Authors : Anil Wadhai & Vishal Umredkar

Background

Legacy modernization is an open-ended journey to streamline process efficiency, improve business performance and create new ways of serving customers. Data migration is a core need of legacy modernization / digital transformation.

Data migration is a process of transferring data from one system to another. It can be done in real-time or in batches depending on the size and complexity of the data. Data migration can be used for system merging, application upgrades and disaster recovery.

Types of Migrations

There are four main component migration
1. Data Migration
2. Application Migration
3. Business Process Migration
4. Infrastructure Migration (On prem to cloud , on prem - on prem , cloud to cloud , cloud to on prem)

a. Storage
b. Database
c. Data Center
d. Cloud Data

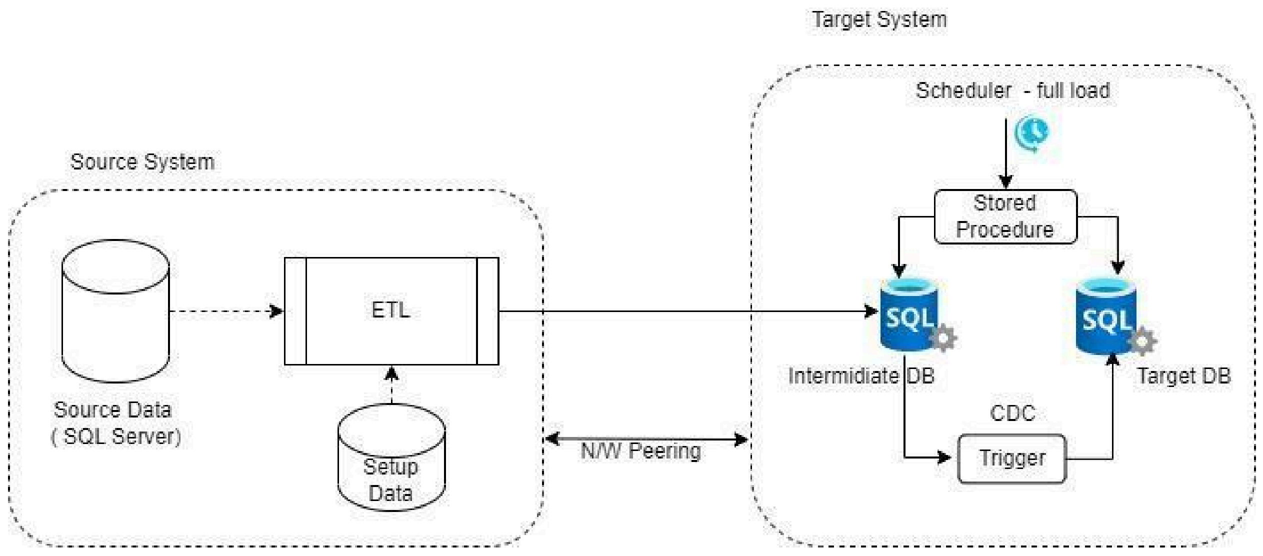
Data Migration is one of the final steps as part of the whole migration life cycle & we are going to cover data migration where source & target systems need to be live as well as system needs to perform complex transformation while migration.

Approaches

Data migration is an important process for businesses to move data from one system to another. It involves transferring data from one source to another in a secure and efficient manner. Data migration can be done in real time or batch mode depending on the needs of the organization.

Real time data migration allows for more flexibility and faster response times, while batch data migration is more suitable for large volumes of data that need to be moved over a longer period of time. There are several tools available that can help with the process, such as analytics engines and streaming data processing engines which can help move data from on-premises / Cloud systems to cloud without any downtime. System merging is also an option when migrating multiple systems into one unified platform.

