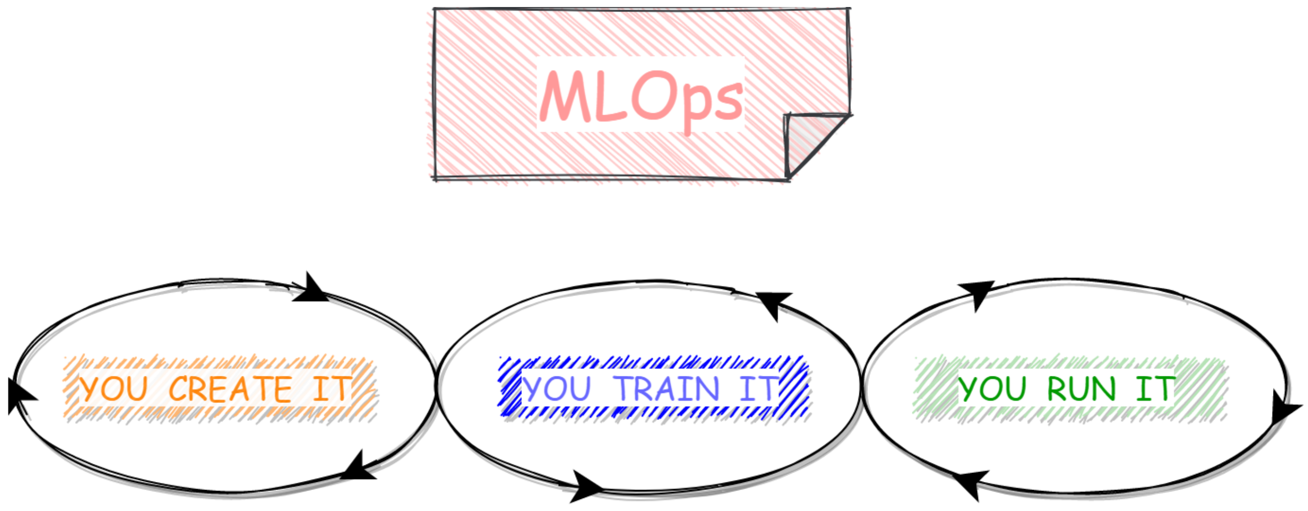
Design for Model Pipelines (MLOps)

Table of Contents



Before you dive into this documentation, we would like to highlight that this is where we want to begin. We will continuously evolve this workflow as we proceed further.

