



MLOps with Azure Machine Learning

Keita Onabuta

Azure CXP – FastTrack for Azure
Senior Customer Engineer for AI/ML

本セッション概要

- 目的
 - MLOps の実現イメージを成熟度モデルを通じて理解する
 - Azure Machine Learning を利用した実装方法を理解する
- 対象者
 - データサイエンティスト、機械学習エンジニアなどの AI/ML に親しみがある方
 - MLOps を実装する技術に興味がある方

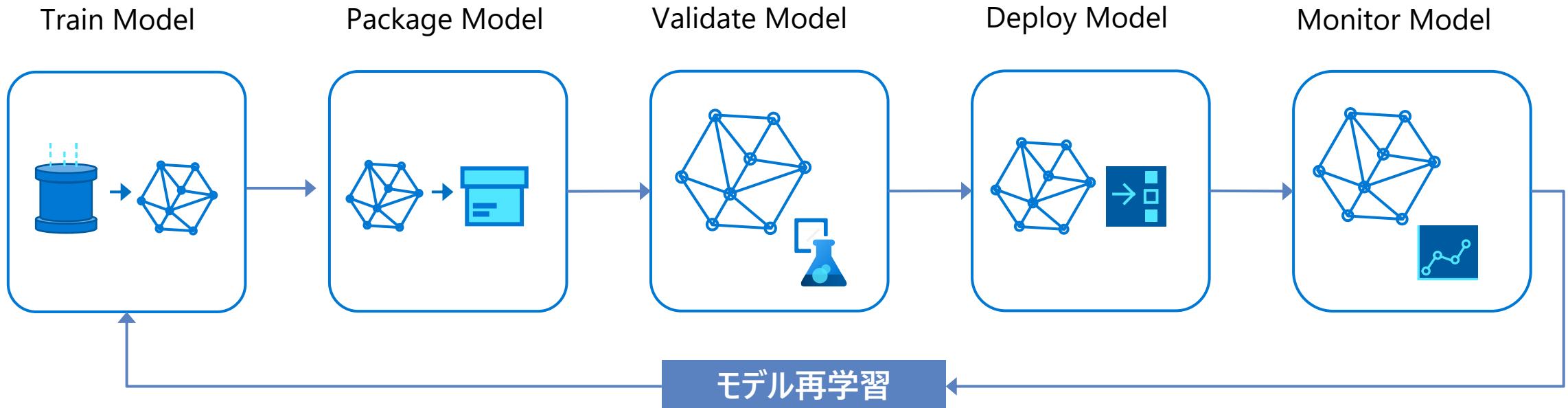
アジェンダ

- MLOps の概要
- Azure ML & GitHub 概要
- MLOps の実現イメージ

MLOps 概要

MLOps とは

機械学習ライフサイクルを管理する手法・概念



開発と学習

ビジネス課題を解決する

パッケージ化

モデルをどこでも使用可能にする

挙動の検証

応答性の観点と規制遵守の観点から