Batch Load

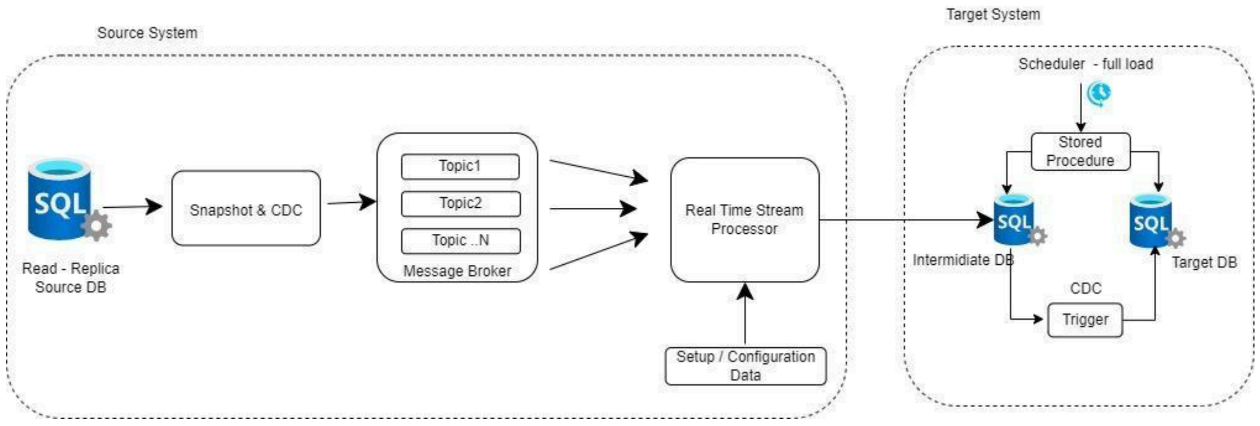


The Batch Load process is described below:

- An ETL job is created which is scheduled to run on a periodic basis. It extracts the data from the source, transforms the data, if required, and loads the same into an intermediate datasource in the target environment. The configurations of the ETL job and expected transformation are stored in another schema in the source database.

- After the data lands in the intermediate database, either of the following two routes can be invoked:
  - An initial load can be invoked through a periodic scheduler which will reload the data in the target database using a stored procedure
  - Assuming the initial load is restored, the CDC(Change Data Capture) trigger can be enabled which will track each and every changed happening on the tracked entities and make sure that all the incoming changes are reflected in the target database.

Real Time System Merge



This approach can be termed as a “Trickle” where the target systems are gradually brought in sync with the source systems. CDC plays a major role in this approach

- A CDC connector keeps reading the change event logs on the source database and converts each change into a message which is delivered to an intermediate Message Broker like Kafka
- Change events corresponding to one entity will be available in respective topics created for each entity
- A Real time stream processor consumes these messages and converts the change events into actual DML (Data Manipulation Language) statements which are executed against an intermediate database.
- After the data lands in the intermediate database, either of the following two routes can be invoked:
  - An initial load can be invoked through a periodic scheduler which will reload the data in the target database using a stored procedure
  - Assuming the initial load is restored, the CDC(Change Data Capture) trigger can be enabled which will track each and every changed happening on the tracked entities and make sure that all the incoming changes are reflected in the target database

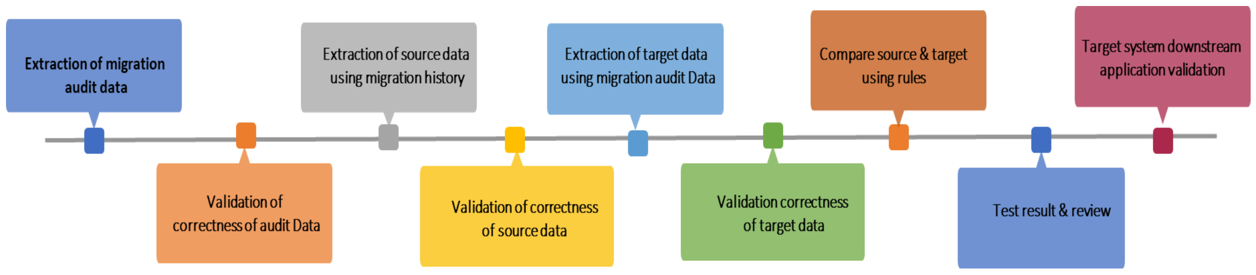
Use Cases

Following is an exemplary list of use cases which require Data Migration:

- Retiring legacy applications and their data stores
- Adopting Digital transformation by moving to cloud-based systems
- Adopting newer versions of the software systems which require change in the data sources
- Implementing Post-Merger Implementation (PMI) and Merging of two systems
- Adapting to data security regulatory changes

