Model Scheduling Development with ADF

This page describes what a model scheduling developer needs to do in ADF for a certain model.

Tasks of a model scheduling

- prerequisites checks, e.g., datasets validation
- execute a scoring notebook
    - databricks linked service, point to the qa-ops-codl-dbricks databricks workspace (when deploy to prd, the release pipeline will replace it with prd-ops-codl-dbricks), and a cluster of either
        - if need additional setups like libraries, an existing cluster that the model scoring requires (to do with the MLOps release pipeline / ADF release pipeline (but duplication of cluster configs)?), with naming convention or sharing variables
        - job cluster with additional configs, manual input or via config file, to install libs via init script. Cannot control the size of new job clusters
        - a new job cluster if default runtime fulfill the needs
    - location of the scoring notebook
    - parameters (model specific) passing to the notebook
- error alert pipeline (can be a parameterized common pipeline, or model specific ones)
    - error notify logic, e.g., http call a logic app that will send out emails to responsibles
- trigger of the execution
- related linked services to the above, e.g., databricks linked services (qa/prd switch handled by DevOps pipeline developer), dataset linked services, keyvault
- what else?

Process of developing model scheduling

1. Create a new branch (after MLOps pipeline released to QA, where the notebooks and clusters deployed in QA)
2. Develop the things required in **Tasks of a model scheduling.**
    a. If using existing cluster, add the cluster ID in PRD (needs to decide the way of adding this, in ADF as parameter, or config file in repo)
    b. if using new job cluster, add cluster config in repo or manual input in linked service
3. PR
4. Merge (will trigger ADF release pipeline)