Last Updated: 4/18/2025

Operationalizing ML Across Clouds with Delta Sharing (Demo)

Share, Deploy, and Serve Machine Learning Models Across AWS and Azure Workspaces Using Unity Catalog and Delta Sharing

Authors: Alexander Booth Smriti Sridhar Dan Pechi

Introduction

This demo showcases how to share and operationalize machine learning models across multiple Databricks workspaces using Unity Catalog and Delta Sharing, including a cross-cloud scenario between AWS and Azure.

In this setup:

- **Workspace A (Dev Workspace on AWS):**
  Train a machine learning model using AutoML, register it in Unity Catalog, and share it via Delta Sharing.

- **Workspace B (Prod Workspace on Azure):**
  Accept the shared model, load it for ad hoc inference, deploy it to a real-time serving endpoint, and perform API-driven predictions.

This cross-cloud workflow highlights the use of native Databricks capabilities for seamless multicloud model sharing, centralized governance, and production deployment without requiring manual artifact handling. However, any two workspaces can be used interchangeably.

The demo is organized across two Databricks Notebooks:

- **Notebook 1 (Workspace A):** Train, Register, and Share the Model
- **Notebook 2 (Workspace B):** Accept, Infer, and Serve the Model

Purpose

This demo demonstrates how organizations can streamline machine learning operations across environments by combining Unity Catalog, Delta Sharing, and Databricks Model Serving.

Key objectives include:

- Show how a model trained in one workspace can be securely shared and consumed by another workspace.
- Enable immediate ad-hoc inference from shared Unity Catalog models.
- Deploy a shared model to real-time serving for production-ready API access.
- Highlight best practices for supporting model experimentation and future A/B testing between shared models.

By following this guide, users will gain practical experience with cross-environment model sharing, centralized governance, scalable deployment, and operationalization using only native Databricks services.

Additionally, this demo highlights how Delta Sharing can facilitate secure model operations across cloud providers, enabling true multicloud machine learning workflows without friction.

How to Run the Demo

This demo is executed across two Databricks workspaces: one for model creation (Dev) and one for model consumption (Prod).

Follow the steps below to complete the end-to-end workflow:

Workspace A (Dev Workspace): Train, Register, and Share the Model

1. **Launch Notebook 1** in Workspace A.
2. **Load and explore** the bike sharing dataset.
3. **Train a model** using Databricks AutoML.
4. **Register the best model** in Unity Catalog.
5. **Share the registered model** using Delta Sharing to make it available to Workspace B.

At the end of this notebook, the model will be registered and shared across workspaces using Unity Catalog and Delta Sharing.

Workspace B (Prod Workspace): Accept, Predict, and Serve the Model

1. **Launch Notebook 2** in Workspace B.
2. **Accept the Delta Share programmatically** by creating a new catalog linked to the shared model.
3. **Load the shared model** directly from Unity Catalog.
4. **Perform ad hoc inference** on new sample data using the loaded model.
5. **Create a real-time serving endpoint** programmatically using the Databricks Model Serving API.
6. **Send inference requests** to the deployed serving endpoint via REST API.
7. **View prediction results** and confirm end-to-end model operationalization.

Prerequisites

Before running the demo, ensure that:

- Both workspaces are **Unity Catalog-enabled**.
- **Delta Sharing** is configured and enabled between the two workspaces.
- **Model Serving** is available and enabled in Workspace B.
- You have appropriate privileges:
    - Workspace A: Ability to create models and delta shares.
    - Workspace B: Ability to create catalogs and serving endpoints.
- Databricks Runtime **15.4 LTS ML** or later is used for both workspaces.

Next Steps

This demo provides a foundation for cross-workspace model sharing and operationalization. The workflow can then be extended to support more advanced use cases, including model experimentation and A/B testing.

Potential next steps include:

- **A/B Testing Across Shared Models**
    - Share multiple versions of a model (e.g., a current production model and a candidate model) via Delta Sharing.
    - Deploy each model to its own serving endpoint or configure a single endpoint to route traffic between versions.
    - Route production inference traffic between the models using predefined splits (e.g., 50/50 or 80/20).
    - Capture predictions and outcomes into Delta tables for analysis.
    - Compare model performance over time to identify improvements or regressions.
- **Model Version Management**
    - Update model aliases such as `@prod` or `@staging` to manage lifecycle transitions.
    - Use Unity Catalog lineage and audit features to track model consumption and downstream impacts.
- **Automation and CI/CD Integration**
    - Integrate model registration, sharing, deployment, and promotion into CI/CD pipelines for fully automated MLOps workflows.
    - Programmatically promote winning models after A/B testing results are finalized.
- **Expand Sharing Across Clouds or Organizations**
    - By using Delta Sharing's open protocol, organizations can share models not only between internal workspaces but also with partners or customers across cloud providers.

By building on this foundation, organizations can achieve fully governed, scalable, and efficient machine learning operations across environments, accelerating experimentation while maintaining control and compliance.

Summary

This demo showcased how Unity Catalog, Delta Sharing, and Databricks Model Serving can be used together to enable secure, scalable, and operationally efficient machine learning workflows across workspaces.

By leveraging native platform capabilities, teams can streamline model sharing, simplify production deployments, and accelerate experimentation with minimal overhead.

The practices demonstrated here form a strong foundation for advanced MLOps strategies, including model monitoring, governance, and A/B testing at enterprise scale.

References

- **Delta Sharing Overview**
  Introduction to Delta Sharing, an open protocol for secure data sharing across organizations and clouds.

- **Unity Catalog Documentation**
  Detailed guide on how Unity Catalog enables centralized governance for all data and AI assets in Databricks.

- **Databricks Model Serving Overview**
  Explanation of Databricks Model Serving and how to deploy real-time APIs for ML models.

- **Manage Models with Unity Catalog**
  How models are registered, governed, and managed through Unity Catalog.

- **Deploy and Query MLflow Models in Unity Catalog**
  Step-by-step guide for registering and serving MLflow models in Unity Catalog.

- **Blog: Introducing Delta Sharing**
  Original Databricks blog announcing Delta Sharing and describing its vision for open data collaboration.

- **Blog: Manage ML Models with Unity Catalog**
  Overview of managing and governing ML models natively with Unity Catalog.