics platforms while the remaining data is continuously moved to one or more Google Cloud data targets.

Phased migration

Large enterprises have massive on-premises databases that hold years' worth of invaluable data. Migrating these databases to the cloud is a large undertaking that can take years. Keeping the legacy databases also comes at a cost—for example, lost opportunity from new cloud database technologies such as BigQuery, Cloud Spanner, and Pub/Sub.

To find a balance between maintaining legacy systems and leveraging the latest cloud database technologies, you can migrate subsets of your data to Google Cloud databases by using Striim in a phased approach and still keep source production databases running. Starting with a small subset also lets your developers test the application tier on the new cloud database without affecting the production on-premises database.

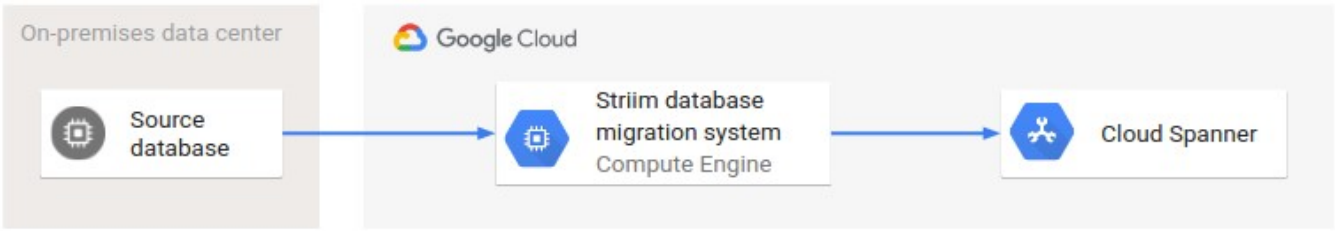
After you ensure that the application tier is compatible, you can slowly increase the subsets that you migrate until the whole database is migrated. This paced migration lets you manage risk and migrate your production systems with minimal downtime.

Deployment architecture

The deployment architectures in this section show the various databases, applications, and cloud services involved in various use cases for database migration and replication.

Basic deployment architecture

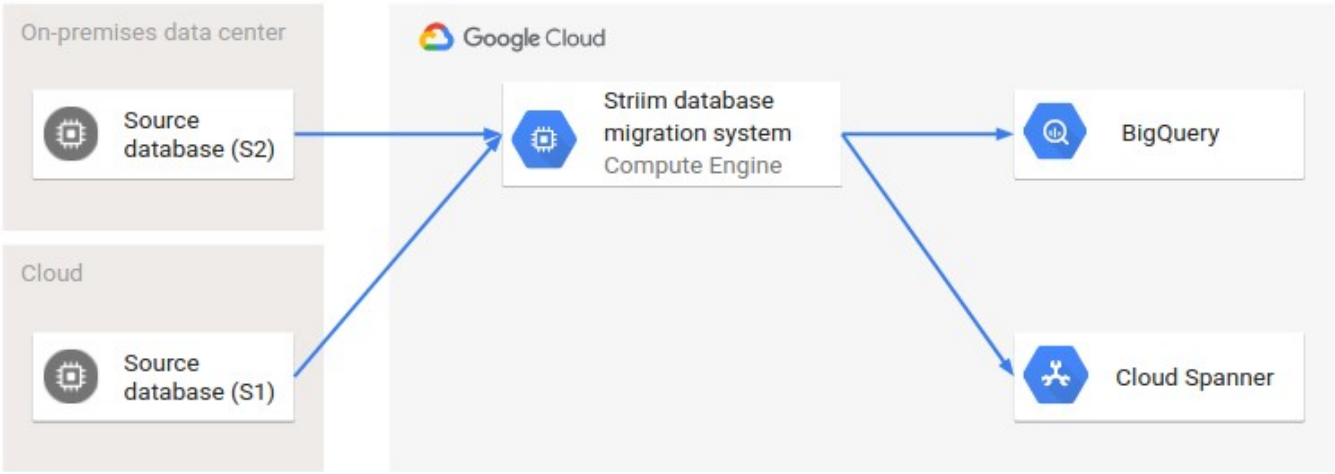
The following diagram shows a simple deployment architecture—the source database system is migrated to the cloud-based database system, Cloud Spanner. This architecture is used throughout this document and works for both continuous replication and full or phased migrations.



In this architecture, the source database system is migrated to the cloud-based database system, which in this case is Cloud Spanner. The database migration system, Striim, is connected to both systems and migrates the data from the source to the target database system.