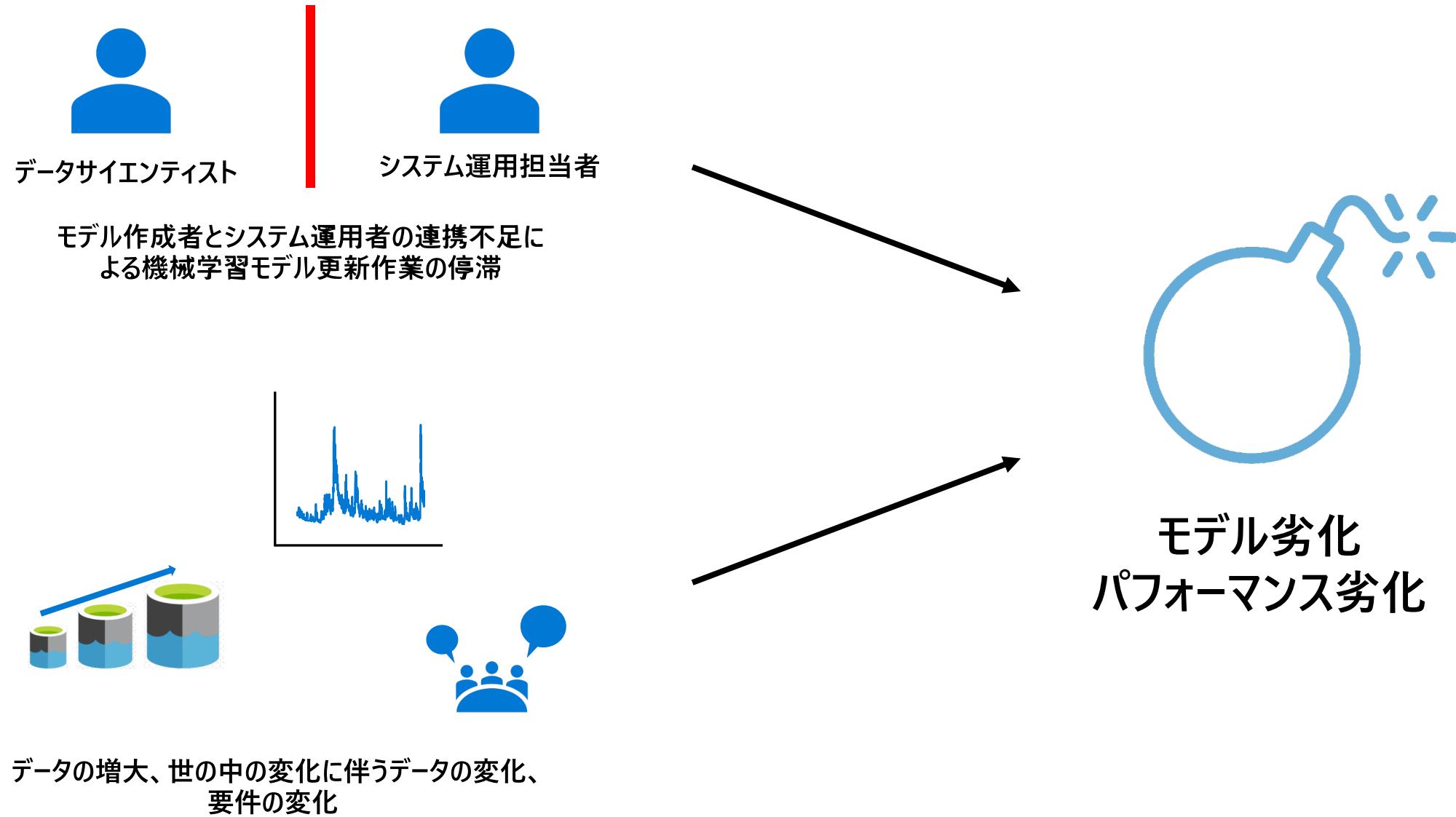
デプロイ

予測値の生成にモデルを使用する

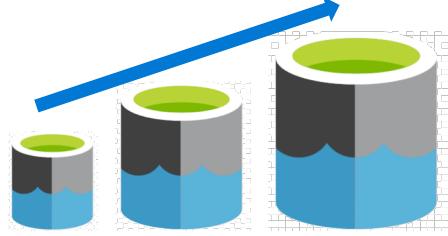
モニタリング

挙動とビジネス価値を監視し、陳腐化したモデルをいつ置換/廃止するか決める

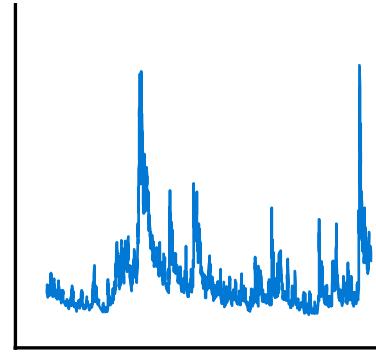
MLOps なき機械学習プロジェクト



継続的な改善が必要



学習に用いるデータは
日々増大していく



データの性質は日々変化
していく



使用状況やビジネス要件
の変化に伴って要望が生
じる



機動的なモデル改善と継続的なモデル作成(再学習)が必要

機械学習システム

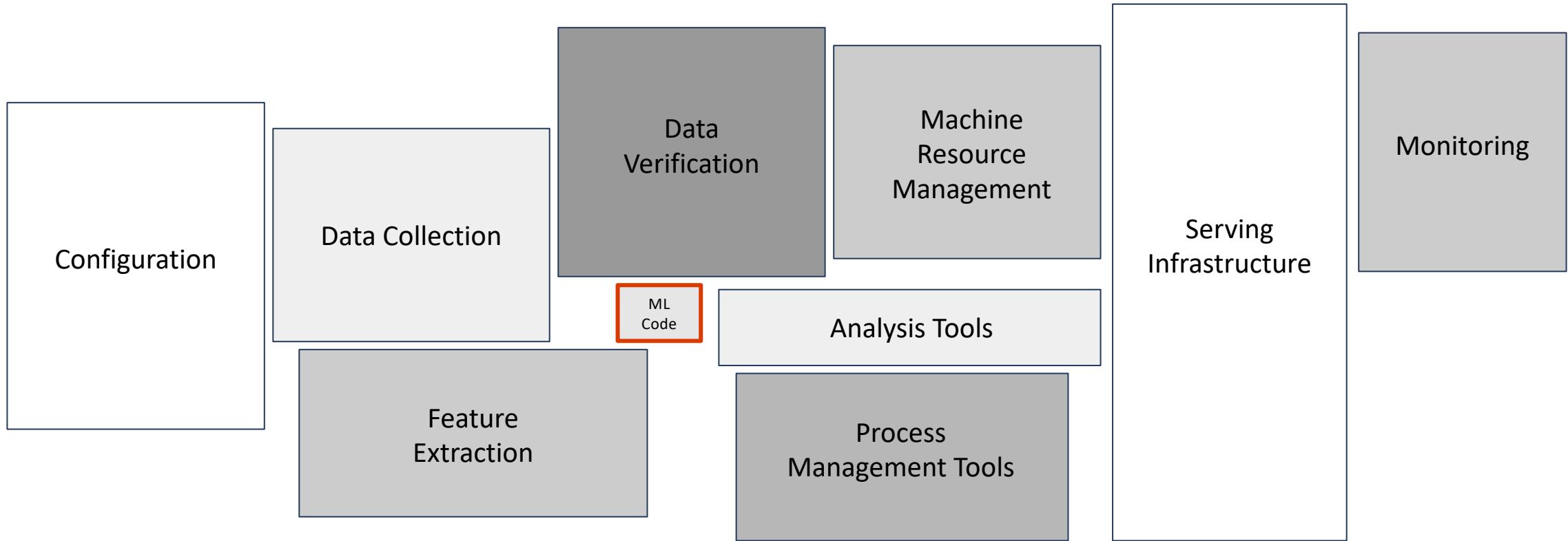


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small green box in the middle. The required surrounding infrastructure is vast and complex.

MLOps を支える要素



MLOps を実現するための要素



People

個々のエンジニアのコードをリポジトリに集約する。
コミットするたびに自動的にビルド/テストが行われ、バグがより早く検出可能である。
コード、データ、モデル、トレーニングパイプラインが共有され、イノベーションが加速する。



Process

コード化されたインフラ (IaC) やモデル開発環境の自動化をブーストするテンプレートを提供する。
コードのコミットから本番までのプロセスを自動化する。



Platform

機能の準備が整い次第、顧客に安全に提供することが可能である。

運用中のパイプライン、インフラ、製品を監視し、期待通りの動作をしていないことを検知できる。

成熟度モデル

Level 0 No MLOps

概要

- 機械学習モデルのライフサイクル全体を管理することは困難
- チームは別々で、リリースは困難
- ほとんどのシステムは "ブラック ボックス" として存在し、デプロイ時およびデプロイ後のフィードバックはほとんどなし

