SLK WHITEPAPER

Legacy Data Migration

What does it take to move from Teradata to Cloud data stores?

Contents

Abstract

For decades, Teradata has been the preferred choice for an on-premise database system, catering to enterprises who want to manage and analyze large volumes of data. However, as storage technologies evolved, they ushered in the age of Cloud Data Lakes such as Snowflake (modern cloud computing data warehouse).

These new-gen systems, armed with modern data analytics capabilities, separate the computations from storage, enabling enterprises to scale their infrastructure automatically and instantly. Many enterprises have begun to evaluate using them in their modern applications. However, the transition from Teradata is not an easy task.

To work with the existing ecosystem, enterprises invariably built pipelines to push data to Teradata. These included customized visualization tools to pull information out of Teradata, designed client applications dependent closely on data from Teradata, and built tooling around Teradata. Therefore, migration from Teradata to new-gen cloud data stores might seem a daunting task.

But that's where SLK comes in! Having worked on Teradata versions over the years, we have found an ingenious way to migrate to any cloud storage in a fast-paced manner with an ecosystem to fast track with automation.

Migration Journey – Steps to Migrate from Teradata to New Gen Cloud Stores

Migrations are usually complicated and can get messy if not planned and executed properly. We recommend a carefully thought-out step-by-step approach to migrate your data.

Step 1:
Understand Teradata Environment & Analyze Current Teradata Usage

Start with analyzing your current Teradata implementation to get a full 360-degree view of your Teradata ecosystem. This process involves finding answers to the following questions:

- What are the priority applications/LOBs?
- Which are the data stores/databases and tables that need to be migrated?
- Which users, roles, and applications can access these databases and tables/objects?
- How is data getting loaded into these tables/objects — which scripts and applications are responsible for data loading?
- How is data in these tables/objects getting used — which scripts and applications pull data from these tables/objects?

Step 2:
Design Integration Plan

Now that you are done with understanding and documenting your current Teradata usage, let us lay out an execution plan for the migration.

Instead of an all-in-one-go approach to migration, we recommend a phased approach as follows:

- Do a quick Discovery to finalize the scope. SLK having tribal knowledge is well equipped to take this project up without spending weeks in discovery phase
- Plan to Migrate either a data store or low impact tables, databases, and applications first, get some success under the belt, and then move-on to the complex migration tasks

No matter which approach you take, by the end of this step you will have a full plan of action for the migration in place.

- Take the output from the previous Analysis step, information, databases and migrations into logical phases
- Phase 1 will repeat based on priority with focus on a particular application, data store or low impact table cluster
- It is best to plan a full vertical slice migration – i.e. end-to-end ingestion, migration, and consumption together This helps isolate issues at every stage early in the process
- Identify tools that can help speed up the migration process. Hand-coding is generally not recommended
- Try out Data Pipeline Automation tools like DataSwitch, Self-Service Ingestion, intelligent Data Management (iDM), Automated Data Validation, and Audit Control for executing the migration. They can reduce time to market substantially by automating a significant portion of the migration and re-tooling. This becomes especially beneficial for repeatable steps in the phased migration approach

Please note that we recommend a focus on outcomes versus a focus on tools. It is generally recommend to not hand-code but hand-correct to stop mess-in and mess-out. Tools used here are mostly one-time use during the migration.

- Once migrated all scripts are enabled in native cloud environment and its tooling environment to take out any other tool dependency

Step 3:
Setup Cloud Data Stores, Snowflake

Now that we have a clear plan of action, let’s get started on executing the migration. First up, you want to set up your Cloud Data Lake and Snowflake accounts and sizing to meet your needs. SLK has built a calculation sheet to fast track this configure the following on Cloud Data Stores:

- Create a Cloud Data Lake on cloud data stores such as Azure Data Lake or AWS S3
- Create structured databases and warehouses on Snowflake
- Create users and accounts on these data stores

Step 4: Migrate Data Definition Language (DDL) and Enable Data Governance

Now that you have primed these cloud data stores with accounts and databases, you can start creating the database structure in stores like Snowflake. This typically involves using DDL exports from Teradata, converting them to Snowflake compatible DDL and executing it.

However, an intelligent Data Pipeline platform like DataSwitch can handle this automatically, saving you a tedious and potentially error-prone process. With this, all you need to do is select the Teradata tables and data fields that needs to be migrated to Snowflake.