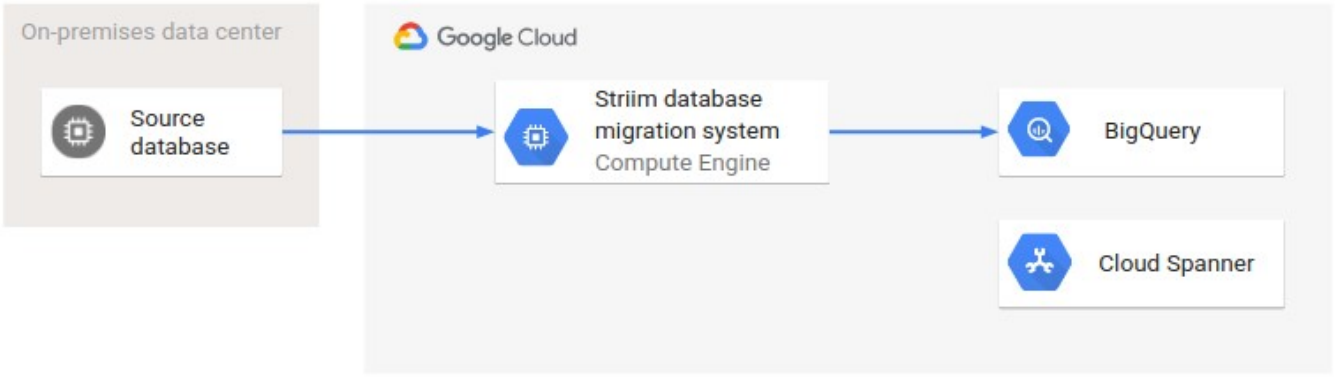
Deployment architecture of medium complexity

The following diagram shows an architecture that involves two source database systems. This architecture is more complex than the preceding architecture and works for database migration and database replication.



The diagram shows the initial architecture with two source database systems. One source is a database system (S1) running in a public cloud that needs to be migrated. The second source is a database system (S2) running in an on-premises data center that remains on-premises while its data and subsequent changes are replicated to BigQuery for analytics. Two target database systems are deployed: Cloud Spanner for receiving the data from S1 (migration), and BigQuery for receiving continuous streams of data from S2.

The following diagram represents the deployment architecture after the database migration is completed. S1 is terminated, and the corresponding Cloud Spanner target is no longer connected to Striim.



This scenario demonstrates that the deployment architecture is not necessarily static. Some use cases, like a one-time database migration, require that the migration system be connected to the systems or services until the migration is completed. Other use cases are more static, as in the case of database replication. This architecture shows how one architecture can serve several use cases.

Complex deployment architecture

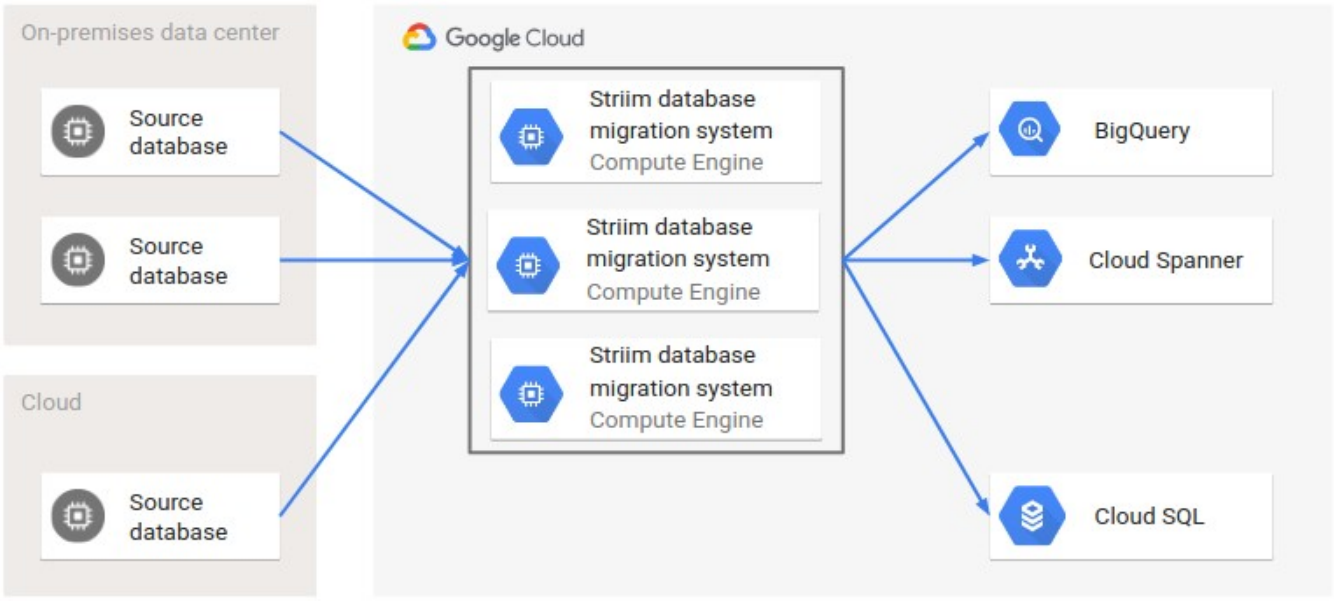
This section describes a migration that is designed as a fallback. A fallback can be necessary when errors that occur after the migration require that you return to processing on the source. The fallback setup ensures that when a migration is reversed, changes in the target are communicated back to the source.

