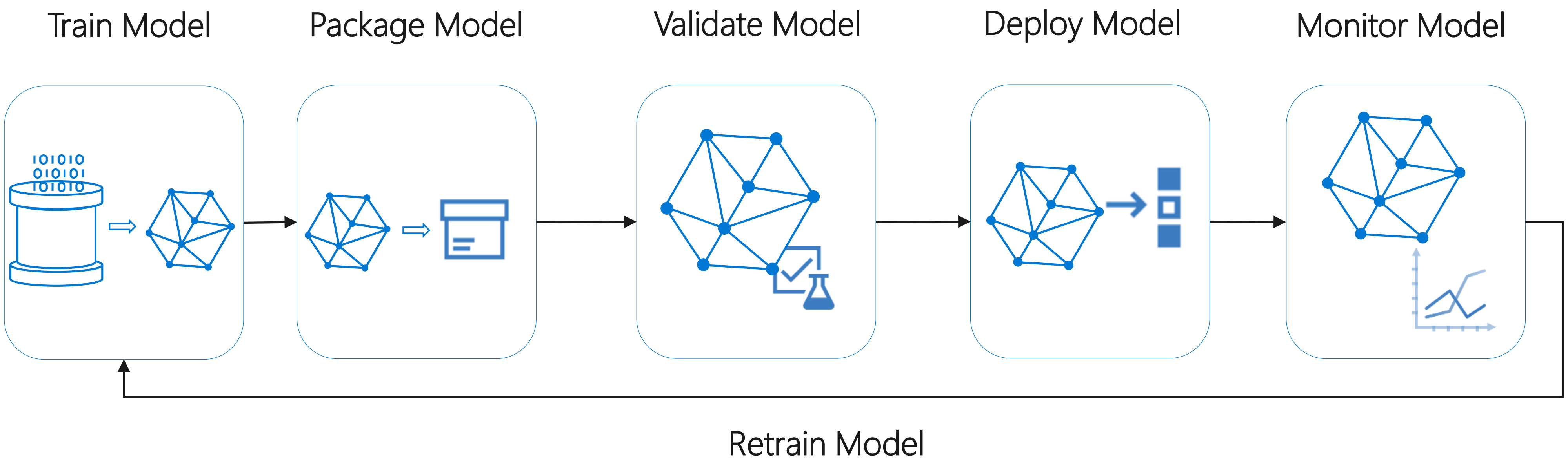
Microsoft

# DevOps for ML
# MLOps

# What does the Machine Learning Lifecycle look like?

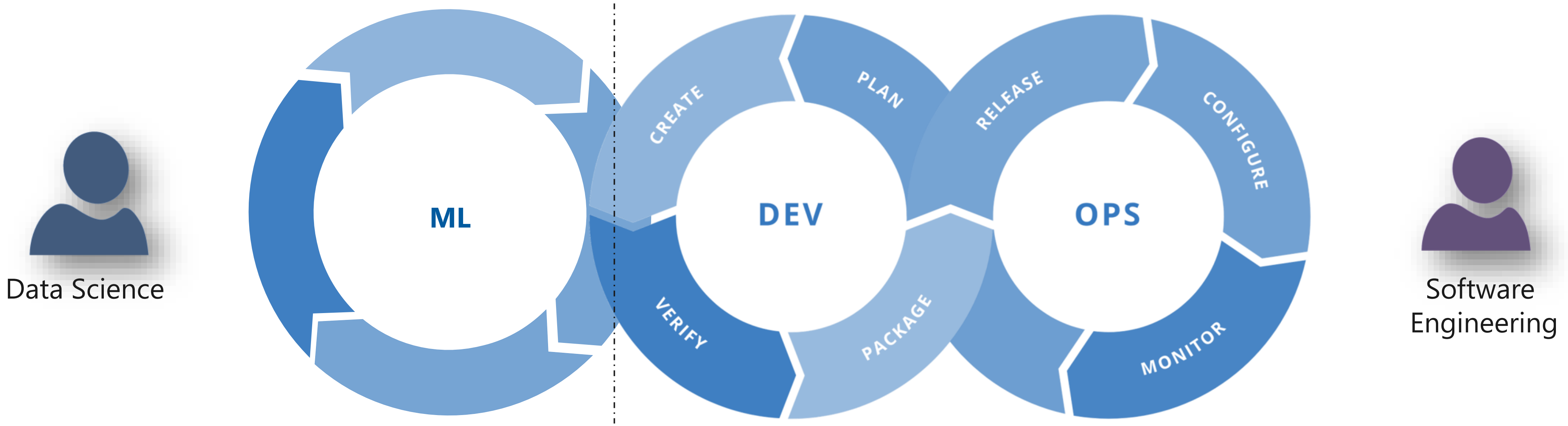
- **Develop & train model** that solves a real business problem
- **Package model** so you can use it somewhere else
- **Validate model behavior –** functionally, in terms of responsiveness, in terms of regulatory compliance
- **Deploy model –** use the model to make predictions
- **Monitor model** behavior & business value, know **when to replace / deprecate a stale model**



# What is MLOps?

# MLOps = ML + DEV + OPS

Help bring ML workflows to production



## Experiment

Data Acquisition
Business Understanding
Initial Modeling

## Develop

Modeling + Testing
Continuous Integration
Continuous Deployment

## Operate

Continuous Delivery
Data Feedback Loop
System + Model Monitoring

# How is MLOps different from DevOps?

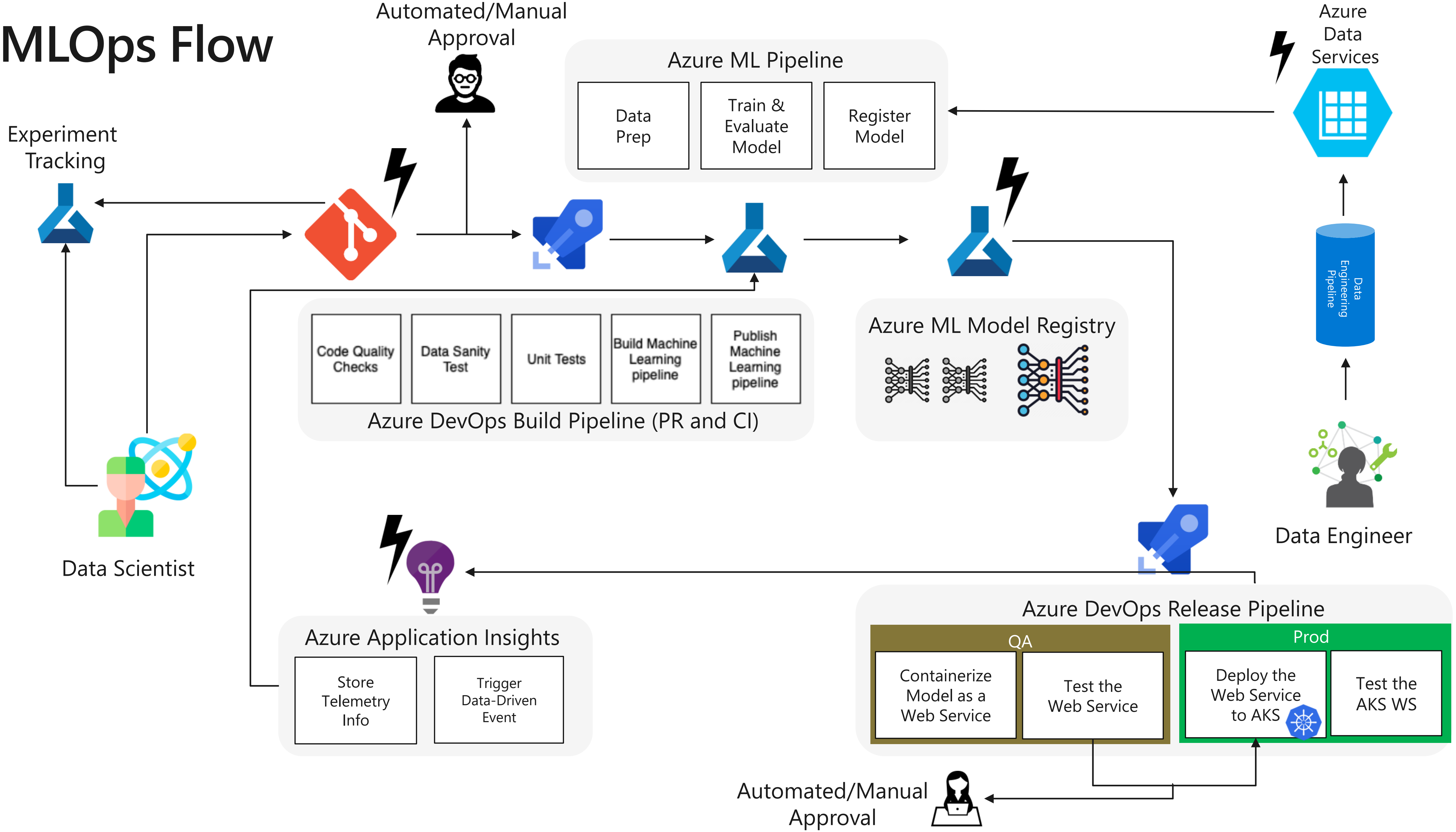
**Data/model versioning != code versioning** - how to version data sets as the schema and origin data change

**Digital audit trail (lineage) requirements** change when dealing with code + data

**Model reuse is different than software reuse**, as models must be tuned based on input data / scenario. To reuse a model you may need to fine-tune / transfer learn on it (meaning you need the training pipeline)

**Model performance tends to decay over time** & you need the ability to retrain them on demand to ensure they remain useful in a  production context.