DataSwitch will automatically create Snowflake DDL-compliant tables and handle the DDL migration based on the data coming from your Teradata database.

Step 5: Load Data (One Time) from Teradata to Snowflake

After Step 4, your Snowflake table should be ready to handle data from Teradata. Now you can load data from Teradata into Snowflake!

If you are running Teradata on premises, it is advisable to push data first to a cloud storage solution like AWS S3 or Azure DataLake before loading it into Snowflake.

This step is another reason for considering an intelligent Data Pipeline platform like DataSwitch. DataSwitch can create scripts that can be run in cloud-native environments using their tools to extract data from Teradata automatically, load it into an intermediate cloud storage if needed, and subsequently load the data into Snowflake.

Step 6: Load Data (Ongoing) from Data Sources to Snowflake

In Step 5, we replicated all historical data from Teradata into Snowflake. To decommission Teradata completely, we now need to connect Snowflake directly with the sources of data that are writing into Teradata.

We recommend connecting each data source to Snowflake while it is still writing to Teradata. This will allow you to test applications in parallel and ensure a seamless transition once you disconnect the source to Teradata link.

Step 7:
Update Applications to Consume Data from Snowflake

Once you have ensured that all your data is now being written into Snowflake, start tuning all your applications that were dependent on Teradata to read data from Snowflake. We recommend running two parallel versions of each application during the migration — one that continues consuming data from Teradata and another that is based on data from Snowflake.

Customizing scripts and applications can be complicated, and it might require significant re-engineering. This is where a Data Pipeline platform like DataSwitch comes in handy. The platform creates scripts to be run using cloud-native tools in their environments using native data loading tools, thereby reusing the existing pipelines!

Step 8:
Validation – Run QA with Parallel Stacks

Once you have configured parallel end-to-end stacks incorporating data flowing in and out of Snowflake and Teradata, respectively, a comprehensive QA cycle should be run to compare functionality across two systems.

In addition to validating the applications, you should compare row counts, sums of numeric fields, and hashes of string fields to identify discrepancies. This step can be easily automated with our automated data validation tools. We recommend doing validations in two phases:

Validation Phase 1: Teradata – Primary, Snowflake – Secondary

In the initial phase, you continue with the Teradata stack as primary and Snowflake as secondary. Run this phase for a short period of time. During this period, run the comparator as a scheduled process to monitor for differences. If no differences are found, you can move on to the next phase of validations.

Validation Phase 2: Snowflake – Primary, Teradata – Secondary

In this phase, we flip over the stacks and make Snowflake as the primary data system and Teradata as the secondary. Just like the previous validation phase, run the comparator as a scheduled process to monitor for differences. If no differences are found, the Teradata system can be decommissioned.

Step 9:
Decommission Teradata

Finally, we are done with migration! With each one of the steps completed and all the applications are running on Snowflake, inform all your Teradata users about their new Snowflake accounts and other changes. Turn off all Teradata load processes and revoke access to Teradata.

Conclusion

This whitepaper introduced you to a comprehensive step-by-step plan for migrating from Teradata to Snowflake designed to avoid any hiccups pre or post migration.

As we saw, a big chunk of the migration process can be achieved more quickly and reliably with intelligent, modern DataPipeline Automation tools like DataSwitch, Self-Service Ingestion, intelligent Data Management (iDM), Automated Data Validation, and Audit Control tools.

While systems such as Teradata have been around for a decade, new-gen systems are slowly replacing them. These advanced storage technologies let you manage and analyze large volumes of data using the latest cloud data analytical tools. Their scalability and flexibility in usage are something enterprises look forward to. Therefore, migration processes that ease the transition from on-premise data systems to cloud systems are the need of the hour. And this is where SLK excels in, with our fool-proof automation systems focused on secure and fast-paced data migration adhering to enterprise standards.

About Us

SLK is a global technology services provider focused on bringing AI, intelligent automation, and analytics together to create leading-edge technology solutions for our customers through a culture of partnership, led by an evolutionary mindset. For over 20 years, we've helped organizations across diverse industries - insurance providers, financial service organizations, investment management companies, and manufacturers - reimagine their business and solve their present and future needs. Being A Great Place To Work Certified, we encourage an approach of constructively challenging the status quo in all that we do to enable peak business performance for our customers and for ourselves, through disruptive technologies, applied innovation, and purposeful automation. Find out how we help leading organizations reimagine their business at https://www.slksoftware.com/