Databricks mlflow Object Relationships

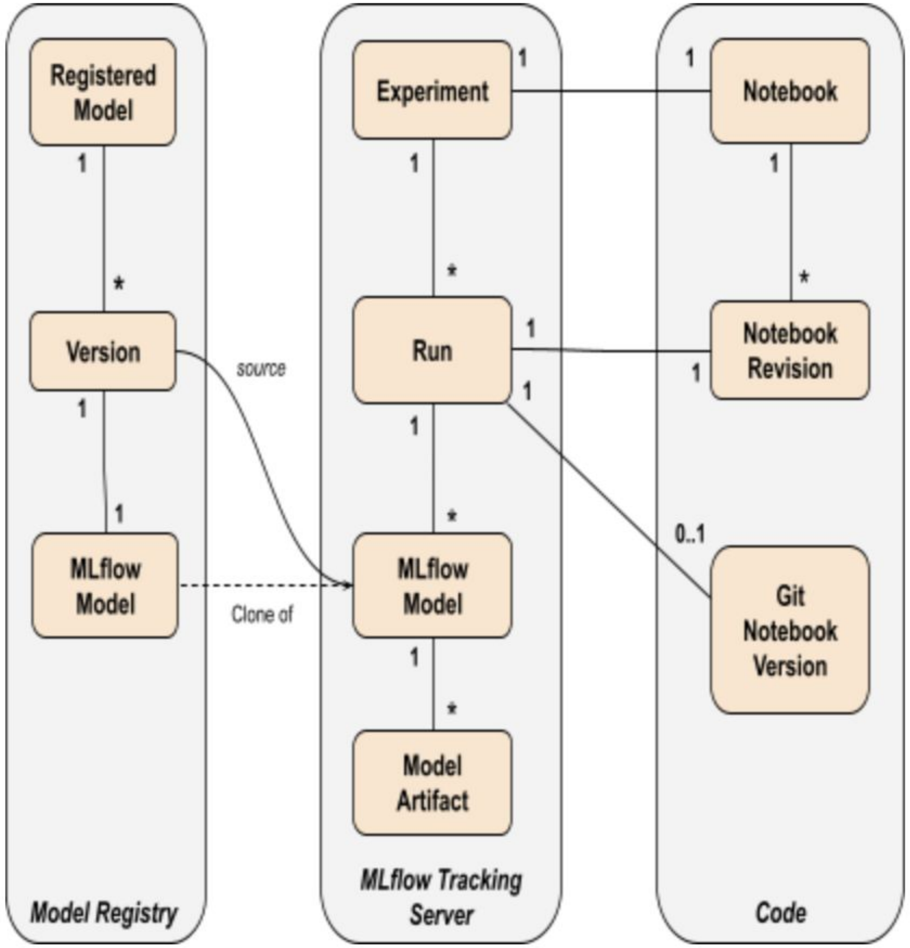
Andre Mesarovic
Sr. Resident Solutions Architect
2 January 2022

databricks

Overview

- Describe the relationships between Databricks MLflow objects:
  - Runs
  - Experiments
    - Notebook experiments
    - Workspace experiments
  - Models
    - Registered models and model versions
    - MLflow models
    - Model artifacts
  - Notebooks