# MLOps Flow



Automated/Manual Approval

Azure ML Pipeline

Data Prep

Train & Evaluate Model

Register Model

Azure Data Services

Experiment Tracking

Data Engineering Pipeline

Code Quality Checks

Data Sanity Test

Unit Tests

Build Machine Learning pipeline

Publish Machine Learning pipeline

Azure DevOps Build Pipeline (PR and CI)

Azure ML Model Registry

Data Scientist

Data Engineer

Azure Application Insights

Store Telemetry Info

Trigger Data-Driven Event

Azure DevOps Release Pipeline

QA

Containerize Model as a Web Service

Test the Web Service

Prod

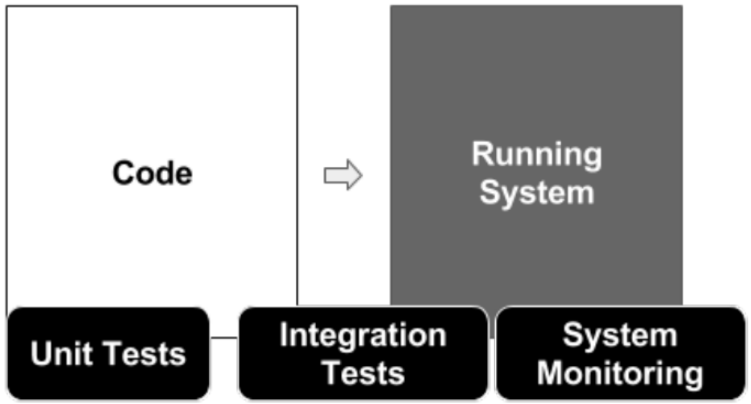
Deploy the Web Service to AKS

Test the AKS WS

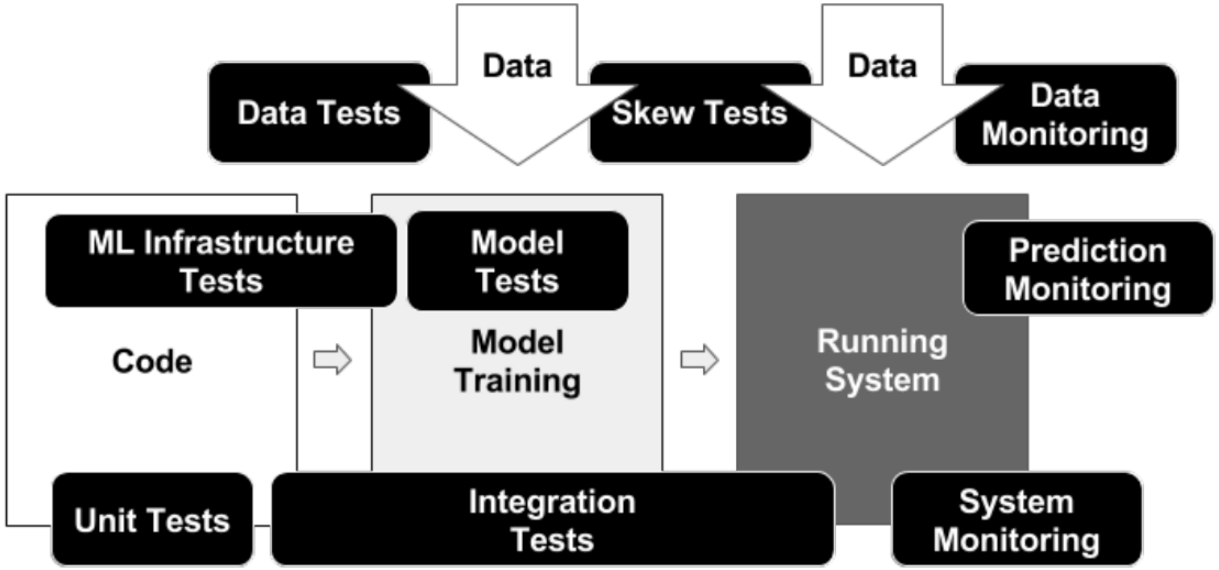
Automated/Manual Approval

# Why is MLOps hard?

# Traditional vs. ML infused systems



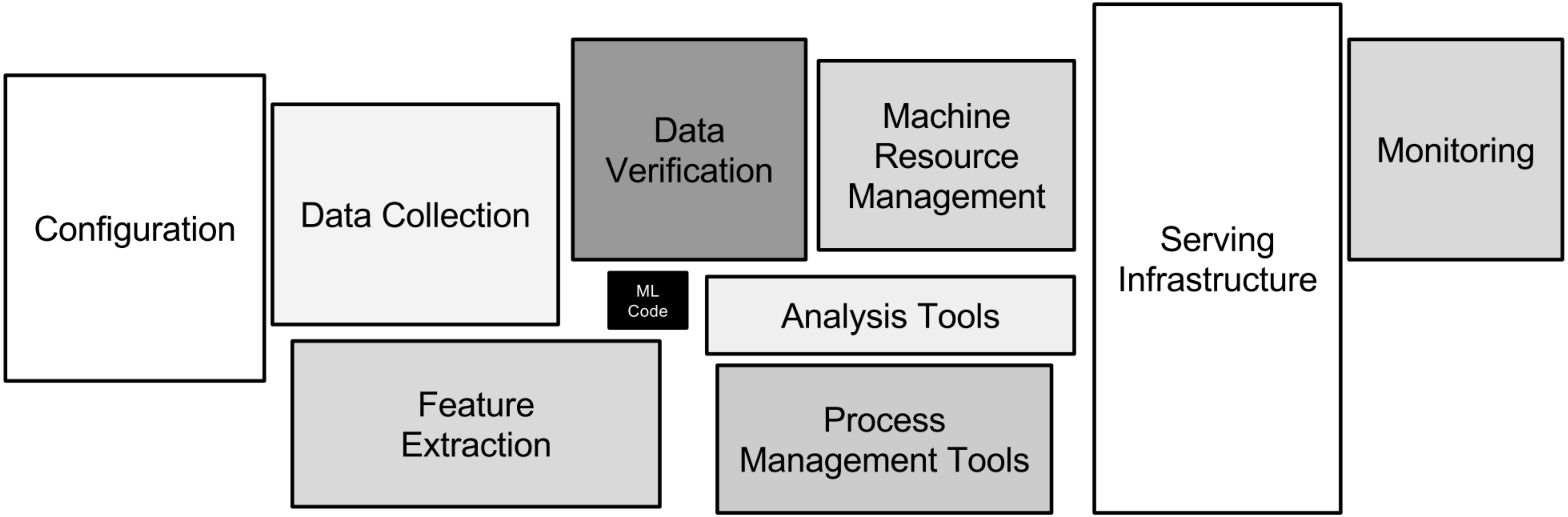
Traditional System Testing and Monitoring