Migration Validations

Data Migration Validation Strategy


Extraction of migration audit data
Validation of correctness of audit Data
Extraction of source data using migration history
Validation of correctness of source data
Extraction of target data using migration audit Data
Validation correctness of target data
Compare source & target using rules
Test result & review
Target system downstream application validation

- Testing script shall use the input for which batch/members/date wants to perform validation, using the input script will extract migration audit data & perform the validation of audit data such as null check, duplicate check, and field values.
- After successful validation of audit data, the script will extract the source data using audit data reference & perform the validation of source data such as null check, duplicate check, missing data, field values & mapping.
- After successful validation of source data, the script will extract the target data using audit data reference & perform the validation of target data such as null check, duplicate check, missing data, field values & mapping.
- After successful validation of target data, the script will transform the source data & compare it with target data and validate the count, field values, correctness of pre-migration field values, correctness of post-migration field values, and generate the test report.
- After successful validation of source & target data, the team will validate all downstream applications. Here the team should use existing tools & test cases to validate the downstream application.

Validation Checklist

Pre-Migration

- Test Strategy document validation
- Data Migration approach validation
- Data Migration scope validation
- Source Target Mapping Validation
- Functional / Non-functional Requirement validation
- Tool / Techstack validation
- Performance test strategy validation
- Rollback test strategy validation.

Post-Migration

- Migration Status / Audit validation
- Data Quality (Duplicate, Mis-match)
- Data loss
- Transformation validation
- Field mapping validation
- Target system Upstream/Downstream application validation

Validation Steps

Understanding of the ecosystem, source system & target system
After the formation of the Testing team, the team should go through the documentation to understand the ecosystem, migration approach, migration scope, set up a channel for test validation confirmation, etc.

Prepare Testing Environment
Prepare access to source & target system, identify & prepare machines for test execution.

Setup of testing tool/framework
Migration includes heavy work of data validation, we are suggested to use a readily available combination of tools for data validation i.e. pandas (https://pandas.pydata.org/), AWS Glue (Pyspark) & GreatExpectation (https://greatexpectations.io/)

Pre-Migration Testing
- Source & Target field mapping
- Source & Target data validation
- Setup data & rules validation
- Downstream application testing

Migration Testing
- Source & Target field mapping
- Source & Target data validation
- Transformation logic Testing
- Downstream application Testing

- Setup data & rules validation
- Migration Audit Testing
- Sampling Testing
- Performance Testing
- Rollback Testing

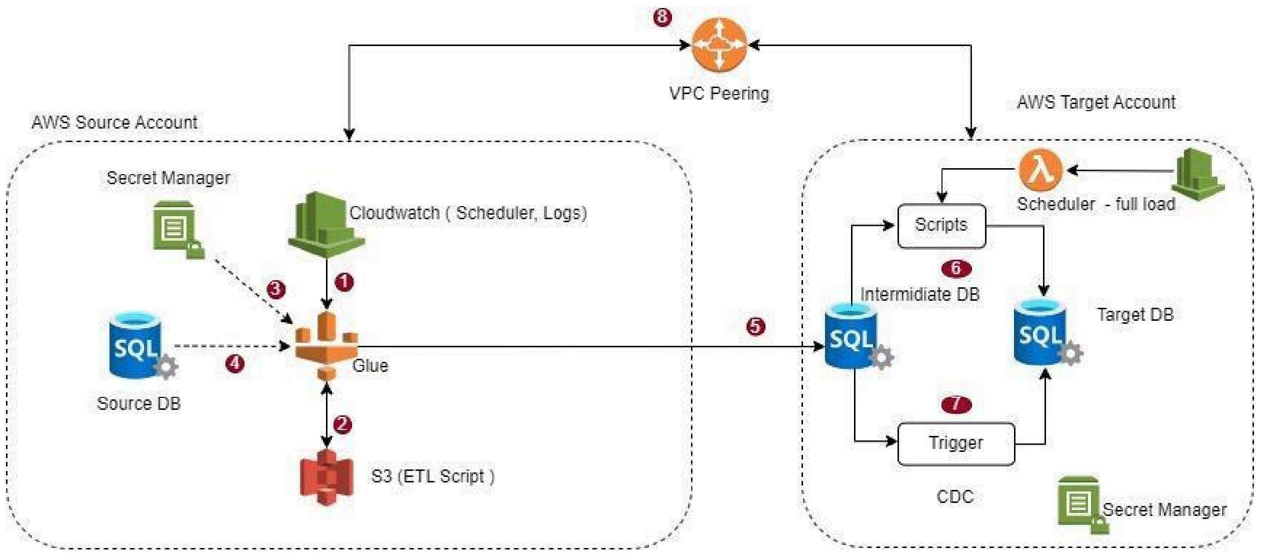
Post Migration Testing
- Migrated Data Validation
- Downstream application Testing

