Analytics Data Storage and Retrieval Solution

**Overview:** A dedicated space with joint read/write access, with functionality to support structured data, spatial objects, and unstructured text, is needed for creating the most robust environment for the analysis, modeling, and sharing of data.

# Basic Requirements

A query-able version of the data should live with the production EDW environment and/or have the ability to join with databases in the EDW in a quick and efficient manner.

Any solution should support streaming data, both reading and writing.

All analysts and data scientists should have the ability to write to the data storage in either a one-time writing or a scheduled fashioned.

Use cases:

* One time writing
  + Sharing/defining cohorts, new features
  + Added exterior data sources
    - Structured data (e.g. csvs)
    - Spatial objects (e.g. GEOJSON)
    - Unstructured data (e.g. JSON)
* Scheduled writing
  + Saving/updating model results, cohorts, features
  + Geocoding new addresses
  + Free text note, document, social media, etc. processing

In the event that all the data originates and can be manipulated directly in SQL, these processes can be defined either through the use of SAM Designer[[1]](#footnote-1) or a SQL view[[2]](#footnote-2).

More often, other statistical programs (such as R and python) may be required to create complex features, score statistical models, etc., and thus writing to the data storage would be performed through an ODBC connection within these programs. The appropriate process and necessary credentials for writing to the data storage through these connections will need to be identified.

New table, feature, etc. create should be lightweight, and facilitate rapid development.

Version/source control practices should be adopted by the analyst, aligning with current EA version control practices.

# Hypothetical Minimal Solution

One Linux based server with 6 [Docker](https://www.docker.com/) containers (3 TEST, 3 PROD; one each for structured, spatial, and unstructured data)

* Structured: [MYSQL](https://www.mysql.com/)
* Spatial: [POSTGIS](https://postgis.net/)
* Unstructured: [Elasticsearch](https://www.elastic.co/)

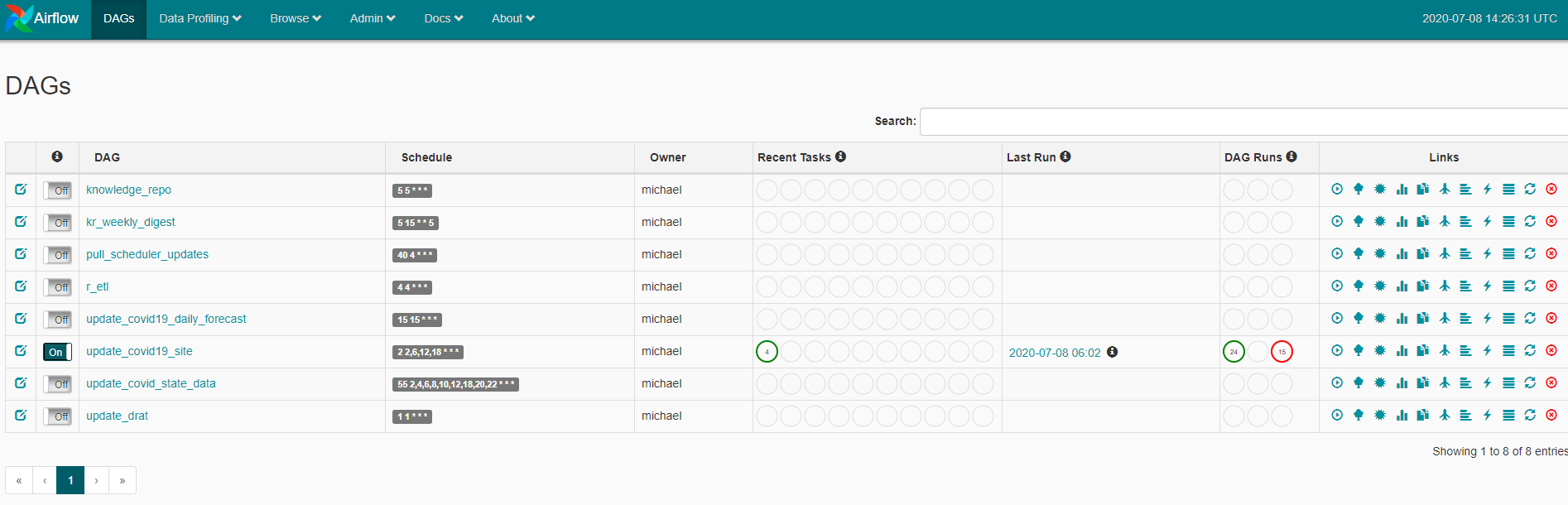
All database read/write would require users to store code in TFS for version controlling

Server-side managed scheduled jobs through [Apache Airflow](https://airflow.apache.org/); data analysts and scientists would create new jobs through python scripts to schedule their process, including moving from a TEST to PROD environment.

Movement from TEST to PROD would require TFS peer approval.

Access would be defined through AD trusted connections

*Airflow Dashboard Example*



1. Process needs to be defined for scheduling databases in TEST to be moved to PROD, preferably at custom time intervals. [↑](#footnote-ref-1)
2. Is this an option? [↑](#footnote-ref-2)