ML-Based System Testing and Monitoring

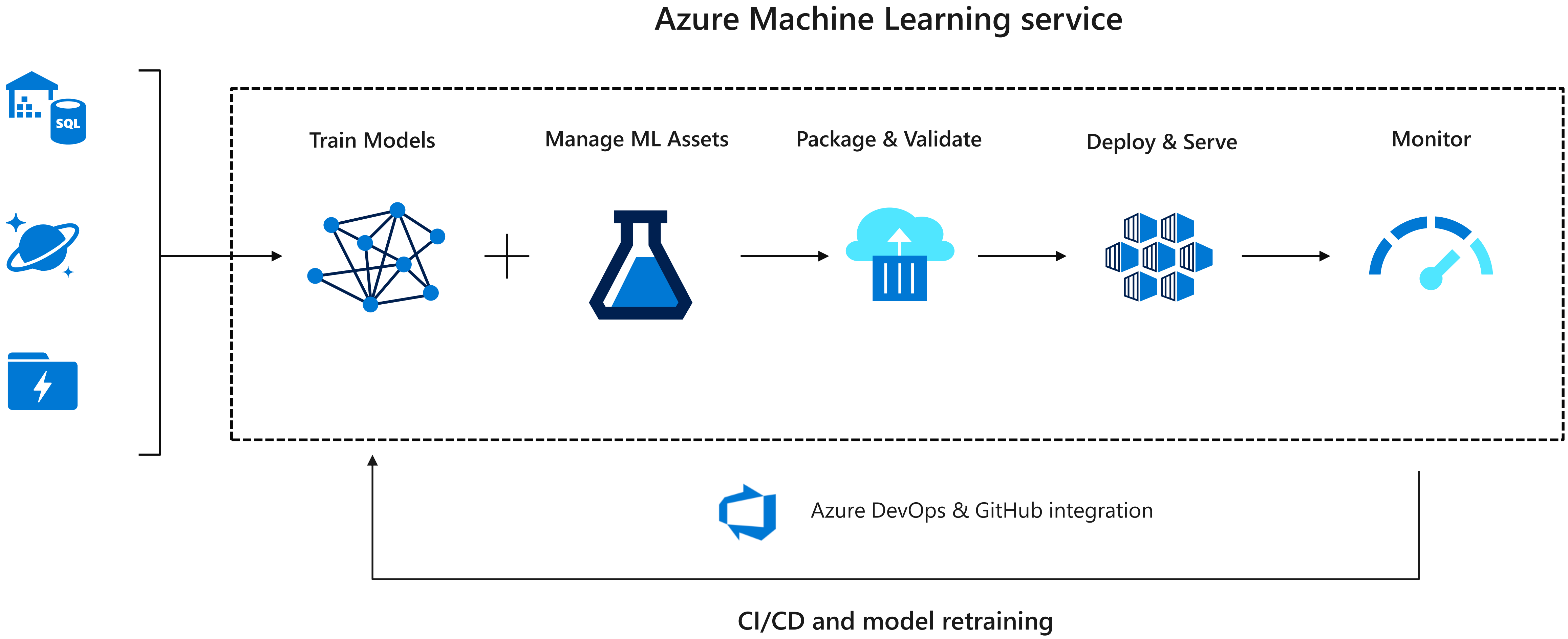
ML introduces two new assets into the software development lifecycle – **data** and **models**.

# More assets & process to manage



*Sculley, D.; Holt, Gary; Golovin, Daniel; Davydov, Eugene; Phillips, Todd; Ebner, Dietmar; Chaudhary, Vinay; Young, Michael; Crespo, Jean-Francois; Dennison, Dan (7 December 2015). "Hidden Technical Debt in Machine Learning Systems"*

# Bring your ML workflows to production with MLOps


Azure Machine Learning service
SQL
Train Models
Manage ML Assets
Package & Validate
Deploy & Serve
Monitor
Azure DevOps & GitHub integration
CI/CD and model retraining

# What is MLOps on Azure?

# What problems does MLOps solve?

- Reproducibility of models and predictions
- E2E operationalization & automation of ML lifecycle
- Collaboration within / across teams on ML workflow
- Enterprise readiness – security, governance, compliance, IaC

# Azure ML Platform - MLOps Capabilities

Manage Assets

- Infrastructure (Compute, Storage)
- Code
- Datasets
- Environments
- Runs
- Models

Understand Model Behavior

- Model Explainability
- Model Profiling
- Model Validation
- Model Deployment

Enable Enterprise ML Lifecycle Management

- Data Pipelines
- ML Pipelines
- Release Pipelines
- Data-Driven Eventing

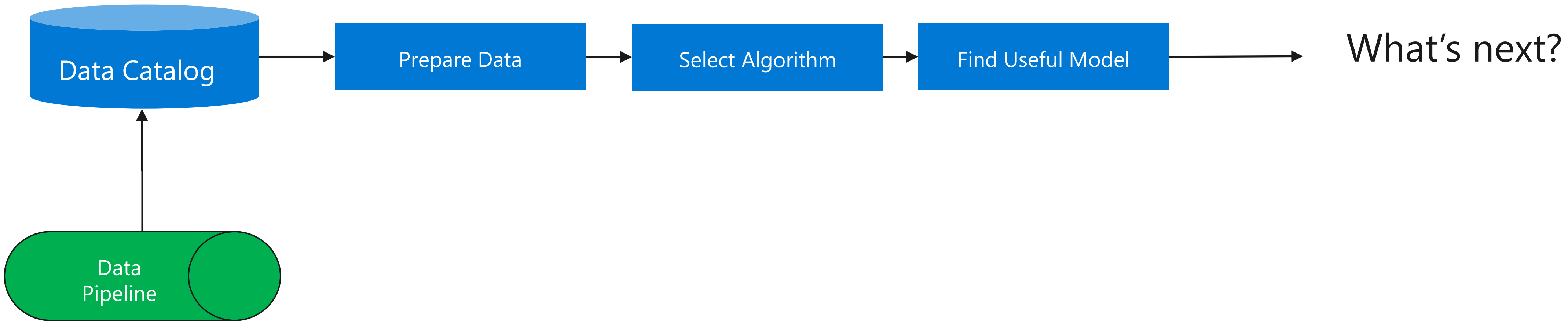
← ML Lineage & Audit Trail →

# ML Workflow – Process Maturity Model

# Phase 0 - Experimentation

Interactive, exploratory, get to something useful.

Data Scientist



Data Catalog

Prepare Data

Select Algorithm

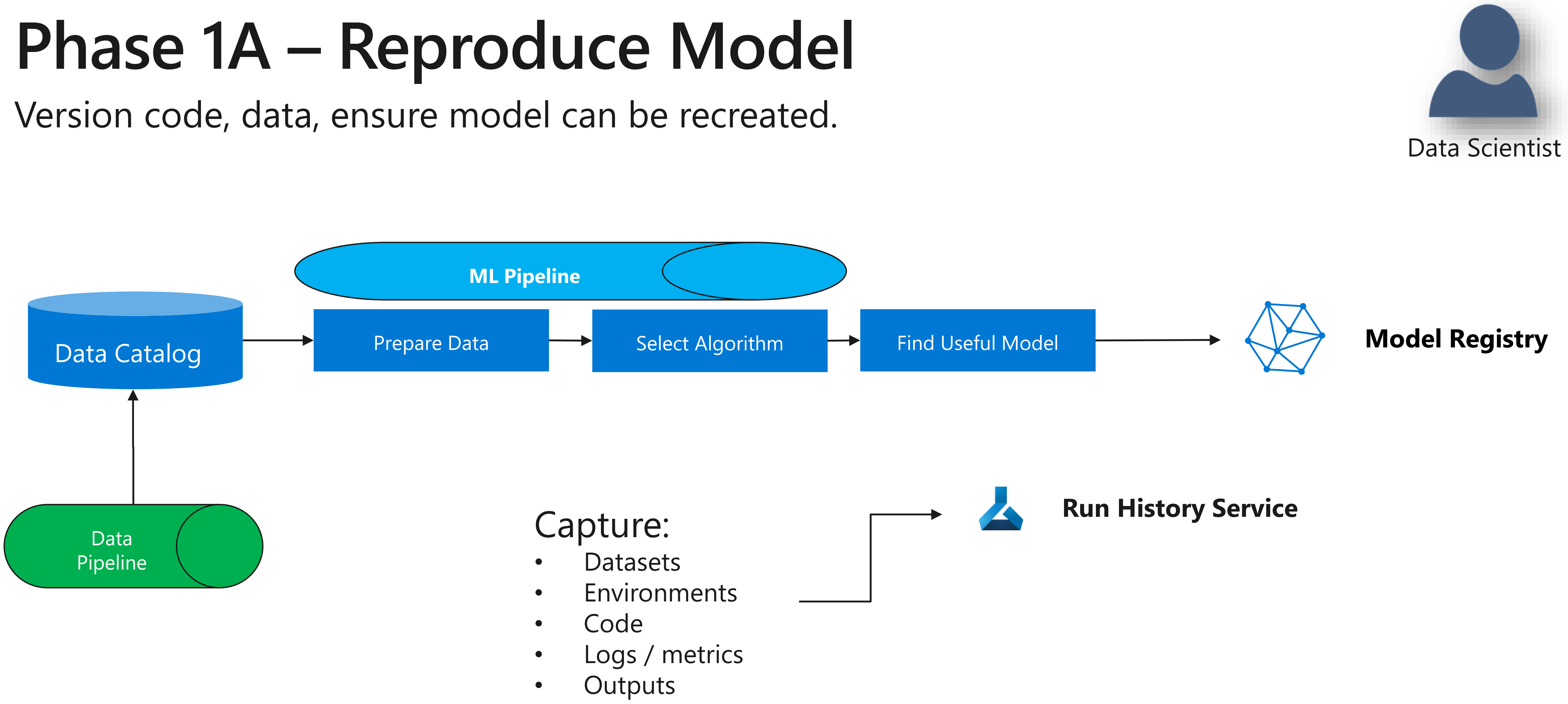
Find Useful Model

What’s next?

