OBJECTIVE: To develop a formalized process to logically refresh EDW data and schemas to facilitate effective development and testing, using a sequenced, cascading approach in consideration of current overall environment state.

This proposed process is in strict consideration of a (one-time) baseline setting project for the production environment as a prerequisite

- Set baseline for PROD:
  Copy (**backup / restore**) to NON-PROD / INTEGRATION
  Copy (**backup / restore**) to SANDBOX1
  Copy (**backup / restore**) to SANDBOX2

- **Data** in PROD is populated through **ETL processes** using **various products and techniques** (*ADF, SSIS, SQL Agent, Alteryx, Etc.*)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in SANDBOX1 environment to NON-PROD / INTEGRATION environment using **GitHub**. (<u>SCHEDULED OR MANUAL</u> *Interval TBD*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in SANDBOX1 environment to NON-PROD / INTEGRATION environment using **GitHub**. (<u>SCHEDULED OR MANUAL</u> *Interval TBD*)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in SANDBOX2 environment to NON-PROD / INTEGRATION environment using **GitHub**. (<u>SCHEDULED OR MANUAL</u> *Interval TBD*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in SANDBOX2 environment to NON-PROD / INTEGRATION environment using **GitHub**. (<u>SCHEDULED OR MANUAL</u> *Interval TBD*)

- Engineer / Adopt process that promotes **DDL** code that is created / changed in NON-PROD / INTEGRATION environment to PROD environment using manual **Release Process**. (<u>MANUAL</u> (controlled) *Interval TBD – (Weekly release window)*)

- Engineer / Adopt process that promotes **DML** code that is created / changed in NON-PROD / INTEGRATION environment to PROD environment using manual **Release Process**. (<u>MANUAL</u> (controlled) *Interval TBD – (Weekly release window)*)

- Engineer / Adopt processes that refresh lower SDLC environments with **DATA** from PROD via either manual or automated "Copy Back" process and <u>subsequently</u> promotes **DDL / DML** to NON-PROD / INTEGRATION environment from SANDBOX environments using GitHub. This will facilitate an integrated testing environment (*i.e. fresh data, which is then overlaid with latest code versions.*)

LEGEND:

= "*Multiple*"

Synchronized

NOT Synchronized

MAY BE Synchronized <u>UNKNOWN</u> LOE

CAN BE Synchronized **KNOWN** LOE

Promotion / Release Process **KNOWN** LOE

Database Backup / Restore Process KNOWN LOE

Programmatic Process UNKNOWN LOE

PROD

SERVER Azure SQLMI

DBMS Azure SQLMI

DB

ETL

- ADF
- SSIS
- SQL Agent
- Alteryx
- Etc.

"*Latest Stable Version of SQL Server*"

MS Azure SQLMI
#Azure Tenant
#Azure Subscription
#Azure VNets
#Azure Subnets

Instance Layer Configs
Server Collation
Accounts
Agent Jobs
Enabled Features

DBs
Compatibility Levels
DB Collations
Data Variations
Code Variations
ETL Variations

PROD:

Data populated by ETL / Business Activity, **then** overlaid with DDL / DML **promoted from NON-PROD / INTEGRATION using** manual release deployment process

Code (DML)
Schema (DDL)
Promotion: **RELEASE PROCESS**

NON-PROD / INTEGRATION

SERVER Azure SQLMI

DBMS Azure SQLMI

DB

Copy Back / Restore

"*Latest Stable Version of SQL Server*"

MS Azure SQLMI
#Azure Tenant
#Azure Subscription
#Azure VNnets
#Azure Subnets

Instance Layer Configs
Server Collation
Accounts
Agent Jobs
Enabled Features

DBs
Compatibility Levels
DB Collations
Data Variations
Code Variations
ETL Variations

NON-PROD / INTEGRATION:

Data populated by DB restores from PROD, **then** overlaid with DDL / DML **promoted from SANDBOX(es) using GitHub**

Code (DML)
Schema (DDL)
Promotion: **GitHub**

SANDBOX1

SERVER Azure SQLMI

DBMS Azure SQLMI

DB

Copy Back / Restore

"*Latest Stable Version of SQL Server*"

MS Azure SQLMI
#Azure Tenant
#Azure Subscription
#Azure VNets
#Azure Subnets

Instance Layer Configs
Server Collation
Accounts
Agent Jobs
Enabled Features

DBs
Compatibility Levels
DB Collations
Data Variations
Code Variations
ETL Variations

SANDBOXES:

Data populated by DB restores from PROD, **and** overlaid with DDL / DML **created during development**

Code (DML)
Schema (DDL)
Promotion: **GitHub**

SANDBOX2

SERVER Azure SQLMI

DBMS Azure SQLMI

DB

Copy Back / Restore

"*Latest Stable Version of SQL Server*"

MS Azure SQLMI
#Azure Tenant
#Azure Subscription
#Azure VNets
#Azure Subnets

Instance Layer Configs
Server Collation
Accounts
Agent Jobs
Enabled Features

DBs
Compatibility Levels
DB Collations
Data Variations
Code Variations
ETL Variations

Sensitive Data Obfuscation Process (*Custom*)