技術

- 手動によるビルドとデプロイ
- モデルおよびアプリケーションの手動によるテスト
- モデルのパフォーマンスの一元的追跡なし
- モデル学習は手動

Level 1 DevOps no MLOps

- 自動ビルド
- アプリケーション コードの自動テスト

Level 2 Automated Training

- トレーニング環境は完全に管理され、追跡可能
- モデルの再現が容易
- リリースは手動であるが、摩擦は少ない

- 自動化されたモデルの学習
- モデル学習のパフォーマンスを一元的に追跡
- モデル管理

Level 3 Automated Model Deployment

- リリースは低摩擦で自動
- デプロイから元のデータまで完全に追跡可能
- 環境全体 (学習 > テスト > 運用) を管理

- デプロイするモデルのパフォーマンスに関する A/B テストを統合
- すべてのコードのテストを自動化
- モデルの学習性能を一元化

Level 4 Full MLOps Automated Retraining

- システムを完全自動化し、監視を容易化
- 運用システムは、改善方法に関する情報を提供。場合によっては、新しいモデルで自動的に改善
- ゼロ ダウンタイム システムに近づく

- モデル学習とテストを自動化
- デプロイされたモデルからの詳細で一元化されたメトリック

MLOps のベネフィット



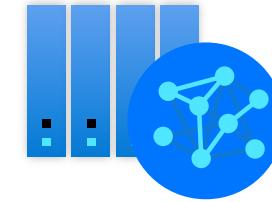
自動化/ 可観測性

コード駆動の生成とデプロイ
再現性と検証可能性を持つ
たパイプライン
全ての成果物にタグを付け、
監査することが可能



検証

SWEの品質管理のベストプラク
ティス
オフラインでのモデル品質の比
較
バイアスの最小化と説明性の確保



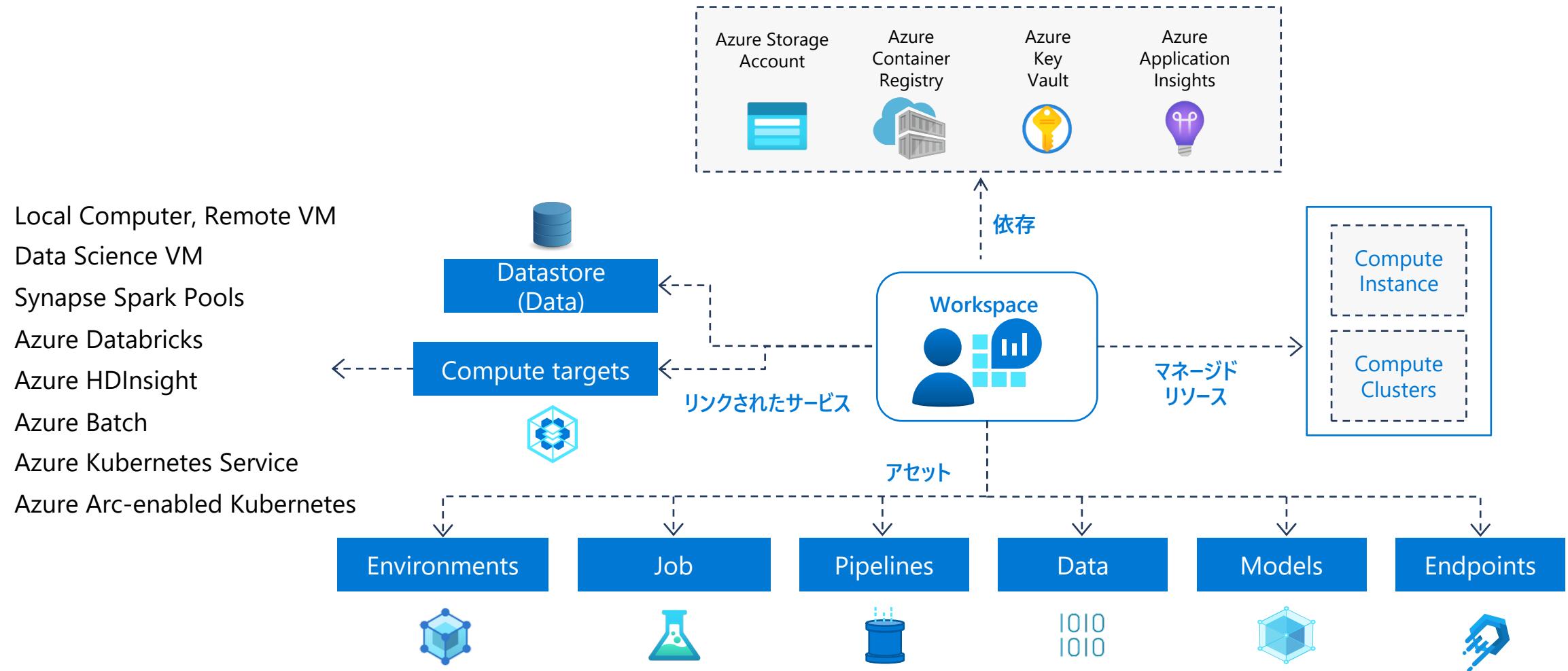
再現性/監査可能性

制御されたロールアウト機能
予測性能と期待性能のライブ
比較
ドリフトの監視とモデル改善に使
用できるフィードバック

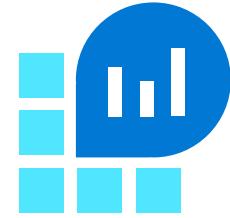
== ML モデルの高速展開と品質・セキュリティ

Azure Machine Learning 基本構成

Azure Machine Learning Workspace は、最上位のリソースであり、あらゆるリソースを操作・運用管理するための一元的な環境を提供します。



Azure Machine Learning 基本構成 (cont'd)



Workspace



データ (Data)

モデル学習で利用するデータの運用管理



モデル (Models)

学習済み機械学習モデルを運用管理



コンピューティング (Computes)

モデル学習や推論で利用する計算環境



パイプライン (Pipelines)

モデル学習や推論のプロセスをパイプライン化



環境 (Environments)

Python パッケージ、Docker イメージを管理



エンドポイント (Endpoints)

クラウドやエッジデバイスにモデルをデプロイ



ジョブ (Job)