Data Pipeline

# Phase 1A – Reproduce Model

Version code, data, ensure model can be recreated.



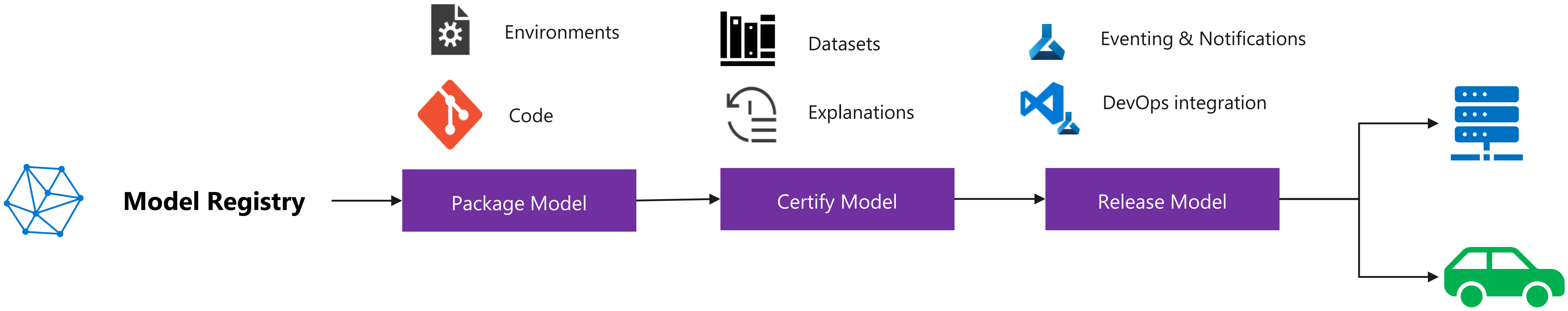
Capture:

- Datasets
- Environments
- Code
- Logs / metrics
- Outputs

# Phase 1B – Deploy Model

Package, certify, deploy

ML Engineer



Environments

Code

Datasets

Explanations

Eventing & Notifications

DevOps integration

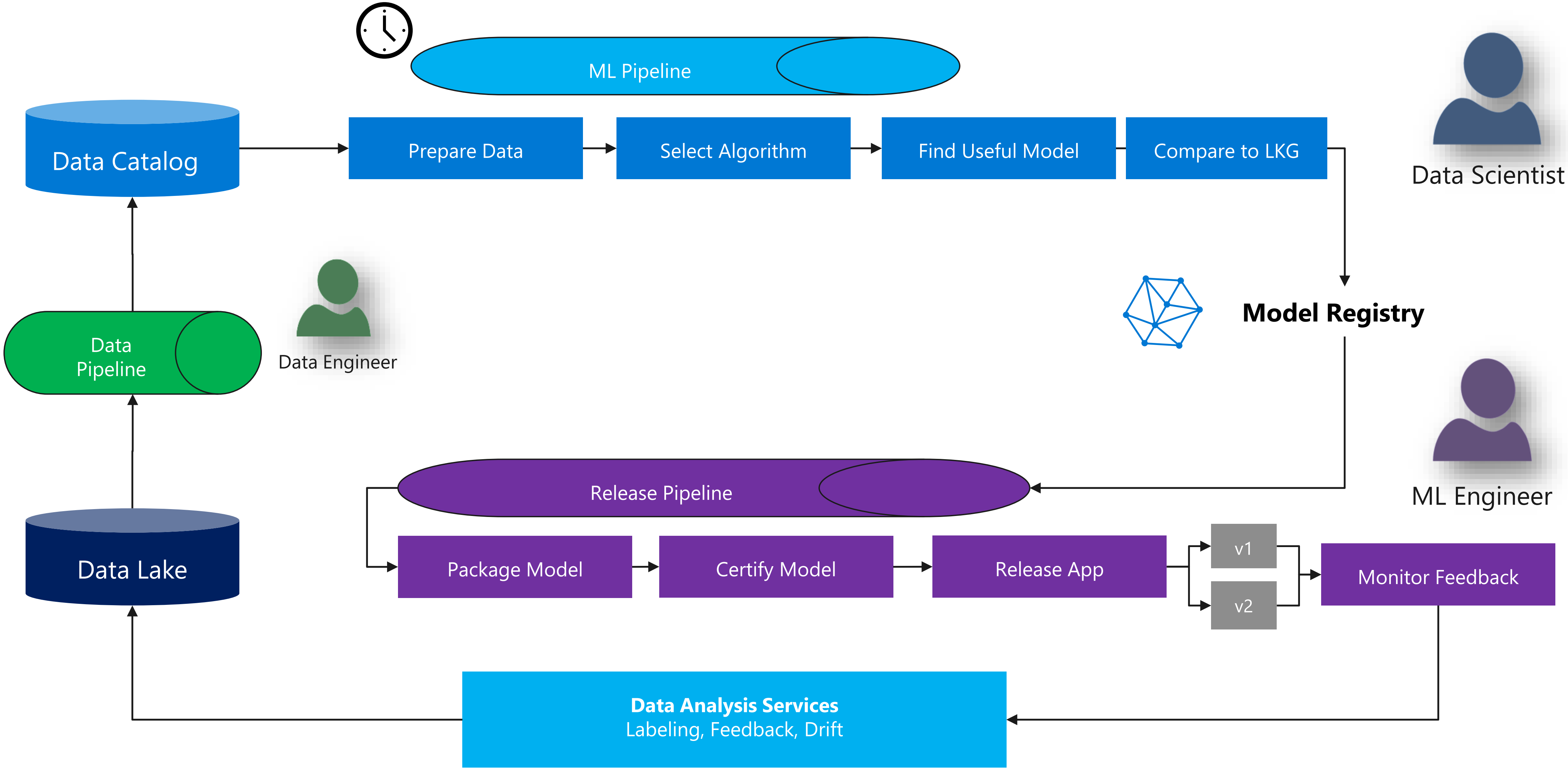
**Model Registry**

Package Model

Certify Model

Release Model

# Phase 2 – Automate ML Lifecycle



ML Pipeline

Data Catalog

Prepare Data

Select Algorithm

Find Useful Model

Compare to LKG

Data Scientist

Data Pipeline

Data Engineer

Model Registry

ML Engineer

Release Pipeline

Data Lake

Package Model

Certify Model

Release App

v1

v2

Monitor Feedback

**Data Analysis Services**
Labeling, Feedback, Drift

# What do we support?

**Asset Management**

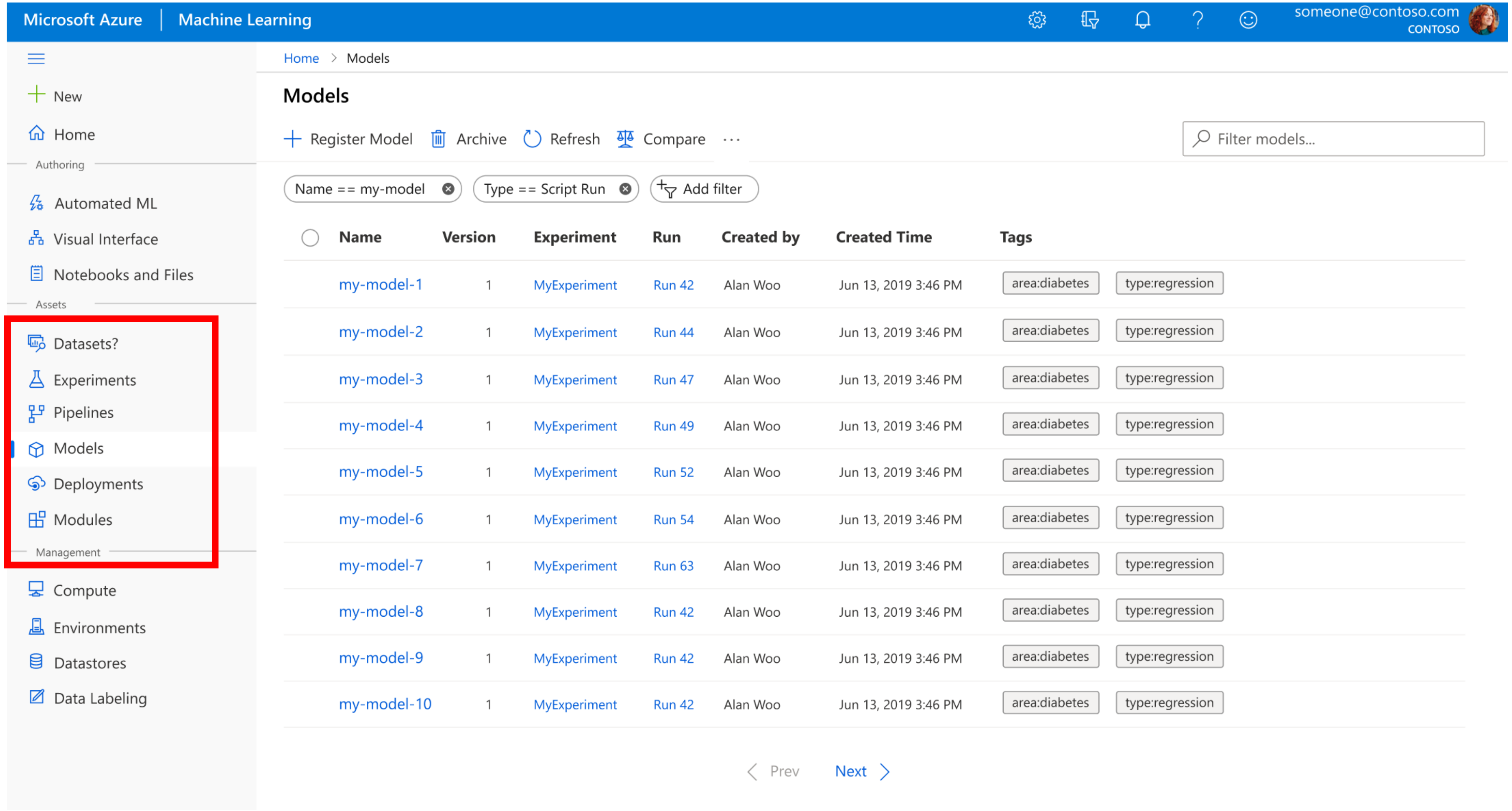
- Code
- Datasets
- Experiments
- Models