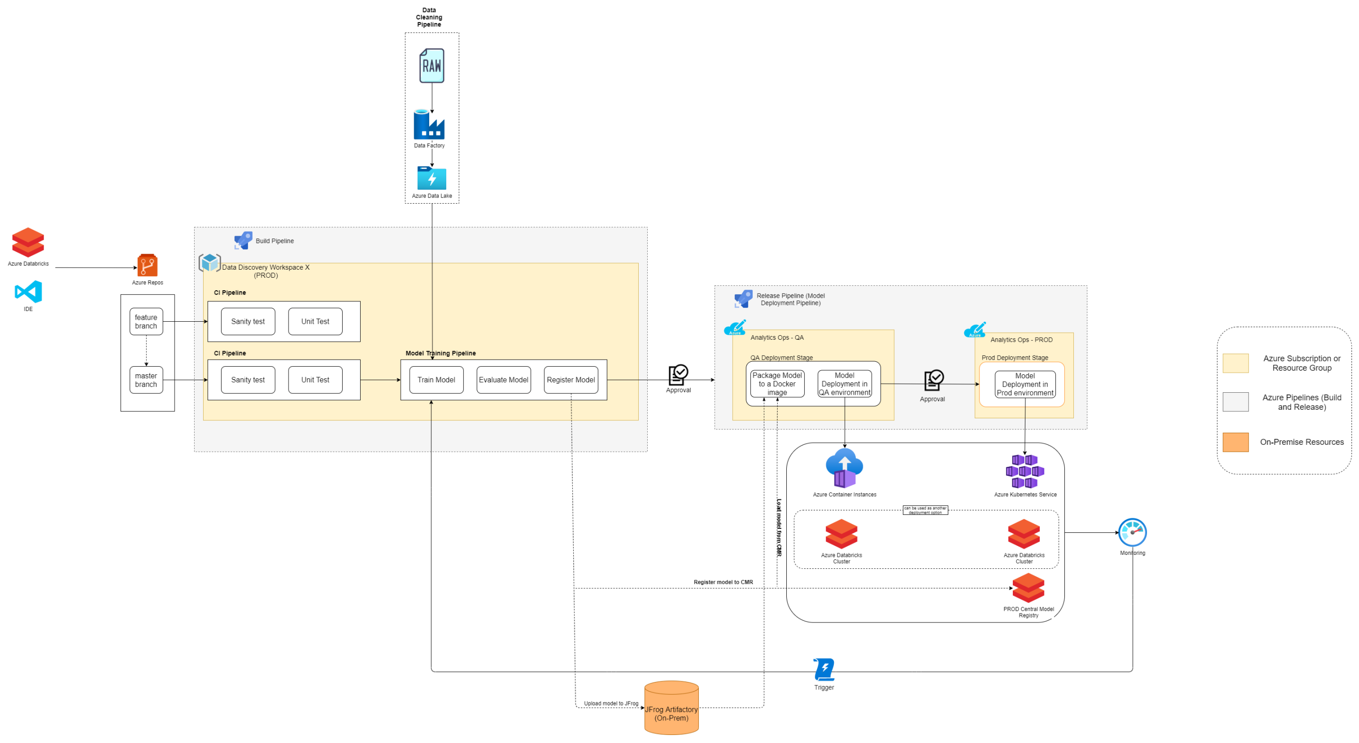
ML Model CICD Workflow

The below diagram shows an overview of the CI/CD workflow of ML models.

NOTE: The below chart is divided into 3 different phases:

1. High-Level Workflow
2. Batch Model Workflow
3. REST Model Workflow

Please browse through different pages created in the below diagram. If you want to go through more details then please check the links mentioned in the **"R eferences"** section later on this page.



The design contains different pipelines:

1. **CI Pipeline -** This pipeline performs sanity tests & unit tests. It is triggered:
   a. When code has been pushed to any branch
2. **Model Training Pipeline -** Model training pipeline trains, evaluates and registers a model (in MLFlow) and uploads it to JFrog Artifactory. This pipeline is triggered:
   a. After successful execution of CI Pipeline on only the master branch
   b. When any new data arrives
   c. When there is any data drift detected while model monitoring
3. **Release Pipeline (Model Deployment Pipeline) -** This pipeline is divided into 2 stages and is responsible for deploying the model in QA and PROD environment. This pipeline retrieves the model from the mlflow model registry, sets up a job in databricks with batch execution. It then deploys the model in QA environment. After a successful deployment of model in QA environment an approval request is sent. After the approval is received for Prod execution, the deployment is done in PROD environment. This pipeline is triggered:
   a. After Model Training Pipeline is executed successfully.
4. **Data Cleansing Pipeline -** This pipeline is responsible for data cleansing. It is triggered:
   a. When the data arrives in the landing zone

Please note that the workflow within the Data Cleansing Pipeline needs more discussion about how the data will be versioned. The idea behind this is to show that the training will be triggered as new data arrives.

MLOps Workflow

Development

1. A "Data Discovery" work environment is provided to a team of data scientists who are working on a usecase. This workspace will have all the services needed to develop the Analytical Model.
2. Data Scientists are free to use an IDE or Azure Databricks workspace for the development work. They will create a feature branch from the master branch.
3. Once the development is done (meaning that they have ensured that the model can be trained and evaluated on the feature branch after implementation), they will raise a PR (to avoid direct push to the master branch). The CI Pipeline for that feature branch can be run and the changes will be validated.
4. After successful execution of CI Pipeline and after receiving approval on the PR the code is ready to be merged to the master branch

Model Training

1. After the code is merged to the master branch, **CI Pipeline** is executed again. After the code validation, the **Model Training pipeline** is executed. The execution happens in the Data Discovery Workspace assigned to the DS.
2. This pipeline is responsible for the Training, Evaluation, and Registration of the model in MLFlow central model registry.