ジョブ実行とメトリックやログの管理



データラベリング (Data Labelling)

画像やテキストへラベルを付与



Workspace

- Workspace は Azure Machine Learning の最上位リソースである。Azure Machine Learning 使用時に作成したあらゆる成果物を集約した一元的な作業スペースを提供する。
- Workspace はモデル学習で利用した Compute Targets の情報やログ、メトリック、出力、スクリプトのスナップショットを含む学習の実行履歴を保持する。

- モデルは Workspace に登録される。
 - 複数の Workspace を作成することができ、それぞれの Workspace は複数人で共有することができる。
 - 新しく Workspace を作成したとき、自動的に以下 Azure リソースが作成される:
 - [Azure Container Registry](#)
 - 学習とデプロイに使用する Docker コンテナの登録に利用される
 - [Azure Storage](#)
 - Workspace のデフォルトの Datastore やログ管理などに使用される
 - [Azure Application Insights](#)
 - 推論エンドポイントをモニタリングした情報が保存される
 - [Azure Key Vault](#)
 - Compute Targets によって使用されるシークレットとその他 Workspace が使用するセンシティブな情報が保存される



Data

- Datastores は Azure のストレージサービスに対する接続情報を保持するために使用されます。
- Data asset、MLTable は Datastores を使用して接続された各種データソース内に保存されたデータを参照する。

Datastores

- Datastores は認証情報や元のデータソースの完全性を危険にさらすことなく接続に必要な情報を保持する。サブスクリプション ID や認証トークンといった接続に必要な情報は Workspace に紐づけられた Key Vault に保存しているため、スクリプトに機密情報を直接実装することなくストレージに対してセキュアにアクセスできる。

Data asset, MLTable

- Azure Machine Learning で利用するデータへのアクセスが容易になる。Data asset を作成するとデータへの参照とメタデータのコピーが作成される。データは元々データがあった位置に維持されるため、余計なストレージコストをかけず、データソースの一貫性を維持することができる。
- MLTable は Data asset の中でも表形式データに対応している。



Environments

Environments は機械学習を実行するソフトウェア環境の構築や運用管理の機能を提供する。

学習や推論スクリプトで利用する Python パッケージ、環境変数、ソフトウェアの設定、ランタイム（Python、Spark や Docker 等）で定義され、最終的には Docker コンテナとして実行される。