This deployment architecture involves replication and migration and is the most complex of the three architectures discussed in this document.

This architecture also demonstrates how Striim can operate in a clustered environment in order to support increased throughput. The cluster in the following architecture consists of three Compute Engine instances in order to provide sufficient capacity. Each Compute Engine instance is placed in a different zone, which provides high availability for protection against zonal and instance failures.

In this case, two source databases that are located in an on-premises data center are replicated to the cloud. One database is replicated to Cloud Spanner, and the other to BigQuery. In addition, the migration moves data from a database in a public cloud to a Cloud SQL instance.

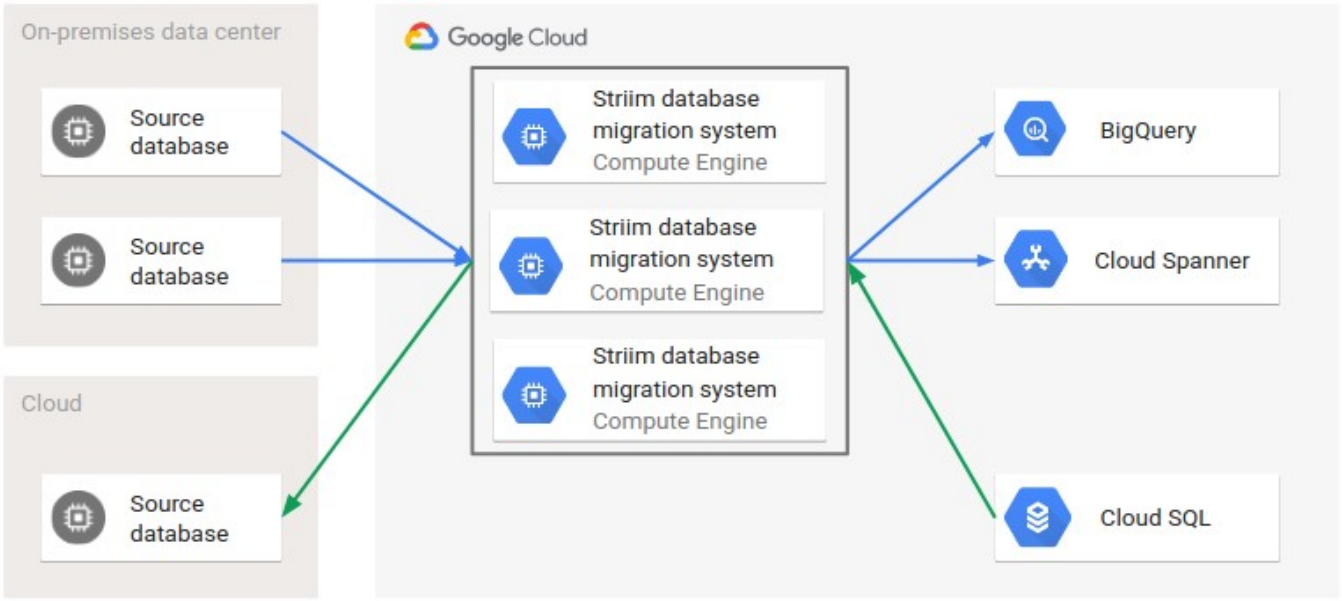
The following diagram shows the deployment architecture before the migration completes.



Three source databases are connected to Striim, and Striim migrates and replicates data to three target databases. The two source databases in the on-premises data center are replicated to BigQuery and Cloud Spanner, and the source database in the cloud is migrated to Cloud SQL.

After the migration completes, the flow is reversed from Cloud SQL to the database in the cloud to prepare for a possible fallback. In order to ensure that any changes on the target are also available on the source, the reverse migration must be set up after the original migration.