**Azure ML Services**

- Model packaging
- Model profiling
- Model deployment

**E2E Automation** w/ CLI + DevOps integration
**E2E Audit Trail** through asset management

# Asset Management

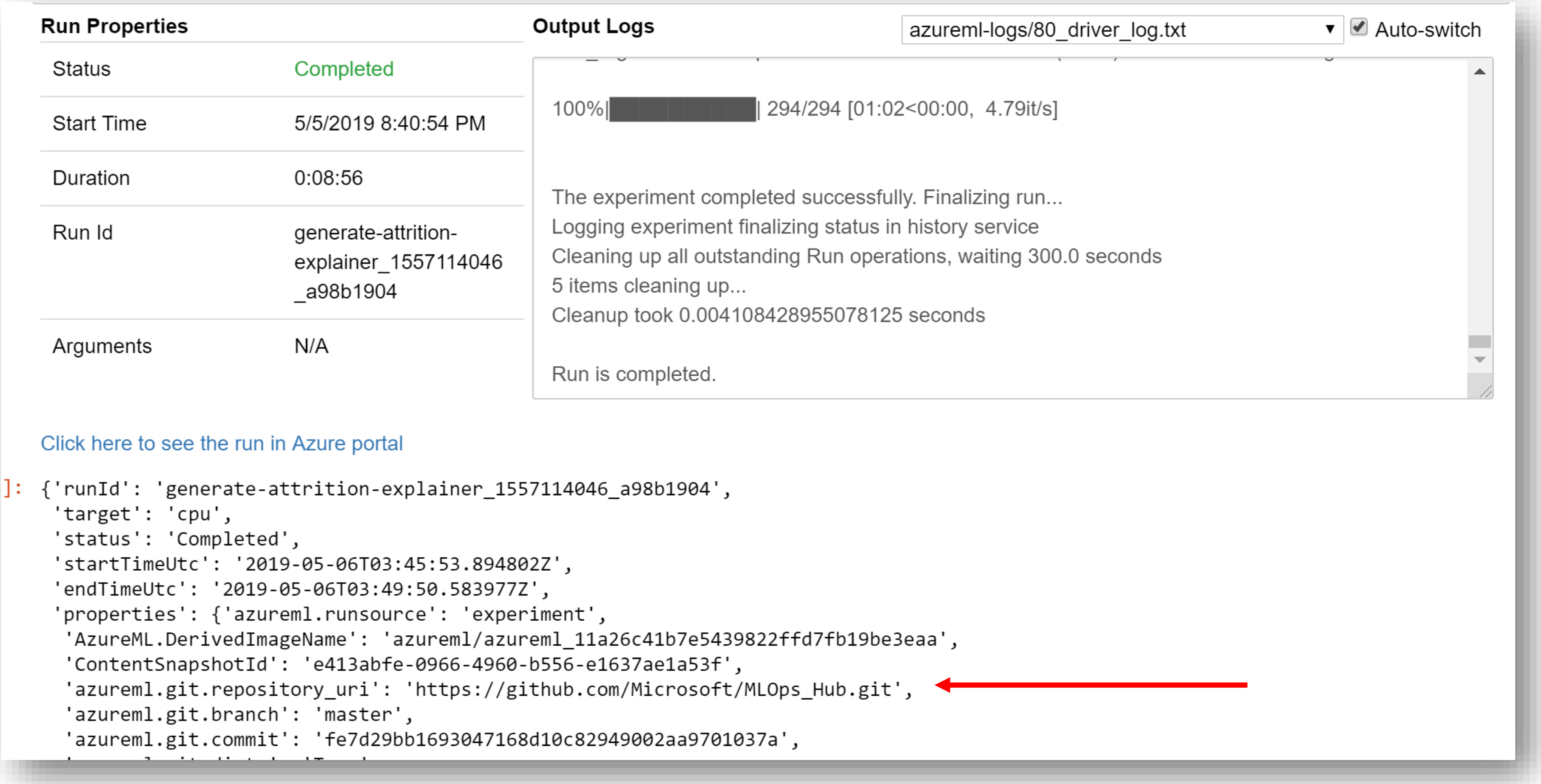


## Models

| Name | Version | Experiment | Run | Created by | Created Time | Tags |
|---|---|---|---|---|---|---|
| my-model-1 | 1 | MyExperiment | Run 42 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-2 | 1 | MyExperiment | Run 44 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-3 | 1 | MyExperiment | Run 47 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-4 | 1 | MyExperiment | Run 49 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-5 | 1 | MyExperiment | Run 52 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-6 | 1 | MyExperiment | Run 54 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-7 | 1 | MyExperiment | Run 63 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-8 | 1 | MyExperiment | Run 42 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-9 | 1 | MyExperiment | Run 42 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |
| my-model-10 | 1 | MyExperiment | Run 42 | Alan Woo | Jun 13, 2019 3:46 PM | area:diabetes<br>type:regression |

# Asset Management

## 1. Git Integration



**Run Properties**

| | |
|---|---|
| Status | Completed |
| Start Time | 5/5/2019 8:40:54 PM |
| Duration | 0:08:56 |
| Run Id | generate-attrition-explainer_1557114046_a98b1904 |
| Arguments | N/A |

**Output Logs** azureml-logs/80_driver_log.txt ☑ Auto-switch

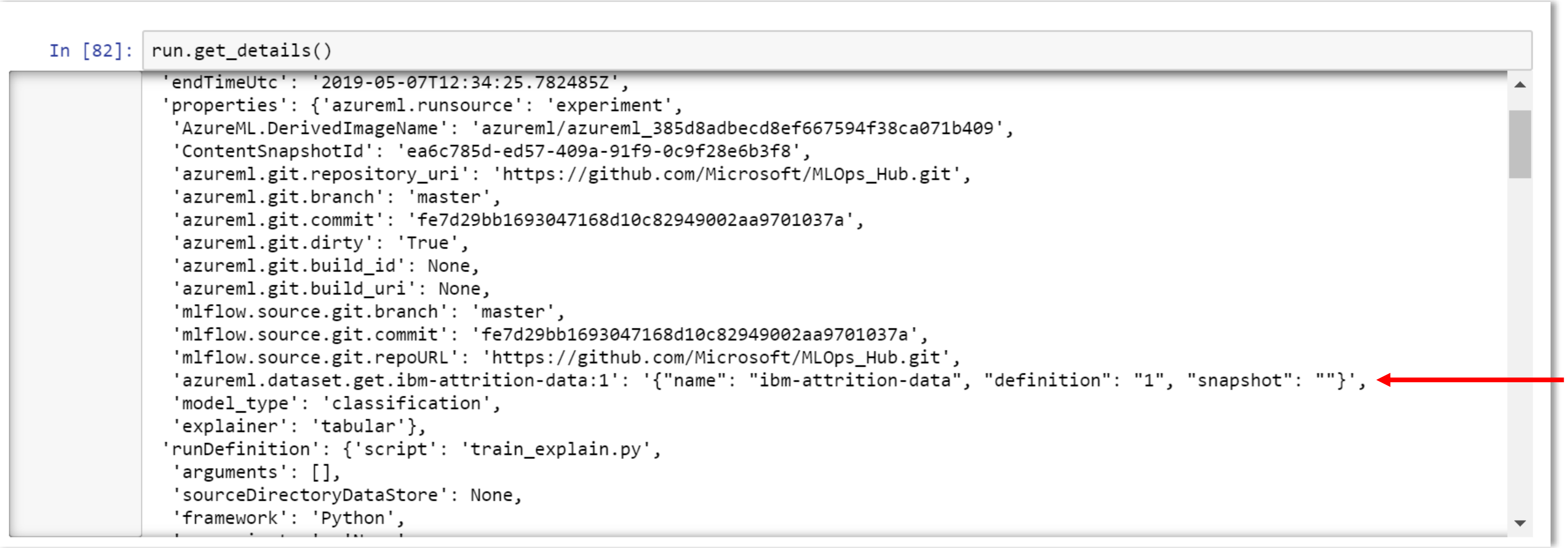
```
100%|██████████| 294/294 [01:02<00:00,  4.79it/s]


The experiment completed successfully. Finalizing run...
Logging experiment finalizing status in history service
Cleaning up all outstanding Run operations, waiting 300.0 seconds
5 items cleaning up...
Cleanup took 0.004108428955078125 seconds

Run is completed.
```