Workspace 上で管理・バージョニングされ、再現性、監査可能性、機械学習ワークフローの異なる計算リソース間での相互利用性を確保する

Environment はローカルコンピューター上で以下のように使用することが可能

- 学習スクリプトの開発
- モデルの学習をスケーリングする際、Azure Machine Learning の計算リソース上で同一環境の再利用
- 同一環境を使用したモデルのデプロイ
- 既存モデルの学習時に使用された環境の再現
- Workspace のデフォルト環境として利用可能な curated environments （代表的なシナリオに沿ってプリセットされた environment）の提供



Job

Job は Experiment と Run から構成され、モデル学習などのプロセスを実行したり、結果を管理します。

Experiment は1つのスクリプトに由来する複数の Run をグループ化したもので、Azure ML Workspace に所属する。各 Run の情報は関連付けられた Experiment の配下に保存される。

Experiment

スクリプトの実行によって生成された複数の Run をグルーピング

- Azure ML Workspace 直下に所属
- 各 Run の情報を保存し、横断的に管理

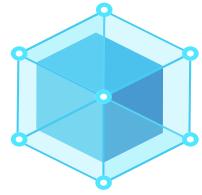
Run

Job を submit することで生成され、以下の情報を保持

- 実行のメタデータ (実行した日時、実行に要した時間等)
- スクリプトから出力したメトリック
- 実験に関する自動収集された、もしくは明示的にアップロードされたファイル
- スクリプトを含むディレクトリの実行直前のスナップショット

Run configuration

計算リソース上でスクリプトがどのように実行されるべきかの定義



Models

Model は学習を経て生成される artifact* である。Model は新しいデータに対する予測結果を得るために使用する。

Model Registry は Azure Machine Learning Workspace 上の全ての Model 追跡する。

* 機械学習の過程で生じる中間生成物、画像、モデル等のあらゆるファイルを artifact と呼称する

Model

Model は Azure Machine Learning 上の Run から生成される

- Note: Azure Machine Learning の外部で学習したモデルを使用することも可能

Azure Machine Learning はフレームワーク非依存であるため Model を作成する際にあらゆる機械学習フレームワークを使用可能

Model は Azure Machine Learning の Workspace 配下に登録可能

Azure Machine Learning に作成された Model はバージョニングされ、モデルのバージョンを比較・選択するためのツールが組み込みで付属

一度 Model が作成されれば、Dockerfile や Docker イメージの生成とあらゆる場所へのモデルのデプロイが可能

Model Registry

Azure Machine Learning Workspace 上の全モデルを追跡

Model は名前とバージョンによって一意に特定される

Model を登録するときに追加でメタデータタグを付与することができ、タグは Model の検索時に利用可能

Docker イメージに使用されている Model は削除できない



Pipelines

1つの Azure Machine Learning Pipeline は 完成した機械学習タスクのワークフローを表し、 機械学習の各サブタスクはパイプラインを構成 数する一連の手順として内包される

Azure Machine Learning Pipeline には、Python スクリプトを呼び出すだけのシンプルなものから、あらゆることを実行するものまで存在する。

タスク

Pipelines は以下のようないくつかの機械学習タスクにフォーカス:

- インポート、検証とクリーニング、変換、正規化、ステージングを含む「データの準備」
- パラメーター化された引数、ファイルのパス、ログとレポート出力の設定を含む「学習の設定」
- 効率的かつ繰り返し実行できる「学習と検証」。特定のデータサブセット、異なるハードウェアからなる計算リソース、分散処理、進捗状況の監視を指定することで、効率性を確保することが可能
- バージョン管理、スケーリング、プロビジョニング、アクセス制御等の要素を含む「デプロイ」