MLflow Object Relationships



databricks

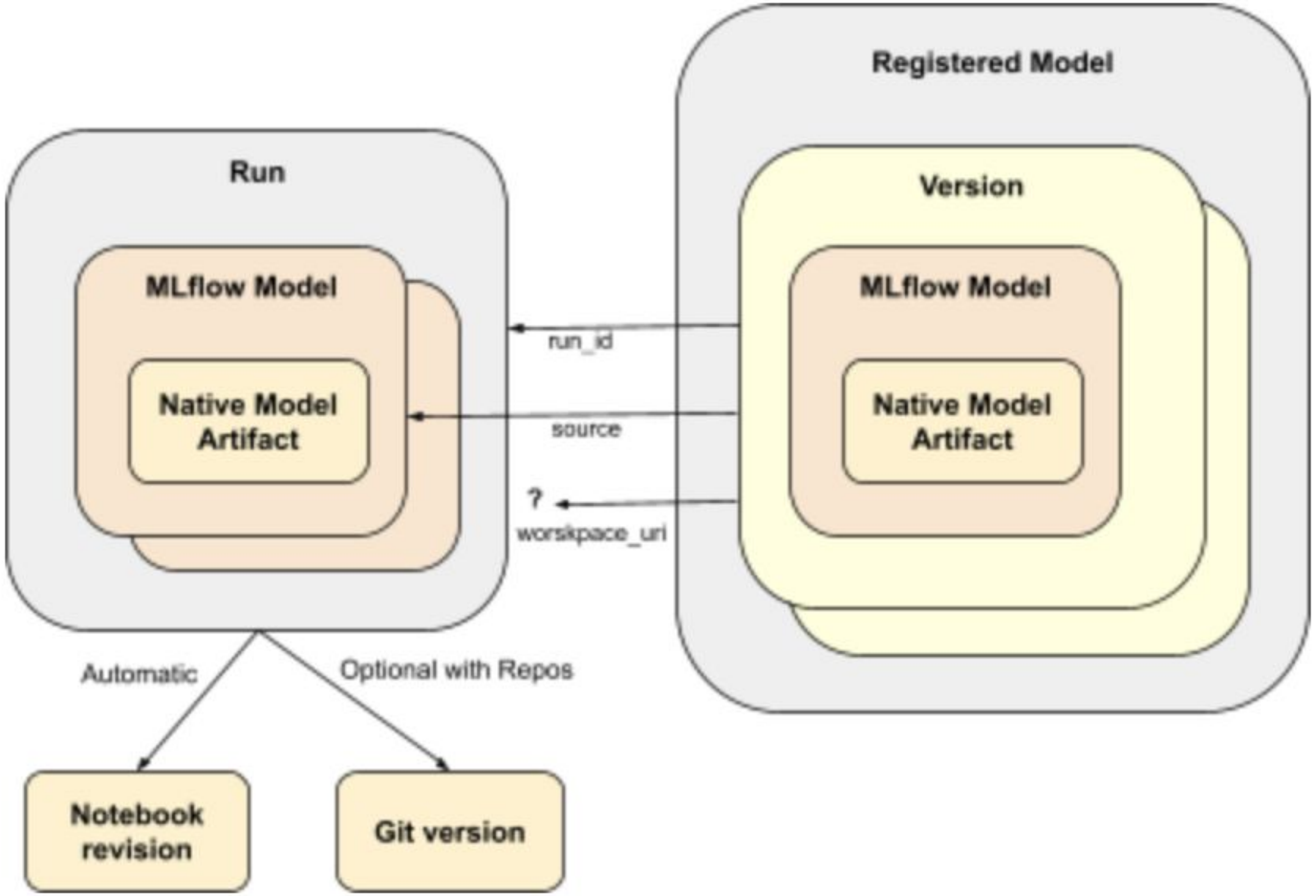
Diagram Legend

- Diagram uses the UML modeling language.
  - *: indicates a many relationship
  - 1: indicates a required one relationship.
  - 0..1: indicates an optional one relationship.
- This is a logical diagram. Not all nuances are captured for simplification.
- The diagram represents a notebook experiment.
- A workspace experiment is not represented in the diagram.

Model Terminology

- Model is an overloaded term with three meanings:
    - **Native model artifact** - this is at the lowest level and is simply the native flavor's serialized format. For sklearn it's a pickle file, for Keras it's a directory with TensorFlow's native SaveModel format files.
    - **MLflow model** - a wrapper around the native model artifact with metadata in the MLmodel file and environment information in conda.yaml and requirements.txt files.
    - **Registered model** - a bucket for model versions. A model version contains one MLflow model that is cached in the model repository. A version has the following links (expressed as tags):
        - run_id - points to the run that generated the version's model.
        - source - points to the path of MLflow model in the run that corresponds to the version's model.
        - workspace_uri - currently missing. Needed if using shared model registry. ML-19472.

Model Relationships



Registered models

- A registered model can have several model versions.
- The *production* and *staging* stage have one "latest" version.
- A version has one MLflow model which is linked to one run.
- Registered model versions are cached in the model registry.
- This is a clone of the run's MLflow model that the version points to.
- If source run is in a different workspace we have a lineage reachability problem. See ML-19472.

Experiments

- An experiment has one or more runs.
- Two types of experiments:
  - Notebook experiment
    - Relationship of experiment to notebook is one-to-one.
    - Workspace path of the experiment is the same as its notebook.
  - Workspace experiment
    - Relationship of experiment to notebook is one-to-many.
    - Explicitly specify the experiment path with *set_experiment* method.
    - Different notebooks can create runs in the experiment.

Runs

- A run belongs to only one experiment.
- A run is linked to one notebook revision. MLflow notebook tags:
  - mlflow.databricks.notebookRevisionID
  - mlflow.databricks.notebookID
  - mlflow.databricks.notebookPath
- Optionally a run can be linked to a git reference.
  - See discussion on Notebook below for details.
- A run can have one or more MLflow models (flavors) such as Sklearn and ONNX.
- Every run has a Pyfunc flavor.

Notebooks

- A notebook has many revisions.
- Optionally, a notebook revision can be checked into git with Databricks Repos.
- Need to capture git reference analogous to the MLflow open source tags:
  - mlflow.source.git.commit
  - mlflow.source.git.repoURL
  - mlflow.gitRepoURL
- See ML-19473
- Two sources of truth for a notebook snapshot that can be confusing:
  - Databricks notebook revision
  - Git version

Thank You!

Happy MLflow journey!