Click here to see the run in Azure portal

```
]: {'runId': 'generate-attrition-explainer_1557114046_a98b1904',
    'target': 'cpu',
    'status': 'Completed',
    'startTimeUtc': '2019-05-06T03:45:53.894802Z',
    'endTimeUtc': '2019-05-06T03:49:50.583977Z',
    'properties': {'azureml.runsource': 'experiment',
     'AzureML.DerivedImageName': 'azureml/azureml_11a26c41b7e5439822ffd7fb19be3eaa',
     'ContentSnapshotId': 'e413abfe-0966-4960-b556-e1637ae1a53f',
     'azureml.git.repository_uri': 'https://github.com/Microsoft/MLOps_Hub.git',
     'azureml.git.branch': 'master',
     'azureml.git.commit': 'fe7d29bb1693047168d10c82949002aa9701037a',
```

# Asset Management

## 2. Dataset Integration



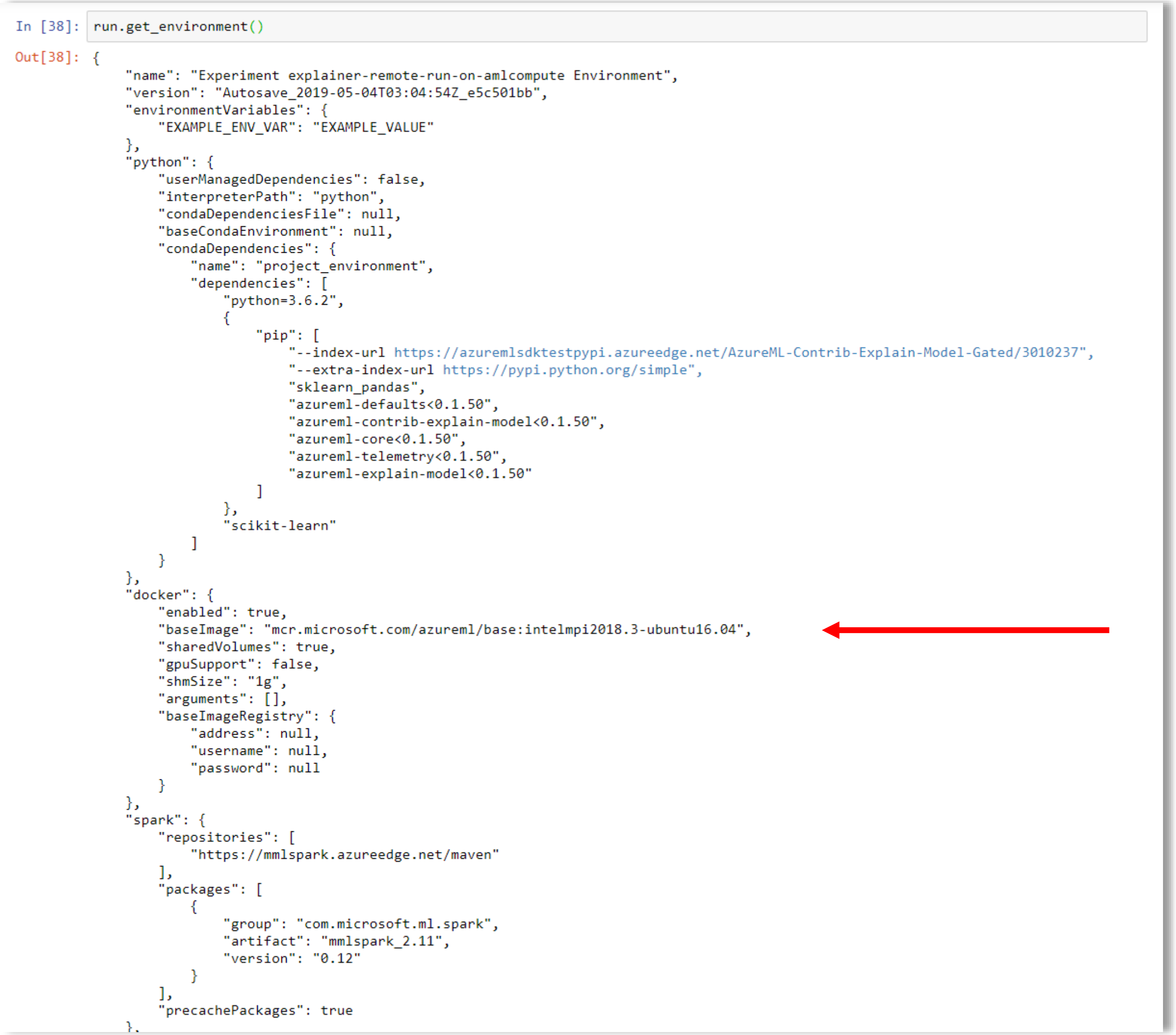
In [82]:

```
run.get_details()
```

```
 'endTimeUtc': '2019-05-07T12:34:25.782485Z',
 'properties': {'azureml.runsource': 'experiment',
  'AzureML.DerivedImageName': 'azureml/azureml_385d8adbecd8ef667594f38ca071b409',
  'ContentSnapshotId': 'ea6c785d-ed57-409a-91f9-0c9f28e6b3f8',
  'azureml.git.repository_uri': 'https://github.com/Microsoft/MLOps_Hub.git',
  'azureml.git.branch': 'master',
  'azureml.git.commit': 'fe7d29bb1693047168d10c82949002aa9701037a',
  'azureml.git.dirty': 'True',
  'azureml.git.build_id': None,
  'azureml.git.build_uri': None,
  'mlflow.source.git.branch': 'master',
  'mlflow.source.git.commit': 'fe7d29bb1693047168d10c82949002aa9701037a',
  'mlflow.source.git.repoURL': 'https://github.com/Microsoft/MLOps_Hub.git',
  'azureml.dataset.get.ibm-attrition-data:1': '{"name": "ibm-attrition-data", "definition": "1", "snapshot": ""}',
  'model_type': 'classification',
  'explainer': 'tabular'},
 'runDefinition': {'script': 'train_explain.py',
  'arguments': [],
  'sourceDirectoryDataStore': None,
  'framework': 'Python',
```