Computes

Computes は学習スクリプトを実行したりモデルを推論用途でホストするための計算リソースです。

Azure Machine Learning Python SDK, CLI, Studio から作成し運用管理することができます。

既存のリソースをアタッチすることもできます。

ローカル環境で実行したのちに、Compute Targets 上でスケールアップ・スケールアウトすることができます。

現在サポートされている Compute Target

Compute Targets	学習	デプロイ
Local Computer	✓	
A Linux VM in Azure (such as the Data Science Virtual Machine)	✓	
Azure ML Compute Clusters	✓	✓
Azure ML Compute Instance		
Azure Databricks	✓	
Azure Data Lake Analytics	✓	
Azure HDInsight	✓	
Azure Container Instance		✓
Azure Kubernetes Service		✓
Azure IoT Edge	✓	
Field-programmable gate array (FPGA)		✓
Azure Functions (preview)		✓
Azure App Service (preview)		✓
Azure Synapse Spark Pool (preview)	✓	
Azure Arc enable Kubernetes (preview)	✓	✓



GitHub

- ・ Git によるソースコード管理
- ・ 多種多様なサービスとの強力な連携機能
- ・ Github Actions によるワークフロー構築

The screenshot shows a GitHub repository page for 'Shuntaldo / azureml-pl-sample'. The 'Code' tab is selected, displaying a list of files and their descriptions:

- pred: Add pred directory (4 months ago)
- train: Support lr tuning (3 months ago)
- deploy_bert_aml.ipynb: update fine-tuning (4 months ago)
- dist_train_bert_aml.ipynb: update fine-tuning (4 months ago)
- train_bert_aml.ipynb: update fine-tuning (4 months ago)
- train_hyperparameter_tune_bert_aml.ipynb: Register best model (3 months ago)

At the bottom of the page, there are links for GitHub footer: © 2021 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, About.



GitHub Actions

- ・イベント駆動でジョブを実行する GitHub のワークフローツール

The screenshot shows the GitHub Actions interface for a job named 'check-bats-version'. The job status is 'succeeded 8 minutes ago in 9s'. The job consists of several steps:

- > Set up job (3s)
- > Run actions/checkout@v2 (1s)
- > Run actions/setup-node@v1 (2s)
- > Run npm install -g bats (2s)
- > Run bats -v (0s)
 - 1 ► Run bats -v
 - 4 Bats 1.2.1
- > Post Run actions/checkout@v2 (1s)
- > Complete job (0s)

A search bar at the top right is labeled 'Search logs'.



Azure Repos

- Azure DevOps に含まれるソースコード管理機能
- Git※もしくは TFVC
- 容量無制限
- AAD ベースの権限管理

The screenshot shows the Azure DevOps interface for the 'nlp-samples' repository. The left sidebar lists various project management features like Overview, Boards, and Pipelines. The main area displays the repository's file structure and history. The 'Files' tab is selected, showing a list of files including '.devcontainer', '.github', 'examples', '.gitignore', 'cgmanifest.json', 'CODE_OF_CONDUCT.md', 'LICENSE', 'NOTICE.md', 'README.md', 'SECURITY.md', and 'SUPPORT.md'. Each file entry includes a preview icon, the file name, the last change date, and the commit hash. A search bar at the top right allows users to find specific files or folders.

Name	Last change	Commits
.devcontainer	8月19日	c1adac24 devcontain...
.github	8月24日	8ffd5f2b Update rin...
examples	8月26日	460e49ce add comm...
.gitignore	8月17日	7cc6f6a5 Add cache ...
cgmanifest.json	7月21日	703d6714 init Keita O...
CODE_OF_CONDUCT.md	7月21日	675d9aec CODE_OF_...
LICENSE	7月21日	2d312e61 LICENSE c...
NOTICE.md	7月21日	703d6714 init Keita O...
README.md	8月18日	fb3da8cd add exampl...
SECURITY.md	7月21日	0908c685 SECURITY...
SUPPORT.md	7月27日	25e16cb8 init Keita O...

