Edifice Data Extraction process:

Steps:

1. Execute 01\_sftp\_extract.sh bash script using CRON.
2. Upload Raw files as is onto ADLS folder on a daily basis.
   1. /data/raw/edifice/current/year\_day/
3. Split data per account type and header level in each file and create a new file for each account type.
   1. /data/raw/edifice/split\_files/year\_day/
4. Pull retailer name from each file and create a folder under parent directory.
   1. /data/raw/edifice/retailer\_files/retailer/year\_day/
5. Add year\_day, account name, flag and customer number (optional) in each account file.
   1. /data/raw/edifice/staging/retailer/year\_day/
6. Compare /data/raw/edifice/staging/retailer/year\_day /\* with /data/raw/edifice/processed/retailer/year\_day/\* and update files if required.
   1. Compare partition to partition for better performance.
7. Create Hive external partitioned table on top of /data/raw/edifice/processed/retailer/year\_day/\*
8. Create Hive ORC internal table from above hive external table.

Execution Steps:

spark-submit --class com.yeti.dwh.edifice.edificeLoader \

--master yarn \

--deploy-mode cluster \

edw\_2.11-1.1.6.jar \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/input \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/output \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/staging \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/archive

spark-submit --class com.yeti.dwh.edifice.HDFSUtil \

--master yarn \

--deploy-mode cluster \

edw\_2.11-1.1.6.jar \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/input \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/output \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/staging \

adl://yetiadls.azuredatalakestore.net/clusters/data/raw/edifice/archive