3. The model binaries are also uploaded to on-prem JFrog Artifactory

We will be using **self-hosted agents** for the Model Training Pipeline because as per the documentation - *Builds can run forever on self-hosted agents (private agents). For Microsoft-hosted agents for a public project, builds can run for six hours. For private projects, the limit is 30 minutes.*

Model Deployment

1. After the model is successfully trained and registered, approval requests to run in QA environment are sent. After the request is approved, the next stage of the Release Pipeline is triggered.
2. This pipeline is divided into 2 different jobs stages:
   a. QA Deployment Stage
      i. Here the model is readied for execution as a Databricks batch job, a cluster workspace is prepared with the needed libraries and model assets
      ii. Model is deployed in **Ops - Analytics Ops** and the work area is **QA** environment
   b. Prod Deployment Stage
      i. After successful execution of QA Deployment Stage, another approval request is sent for PROD Deployment
      ii. The deployment is done in **PROD** environment in **Ops - "Analytics Ops"** work area after the request is approved
      iii. The deployment is done in Azure Kubernetes Service (optionally, the model can be deployed on Azure Databricks Cluster as well instead of using Azure Kubernetes Service)

Model Re-Training

1. Model is continuously re-trained as soon as new data is available in the landing zone
2. **Data Cleansing Pipeline** is responsible for cleaning the data and triggering the **Model Training Pipeline**

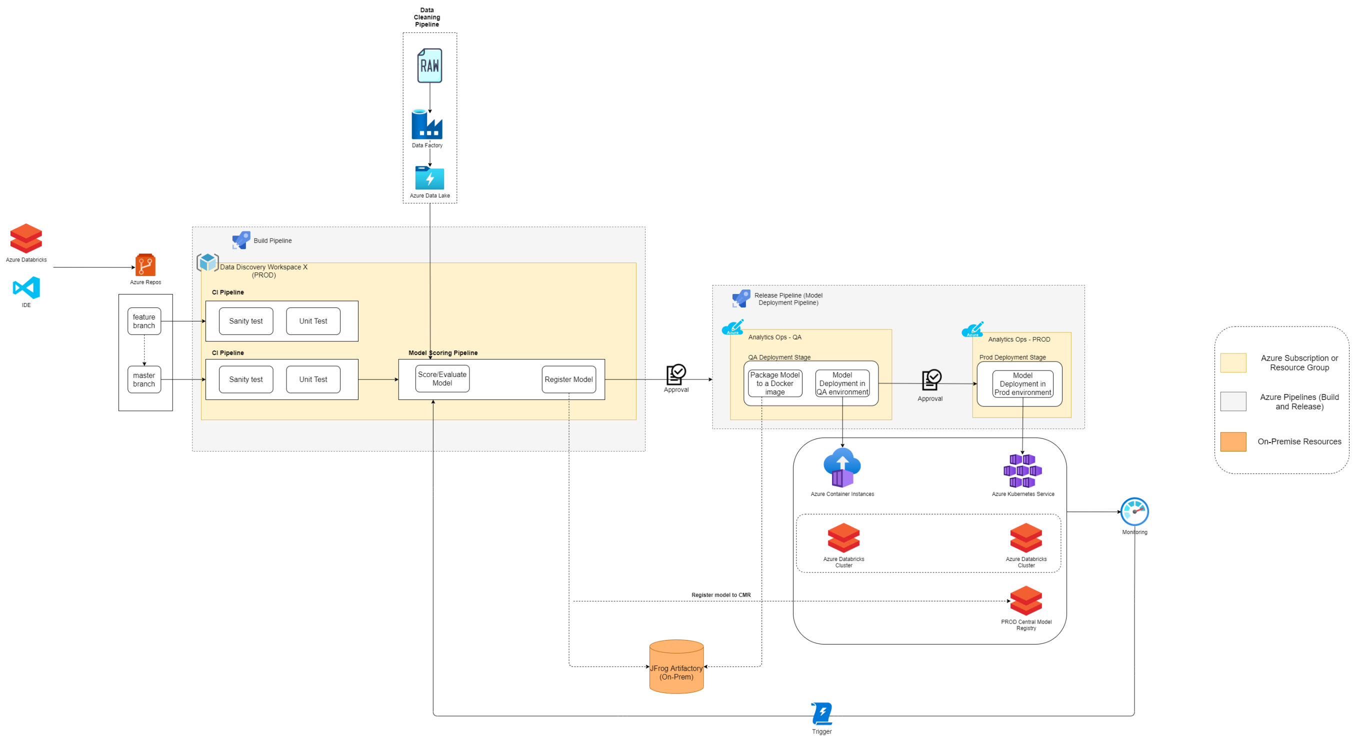
Model Monitoring

1. As the model is deployed, the system starts monitoring the model. If it encounters any drifts then it triggers **Model Training Pipeline**
2. A newer version of the model is trained. The scores are compared and the best model is deployed in the production