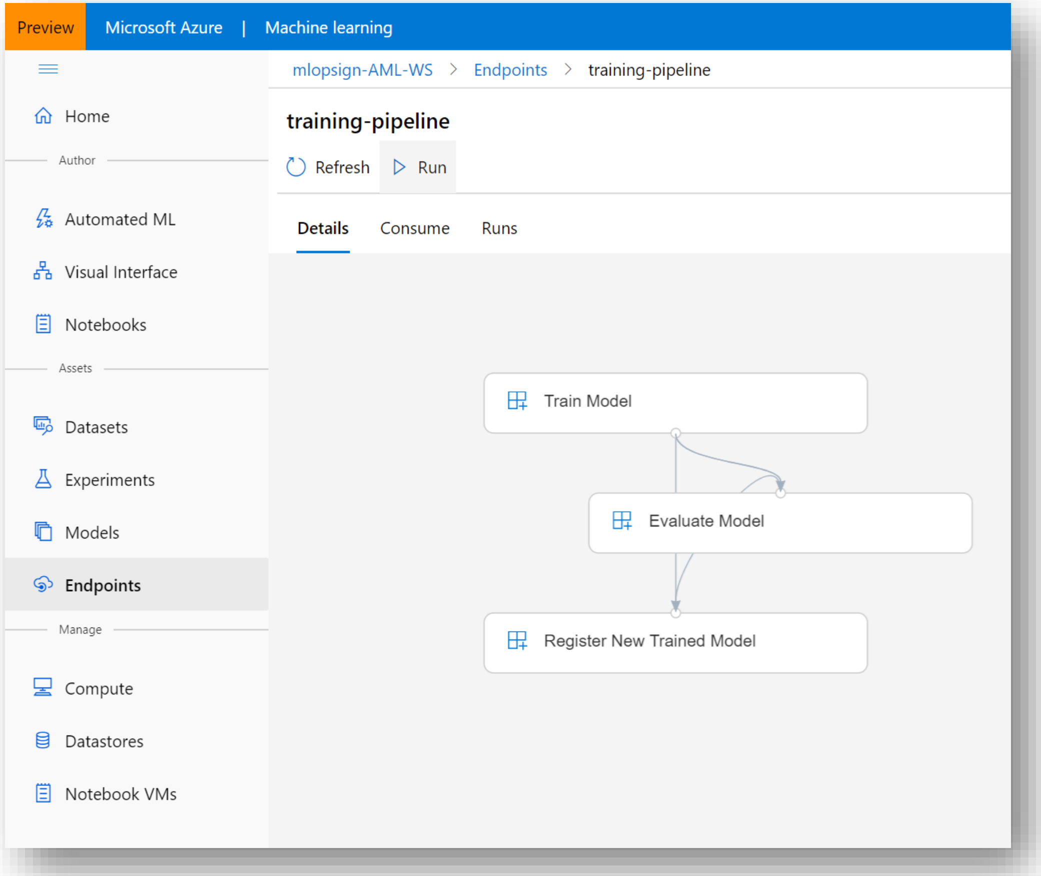
# Asset Management

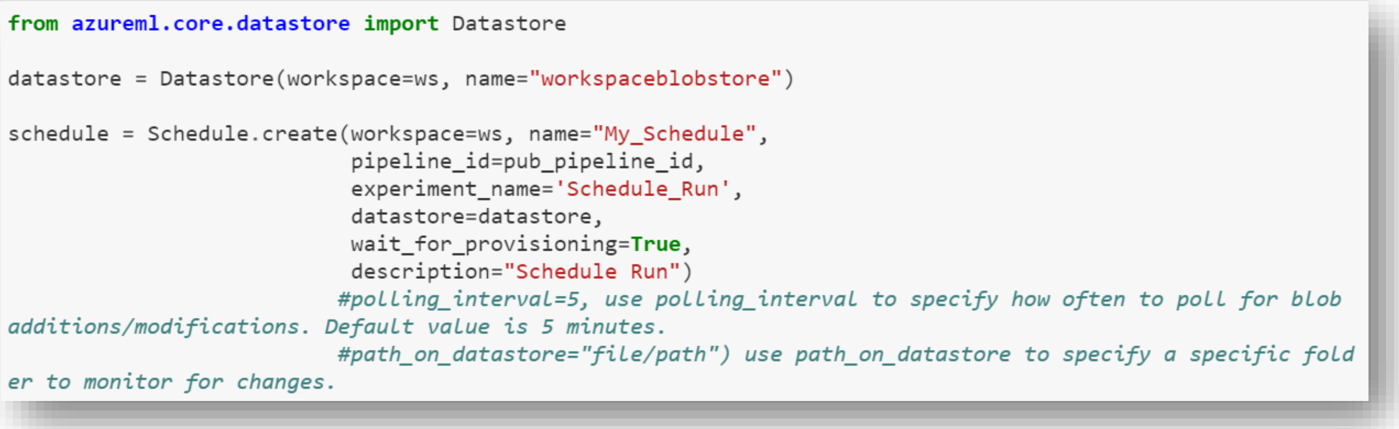
3. Environment



```
In [38]: run.get_environment()

Out[38]: {
             "name": "Experiment explainer-remote-run-on-amlcompute Environment",
             "version": "Autosave_2019-05-04T03:04:54Z_e5c501bb",
             "environmentVariables": {
                 "EXAMPLE_ENV_VAR": "EXAMPLE_VALUE"
             },
             "python": {
                 "userManagedDependencies": false,
                 "interpreterPath": "python",
                 "condaDependenciesFile": null,
                 "baseCondaEnvironment": null,
                 "condaDependencies": {
                     "name": "project_environment",
                     "dependencies": [
                         "python=3.6.2",
                         {
                             "pip": [
                                 "--index-url https://azuremlsdktestpypi.azureedge.net/AzureML-Contrib-Explain-Model-Gated/3010237",
                                 "--extra-index-url https://pypi.python.org/simple",
                                 "sklearn_pandas",
                                 "azureml-defaults<0.1.50",
                                 "azureml-contrib-explain-model<0.1.50",
                                 "azureml-core<0.1.50",
                                 "azureml-telemetry<0.1.50",
                                 "azureml-explain-model<0.1.50"
                             ]
                         },
                         "scikit-learn"
                     ]
                 }
             },
             "docker": {
                 "enabled": true,
                 "baseImage": "mcr.microsoft.com/azureml/base:intelmpi2018.3-ubuntu16.04",
                 "sharedVolumes": true,
                 "gpuSupport": false,
                 "shmSize": "1g",
                 "arguments": [],
                 "baseImageRegistry": {
                     "address": null,
                     "username": null,
                     "password": null
                 }
             },
             "spark": {
                 "repositories": [
                     "https://mmlspark.azureedge.net/maven"
                 ],
                 "packages": [
                     {
                         "group": "com.microsoft.ml.spark",
                         "artifact": "mmlspark_2.11",
                         "version": "0.12"
                     }
                 ],
                 "precachePackages": true
             },
```

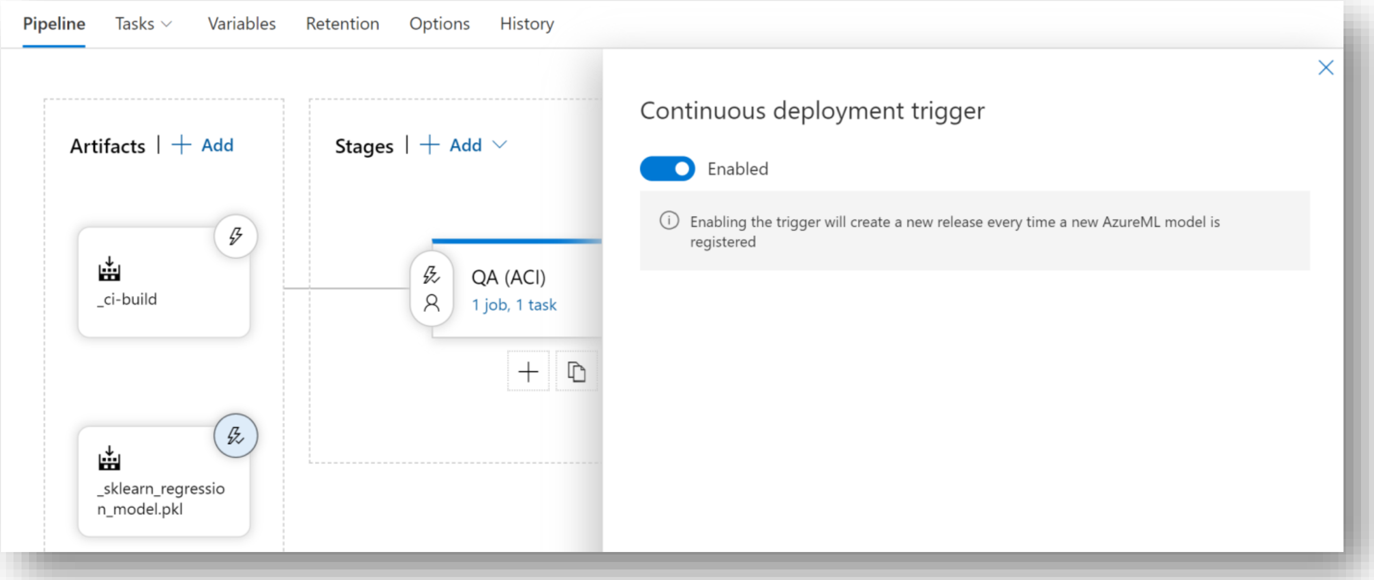
# Automated Training





```
from azureml.core.datastore import Datastore

datastore = Datastore(workspace=ws, name="workspaceblobstore")

schedule = Schedule.create(workspace=ws, name="My_Schedule",
                           pipeline_id=pub_pipeline_id,
                           experiment_name='Schedule_Run',
                           datastore=datastore,
                           wait_for_provisioning=True,
                           description="Schedule Run")
                          #polling_interval=5, use polling_interval to specify how often to poll for blob
additions/modifications. Default value is 5 minutes.
                          #path_on_datastore="file/path") use path_on_datastore to specify a specific fold
er to monitor for changes.
```

