



Plan your migration

Plan your migration

Aside from any application migration, your **database** migration generally follows this process:

- schema conversion
- data extraction
- data transformation
- data load
- testing and post migration activities

Schema conversion

If you are migrating to another *brand* of database, for example from DB2 LUW to PostgreSQL, or from Oracle to PostgreSQL, you will need to convert your Schema to the format of your *Target* database. As discussed in [Migration tools](#), we recommend automating this process as much as possible. This will speed up the process and reduce errors. While carrying out a manual conversion process it's easy to miss items that need converting, or make errors such as *typos*, leaving in *cut-and-pasted* values that you actually intended to alter, and so on.

- The [AWS Schema Conversion Tool \(SCT\)](#) is designed to do the schema conversion for you. It generates a report before you even apply the changes, so there will be no surprises with the *Target* database format it will create for you.
For example, the Report on our conversion of DB2 LUW to PostgreSQL, showed that **Foreign keys** had to be manually created in PostgreSQL. This is because PostgreSQL needs a unique index on the Foreign key columns.
- The [AWS Database Migration Tool \(DMS\)](#) is sophisticated enough to natively convert schemas for *most* databases, so you may even be able to skip using SCT.

You can follow our example that uses [SCT for schema conversion](#)

Data extraction, transformation and loading

Again, as discussed in [Migration tools](#), we recommend automating this process as much as possible. You can of course, carry out a manual data unload/load process.

- The [AWS Database Migration Tool \(DMS\)](#) is designed to move your data between your *Source* and *Target* databases. It won't be perfect, but it does log or report any issues it encounters. For example:
 1. PostgreSQL is **case sensitive** Table names need to be converted from upper case (DB2 LUW) to lower case (PostgreSQL) as part of the migration.
 2. When doing an incremental load, we found DMS couldn't load the parent tables before loading the child tables, resulting in **foreign key** issues. We overcame this by updating the Relational Data Store (RDS) parameters in DMS. We selected the order of table loading, instead of using the table name)default.

Testing and post-migration activities

Test before, and after implementation

After your migration is complete, do your testing before, and continue after your "go-live" production implementation:

- verify any undiscovered **data feeds** haven't broken, either causing your system problems by omission, or problems for hitherto unidentified consumers downstream.
- verify you have satisfied customers. Identify accessibility issues to the new system, and its utility. A range of problems can manifest themselves, such as vague documentation, slow response times, missing data, errors saving data, problems uploading/downloading files.
- check the expected system logs are being produced. Check the logs produce the wanted CSAM compliance information, but aren't swamping the reporting systems (Splunk, Dynatrace) with unnecessary data.
- check your new support model is working. This includes people and processes. For example, has the Service Desk been able to find who to call about any issues? Is your documentation accurate and easy to follow? Have your support Staff identified any extra training they require?
- ensure your **back-up** and **restore** strategy works. Once you de-commission your on-premises resources, there may no longer be another source for backups available.

Post implementation clean-up

After your "go-live" on the Cloud, with your new set up **working reliably**, remember to clean up any resources that are no longer required. This will typically save BNZ a lot of expense:

- cancel unneeded **software licences**
- de-commission **on-premises resources** (VMs, Servers, storage)
- verify any unwanted **data feeds** are switched off
- update or archive documents and procedures referring to any component that has been replaced or removed
- shut down any test servers and databases you have used in your Landing Zone in the Cloud for your migration and testing work to save money. It costs over \$300 a month to run a Virtual Machine (VM) in the Cloud.

 [Edit this page](#)