Database migration to the cloud and synchronization

As part of the CMR Cloud prototyping activity we spent some time testing out the AWS Database Migration Service (DMS). Based on our testing we determined that using DMS can save us from having to write a significant amount of custom code to perform database synchronization.

DMS Limitations

- Replication cannot replicate BLOB columns for Oracle 12c databases. (Full initial migration does correctly copy the BLOB data, just not the replication task afterwards).
- Does not migrate everything:
    - "The migration service only migrates tables, data, primary keys, and when necessary unique indexes. Secondary indexes, constraints, defaults, sequences, accounts etc. need to be migrated using other means: in a homogeneous environment, native tools such as datapump will work, in heterogeneous environments the Amazon Schema Conversion Tool is a good option."
- Running with Full LOB mode is roughly 100 times slower than using Limited LOB mode with truncation at 32K.
- Redo transaction logs on the source database must be 4GB or smaller.
- Can only migrate tables with supplemental logging turned on.
- See Oracle as a Source Limitations for a full list.

Custom Code

Prior to performing the migration we will need to modify the custom code:

- Add primary keys to all database tables where possible (14 tables in BUSINESS schema, 4 tables in METADATA_DB).
- Create a migration to add all secondary indexes, constraints, and set DEFAULT values.
- Create a script to add all database users and grant permissions.
- Create a script to copy all database sequences.
- Create a script to compare counts on all database tables.
- Modify bootstrap to allow indexing of all data later than a provided date and time.

Migration Plan

For each environment:

1. Setup an Oracle RDS instance.
    a. Setup all configuration parameters to match current setup (such as 11g optimizer mode).
2. Create database users and grant permissions in new Oracle RDS instance.
3. Reconfigure transaction logs on source database such that each log is 4GB (currently ops logs are 10GB).
4. Configure transaction archive logs on source database to not be deleted immediately after an Oracle backup.
5. Setup supplemental logging on all source database tables.
    a. Note that tables without primary keys will need to have all columns logged.
6. Open GSFC firewall to allow AWS DMS to access the source database.
7. Setup DMS replication instance.
8. Setup DMS migration source and target endpoints.
    a. Ensure to set the charLengthSemantics=CHAR parameter for the business schema target endpoint.
9. Setup DMS migration tasks*.
    a. Business schema - SERVICE_OPTION_DEFINITION_FORM migration by itself with Full LOB mode
    b. Business schema - all other tables with Limited LOB mode 32K
    c. Metadata_db - Limited LOB mode 32K and skip GET_CONCEPTS_WORK_AREA
10. Monitor initial migration
    a. Ops metadata_db will take between 2 hours and 10 minutes and 5 hours. Business schema took between 3 and 10 minutes in workload with 15 million entries in the security tokens table.
    b. SIT and UAT should take minutes.
11. Add secondary indexes, constraints, and update tables to set any DEFAULT column values.
12. Setup database sequences.
13. Index all data into Elasticsearch.
14. Verify counts and retrieve metadata for sample collections and granules.

For SIT and UAT we will rerun the migrations daily since replication will not work. For workload we will rerun migrations as needed. For operations we will perform additional testing to ensure replication is working as expected as well as setup a job to reindex everything ingested after a given date to ensure that replicated data is being indexed into Elastic.

* For operations environment we will setup tasks to perform initial load and replicate changes. For all 12c environments (SIT, UAT, workload) setup to only perform the initial load since BLOB data cannot be replicated.