Risk and Mitigations

- Migration may impact source/target system performances during migration activity so needs to choose a less busy time window
- In case of data corruption or data integrity issues, a Rollback plan should be in place

Sample Implementation

Batch Load



AWS Glue - AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data. Here Glue will extract data from the source by referring to the Migration_Status history, perform transformation & load to the intermediate database. Once the load successfully completes it will update the status in Migration_Status history.

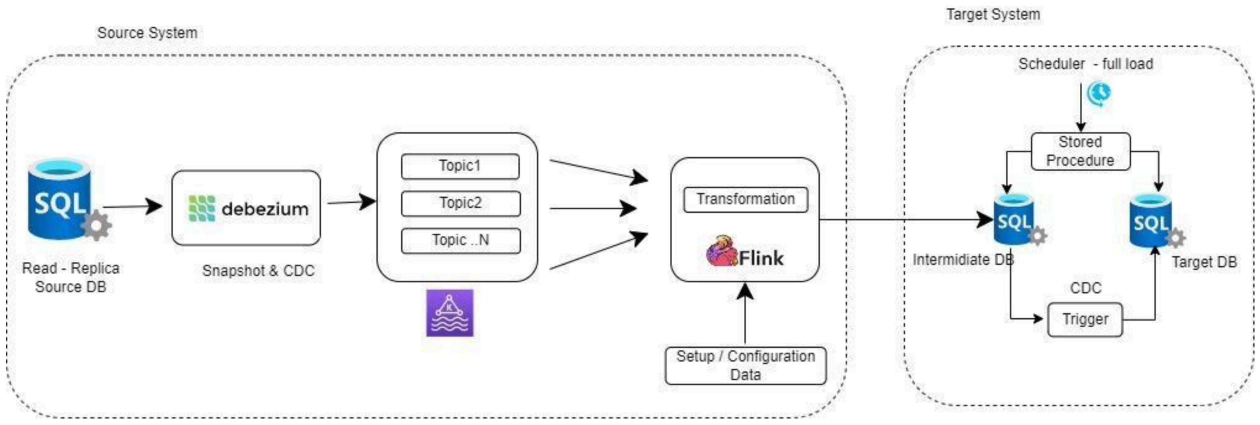
AWS S3 - AWS Simple storage service will be used to store all ETL script & log storage.

AWS Secret Manager - AWS Secret manager securely encrypts and manages the secrets. It shall be used to manage source and target database credentials.

AWS Cloudwatch - CloudWatch Events Rule shall be used to trigger the ETL script in a scheduled manner. CloudWatch Logs shall be used for monitoring Glue logs

Script - Script will pick the changed data from the intermediate DB and will update the target DBs.

Real Time / Streaming Load



Debezium - Debezium is a distributed platform that turns your existing databases into event streams, so applications can see and respond immediately to each row-level change in the databases. Debezium is built on top of Apache Kafka and provides Kafka Connect compatible connectors that monitor specific database management systems.

Kafka - Kafka is a distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. Core capabilities of Kafka are High throughput, Scalable, Permanent Storage & High Availability. Various clouds provide managed service for Kafka also you have the option to self host using open source or Confluent.

Flink - Flink is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Flink has been designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. We do have different options for Flink cluster deployment using Flink Kubernetes operator, AWS KDA etc.

Script - Script will pick the changed data from the intermediate DB and will update the target DBs.

Conclusion

Migration approach & sample examples targeting only data migration with consideration of source & target are RDBMS having terabytes of data to migrate, though tools / techstack may vary based on use case , env , data volume etc.