Note: *For now the Data Discovery Workspace will not be decommissioned at any point in time.*

Rule-Based Model CICD Workflow

The below diagram shows a very high-level overview of CI/CD workflow for rule-based models.



The workflow for the rule-based models will be quite similar to the ML models. The differences are listed below:

1. Rule-based models will not be having training step. Instead, there will be a scoring/evaluation step in the pipeline. Registering the model will be optional and will depend on the model usecase.
2. Instead of Model Training Pipeline, there will be a Model Scoring Pipeline that will do model scoring/evaluation.
3. The model will be monitored and Model Scoring Pipeline will be triggered in case any threshold is breached.

Measuring the success criteria of the MLOps Workflow

The whole solution should be divided into different levels. This will give us a clear understanding of where we are while implementing the above workflow as the requirements of a fully mature environment can be a bit overwhelming.

Level	Description
Level 1	DevOps but no MLOps
Level 2	Automated Training
Level 3	Automated Model Deployment
Level 4	Automated Training after new data arrives
Level 5	Automated Operations (full MLOps including monitoring)

Centralized ML Flow Registry

There was a meeting involving the Databricks team where we gave a walk-through of the design (above). It all seemed fine to them. When it came to de-commissioning of the DD Workspace, they mentioned that it would be a good idea to register the model in a centralized Databricks workspace. This workspace should not have any computes in it and should only be used for model registry purposes.

On the question of **where should this workspace exist?** - The decision was to spin this up in **PROD Ops** subscription.

You can read more about the centralized ML Flow workspace here.

References

MLOps Two-Steps Implementation - Basic to Enhanced

AzureML Explorations

Base image configuration

FAQs

Q. How many types of models are we looking at?

1. *Machine Learning Models*
2. *Rule-based Models*

The models can further have categories of: (a) Batch (b) Real-time

Q. Are there any references or examples that will be provided to us which we can refer to during the model development?

Yes, there will be few demo models that will be provided later which could be used as a reference

Q. What are the pre-requisites before I start model development on Azure? Is there any checklist?

There will be a checklist in place so that there are no blockers

Q. Can I run multiple versions of models?

Yes, this is something that will be focused on in this solution. So far, it seems like if this problem is resolved then the applications consuming the model will also have to make changes from their end as well.

Q. How long will the Data Discovery Workspace persist (considering we can only create 7 workspaces at the moment)? When should this be decommissioned? What will happen to the registered models, datasets and other model-related data?

After having several discussions, it has been decided that the Data Discovery Workspace will not be decommissioned for now.

Q: Which image should be used for packaging the models?

see Base image configuration

Q. Are the model pipelines generic?

No, the pipelines are not generic as of today.

Q. Why aren't the pipelines generic?

If we create generic pipelines then everyone will be able to see model logs in Azure pipelines and this is something that we would like to avoid. And as of today, there is no mechanism in Azure Pipelines to segregate this. Having generic pipelines will make debugging of jobs in Azure Pipeline a bit difficult.

Also, there is no One-Size-Fits-All. Considering the services that are exposed to the DS, there could be a lot of different design patterns that they could use. Creating a generic pipeline will create restrictions.

The decision is - have different pipelines for each model, discover different design patterns and then create generic pipelines.

Q. Could the pipeline process be eased somehow?

Yes, there is a task in the backlog to create a CLI which initializes the model repository and creates pipeline files automatically based on the values user has selected.

Q. What will be the cluster configuration of ACI and AKS?

TBD

Q. Will the models be deployed using Ab Initio? How will that fit in this workflow?

TBD

Q. What will be the workflow of the Data Cleansing Pipeline?

TBD

Q. What if there is a model already trained somewhere else (BYOM)?

TBD