



# Managing the model lifecycle with Databricks ML

MLflow and the model lifecycle



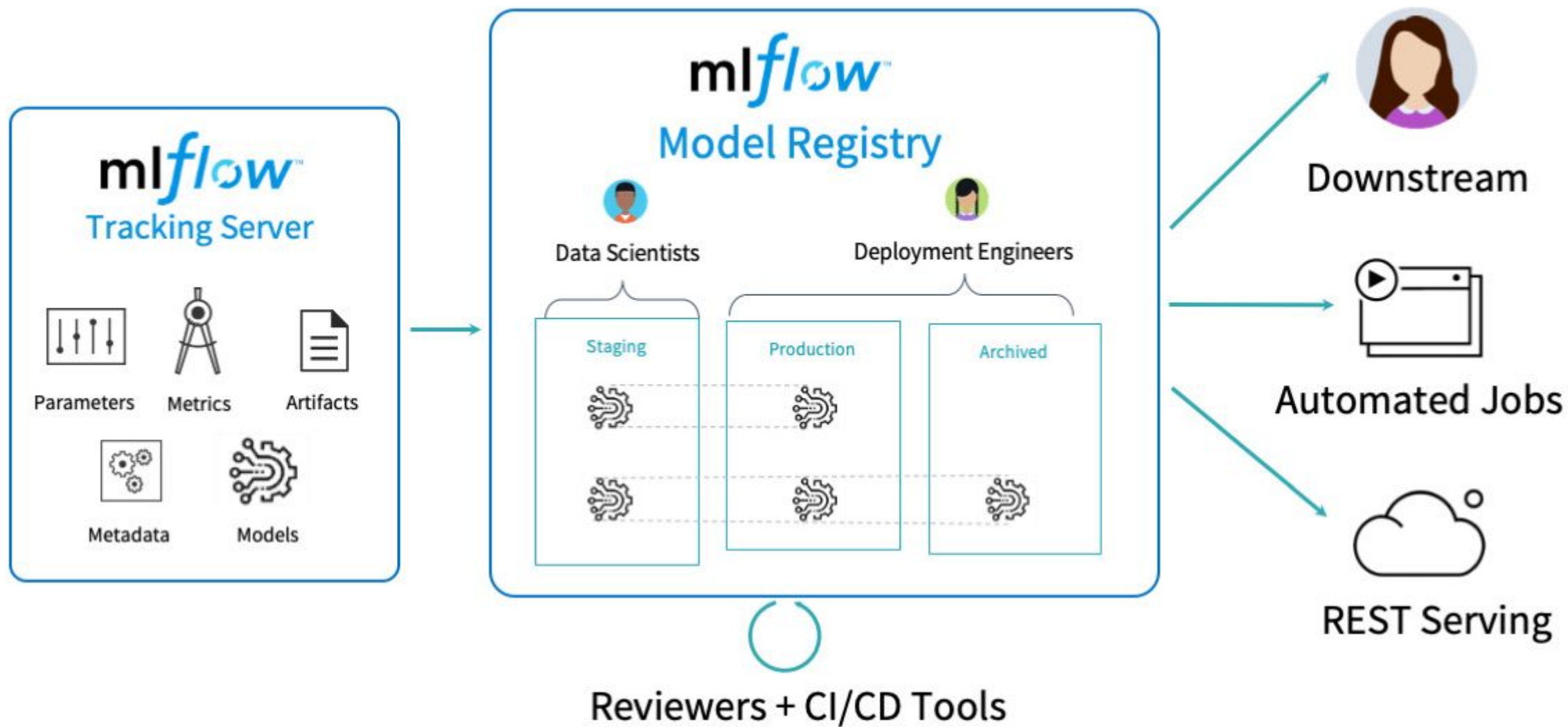
# Learning objectives

- Describe the purpose of MLflow.
- Describe how MLflow supports Machine Learning initiatives.

# What is MLflow?



An open-source platform for  
managing the end-to-end  
machine learning lifecycle.



# MLflow Tracking and Autologging

# MLflow Models

## MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register it to the model registry](#) to version control and deploy as a REST endpoint for [real time serving](#).

### Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
Inputs (56)	
state_AK	double
state_AL	double
state_AR	double
state_AZ	double
state_CA	double
Outputs (1)	
-	Tensor (dtype: float64, shape: ...)

### Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/8ccea8bc64504eb29bc0786b9d230d6e/model'

# Load model as a Spark UDF. Override result_type if the model does not return double value
s.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/8ccea8bc64504eb29bc0786b9d230d6e/model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
loaded_model.predict(pd.DataFrame(data))
```

# MLflow Model Registry

The image displays the Databricks MLflow Model Registry interface. The top navigation bar includes the Databricks logo, a search bar, and a user profile icon. The left sidebar contains a navigation menu with options like Machine Learning, New, Workspace, Repos, Recents, Data, Compute, Workflows, Experiments, Feature Store, Models (highlighted), Partner Connect, and Menu options.

The main content area is titled "Registered Models" and features a "Create Model" button. Below this, there is a list of registered models. The first model is highlighted, showing its name, "Name", and a "Permissions" button.

A detailed view of a specific model is shown, titled "Registered Models > -rf-model\_5ff8f5". This view includes tabs for "Details" and "Serving". The "Details" tab is active, showing the model's name, a "Permissions" button, and a "Use model for inference" button. Below this, there is a "Notify me about" section with a dropdown menu set to "Activity on versions I follow". The "Created Time" is 2022-09-06 09:59:59, and the "Last Modified" time is 2022-09-06 10:04:49. The "Creator" field is empty.

The "Description" section is expanded, showing the model's description. The "Tags" section is also expanded. The "Versions" section is expanded, showing a table of model versions. The table has columns for "Version", "Registered at", "Created by", "Stage", "Pending Requests", and "Description".

Version	Registered at	Created by	Stage	Pending Requests	Description
Version 2	2022-09-06 10:02:44		Production	-	This model version is a rand...
Version 1	2022-09-06 09:59:59		Production	-	This model version was built ...

The bottom of the interface shows a pagination bar with a "1" button and navigation arrows.



# MLflow Model Serving

The screenshot shows the Databricks MLflow Model Serving interface. The top navigation bar includes the Databricks logo, a search bar, and a user profile icon. The left sidebar contains navigation links for Machine Learning, New, Workspace, Repos, and Recents. The main content area displays the 'Registered Models' page for a model named '-rf-model\_5ff8f5'. The 'Details' tab is selected, showing the model's name, a 'Notify me about' dropdown set to 'Activity on versions I follow', and the creation and modification timestamps. A 'Permissions' button and a 'Use model for inference' button are visible in the top right corner.

The screenshot shows the Databricks MLflow Model Serving interface, specifically the 'Serving' tab for a model named '-model\_5ff8f5'. The 'Details' tab is also visible. The 'Serving' tab contains a message: 'Enable real-time model serving behind a REST API interface. This will launch a single-node cluster that will host all active versions of this model. [Learn more.](#)' Below this message is a blue button labeled 'Enable Serving'.



# Managing the model lifecycle with Databricks ML

Use Model Registry to manage the model lifecycle

>\_



# Learning objectives

- Use Model Registry to manage the model lifecycle.



# Managing the model lifecycle with Databricks ML

Deploy a model for batch inference

>\_

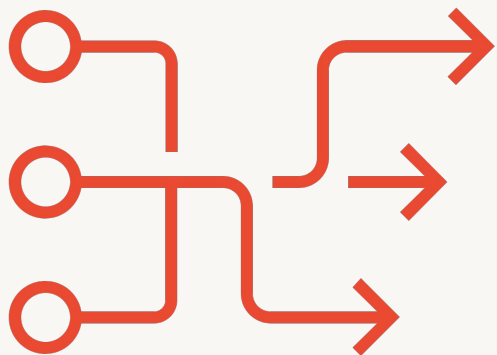


# Learning objectives

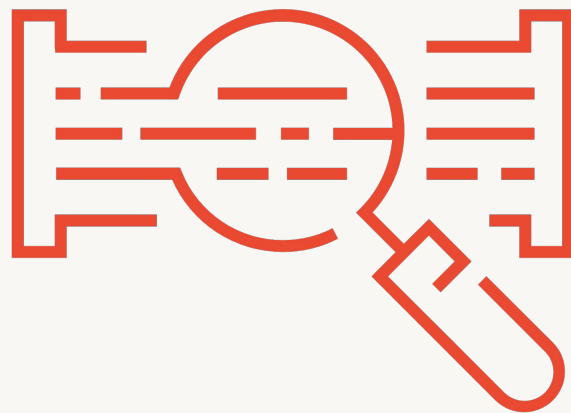
- Use a registered model and feature table to perform batch inference.

# Deploying Models to Production

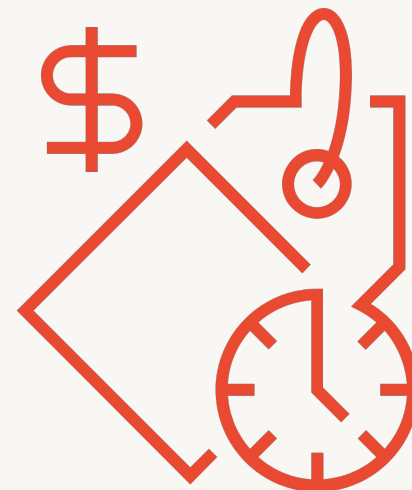
## Batch



## Streaming



## Real-time serving





# Managing the model lifecycle with Databricks ML

Automatic Model Retraining

>\_



# Learning objectives

- Schedule a model refresh using Databricks Workflows and AutoML.