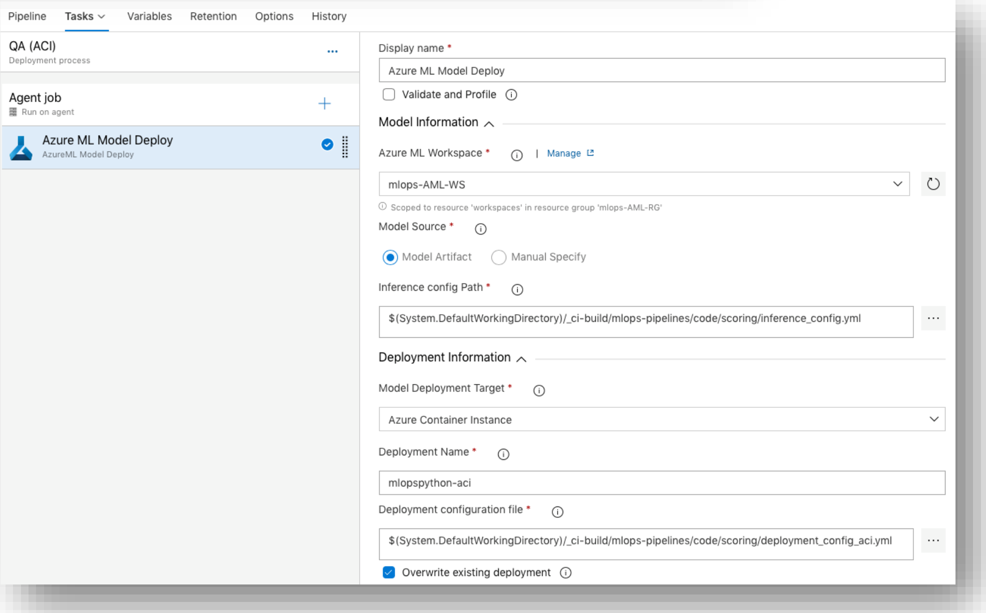
# Automated Deployment



Continuous deployment trigger

● Enabled

Enabling the trigger will create a new release every time a new AzureML model is registered



Display name *

Azure ML Model Deploy

□ Validate and Profile

Model Information

Azure ML Workspace *

mlops-AML-WS

Scoped to resource 'workspaces' in resource group 'mlops-AML-RG'

Model Source *

● Model Artifact ○ Manual Specify

Inference config Path *

$(System.DefaultWorkingDirectory)/_ci-build/mlops-pipelines/code/scoring/inference_config.yml

Deployment Information

Model Deployment Target *

Azure Container Instance

Deployment Name *

mlopspython-aci

Deployment configuration file *

$(System.DefaultWorkingDirectory)/_ci-build/mlops-pipelines/code/scoring/deployment_config_aci.yml

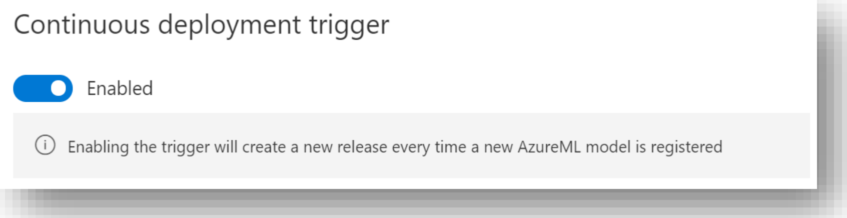
☑ Overwrite existing deployment

# E2E Automation

### Azure ML CLI

```
az ml model deploy -n mydetector -m videodetector:1 -
-ic inferenceConfig.json --dc deploymentConfig.json
```

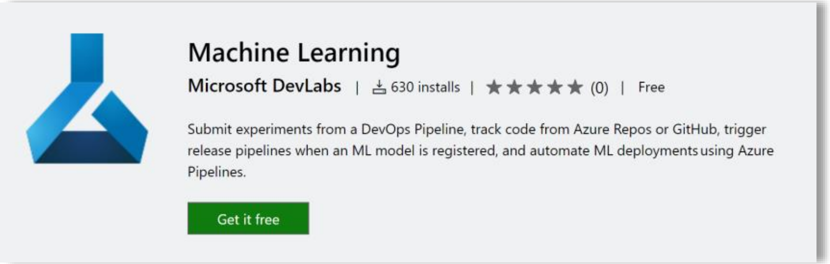
### Triggers to automate model  deployment using the Machine Learning extension



Continuous deployment trigger

Enabled

ⓘ Enabling the trigger will create a new release every time a new AzureML model is registered

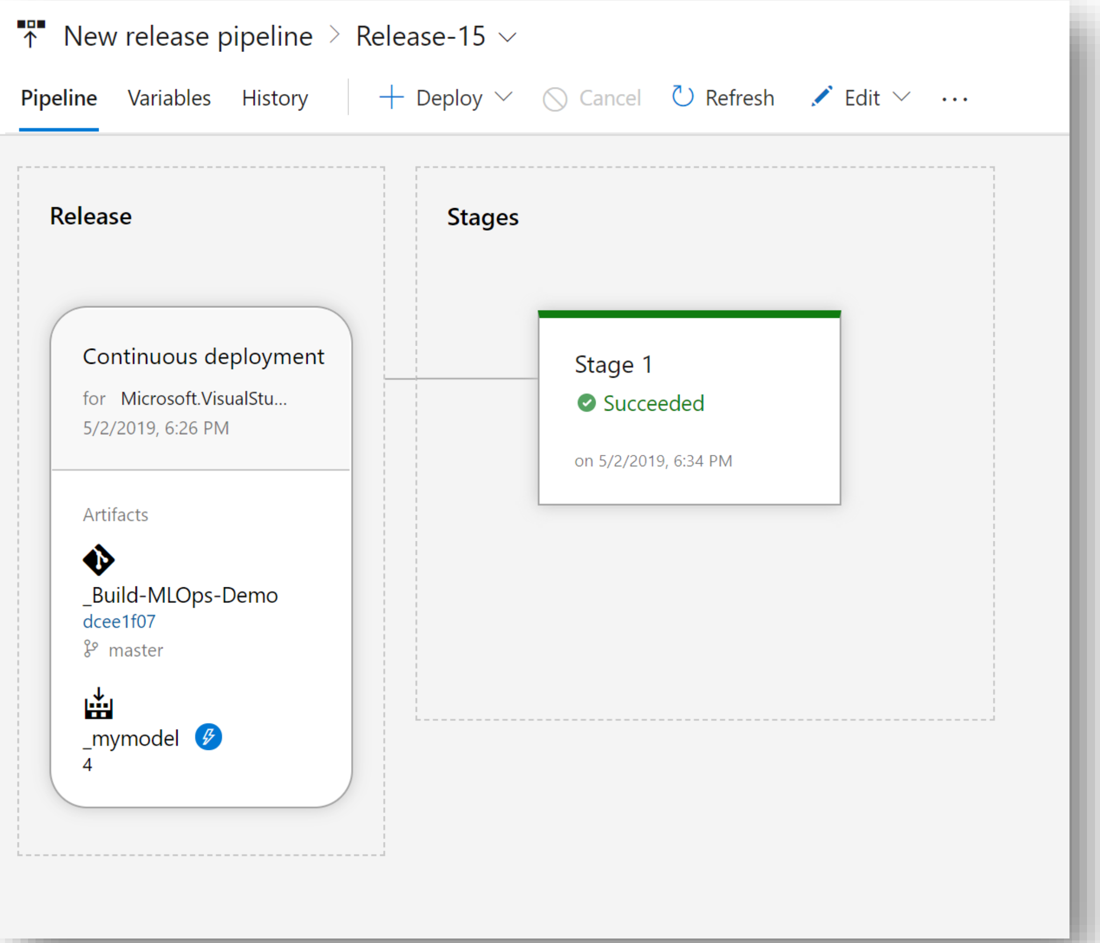


Machine Learning

Microsoft DevLabs | 630 installs | ★★★★★ (0) | Free

Submit experiments from a DevOps Pipeline, track code from Azure Repos or GitHub, trigger release pipelines when an ML model is registered, and automate ML deployments using Azure Pipelines.

### Integrated with Azure DevOps



Continuous deployment

for Microsoft.VisualStu...

5/2/2019, 6:26 PM

Artifacts

_Build-MLOps-Demo

dcee1f07

master

_mymodel

4

Stage 1

Succeeded

on 5/2/2019, 6:34 PM

# Demo video of MLOps E2E

- https://www.youtube.com/watch?v=1XJwcfQP7YI

# Key Takeaways

## Better together: ML + DevOps mindset

MLOps provides structure for building, deploying and managing and an enterprise-ready AI application lifecycle

## MLOps enhances delivery

Adoption will increase the agility, quality and delivery of AI project teams.

## More than technology

MLOps is a conversation about people, process and technology
AI principles and practices need to be understood by all roles