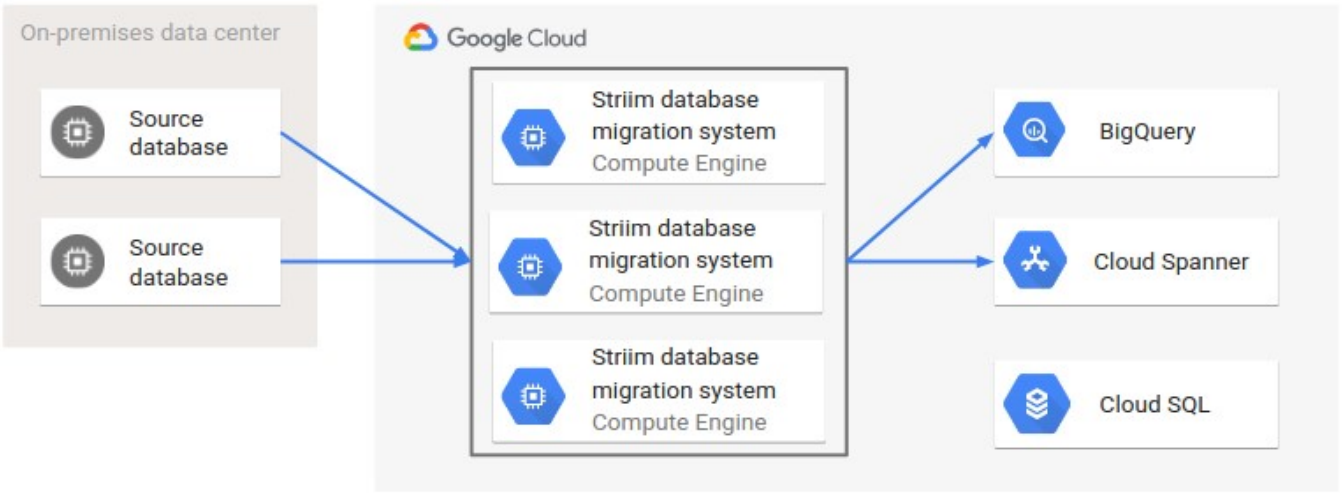
The following diagram shows the configuration change for a potential fallback.



The direction of the data flow is reversed between the original source cloud database and Cloud SQL. After the migration is completed and the need for a fallback is ruled out, the related systems in

the migration are disconnected, including Cloud SQL and the source database in the on-premises data center.

The following diagram shows how this deployment architecture supports the continuous replication use case.



The two source databases in the on-premises data center are continuously replicated to BigQuery and Cloud Spanner.

As this discussion shows, this deployment architecture can support various uses cases, regardless of complexity. The configuration can also be dynamically changed—for example, when the direction of the data movement is reversed.

This architecture can also scale to help support an increased load or spikes.

Complexity of deployment architecture versus migration specification

In the preceding architectures, the two dimensions of complexity include the following:

- Number of database systems connected to Striim
- Change of deployment architecture over time as use cases are completed

Complexity can also be considered in the context of the migration specifications themselves, independent of the deployment architecture.

For example, if an on-premises MySQL database is migrated to a MySQL database in the cloud, and if the schema and data are identical, the migration specification is straightforward, because the specification is a copy. However, if the migration is from an Oracle database to a Cloud Spanner database, the migration complexity significantly increases, because the source and target schemas are different, and the data must be transformed during migration.

The following table provides a high-level approach to determining the complexity of a migration specification. Features are listed along with an indication of migration or replication complexity. You can use the column titled **Your use case** to assess your requirements.

Feature	Similar	Different	Your use case
Source and target database system	Low complexity	High complexity	
Source and target database schema	Low complexity	High complexity	
Source and target database data	Low complexity	High complexity	
Data partitioning (if present)	Low complexity	High complexity	
Data separation or consolidation	High complexity	High complexity	

Migration complexity is a spectrum based on what use case is implemented. For your use case, you can determine overall complexity based on the ratio of low complexity to high complexity for the features of your use case.

Other considerations can play a role in determining complexity, especially in advanced use cases. For example:

- Will several of your source databases be consolidated into a single target database?
- Will data from several of your source databases be redistributed into a set of target databases —for example, re-sharding or re-partitioning?
- Will data be enriched or reduced during migration in order to provide a well-curated dataset in the target system?

Each feature that you add to your use case increases the complexity, which a system like Striim is built to support.

What's next

- Learn more about Striim's platform for streaming integration with intelligence (http://www.striim.com/), schedule a demo (https://go2.striim.com/demo-region-select) with a Striim technologist, and read the blog (http://www.striim.com/blog).
- Check out Striim on Google Cloud Marketplace (https://console.cloud.google.com/marketplace/browse?q=striim) for different database migration endpoints and different pricing models.

- Understand the concepts and principles of database migration by reading Database migration: Concepts and principles (Part 1) (/solutions/database-migration-concepts-principles-part-1) and Database migration: Concepts and principles (Part 2) (/solutions/database-migration-concepts-principles-part-2).
- See the resources and guides offered for database migration in Google Cloud (/db-migration).
- Try out other Google Cloud features for yourself. Have a look at our tutorials (/docs/tutorials).

Last updated 2020-06-19.