※ Azure Machine Learning は Git と連携可能 (TFVC は未対応)

<https://azure.microsoft.com/ja-jp/services/devops/repos/>



Azure Pipelines

- Azure DevOps に含まれるパイプラインツール
- GitHub Actions と比較すると、Azure と緊密に連携した処理に強み

The screenshot shows the Azure DevOps Pipelines interface. At the top, there's a navigation bar with the Azure DevOps logo, a search bar, and various icons. Below the navigation, a progress bar indicates the steps: Connect (checkmark), Select (checkmark), Configure (checkmark), and Review (highlighted). A message about an upcoming change to Azure DevOps organizations in East Asia is displayed. The main area is titled "Review your pipeline YAML" and shows a YAML configuration file for a pipeline named "azure-pipelines.yml". The file includes comments explaining the purpose of the pipeline and its components like triggers, pool, and steps. On the right side of the code editor, there are buttons for "Variables" and "Save and run". Below the code editor, there's a link to "Show assistant".

```
# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
- aml-datasets

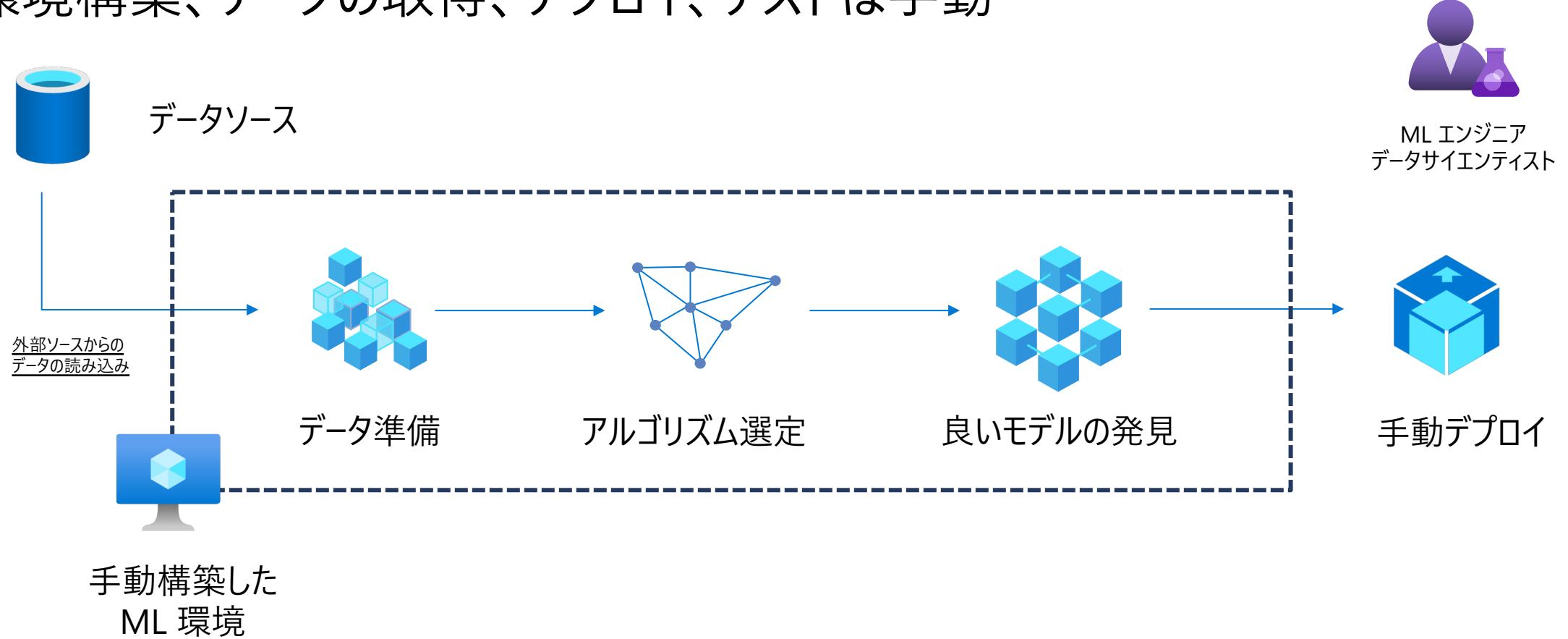
pool:
vmImage: ubuntu-latest

steps:
- script: echo Hello, world!
  displayName: 'Run a one-line script'
- script: |
  echo Add other tasks to build, test, and deploy your project.
  echo See https://aka.ms/yaml
  displayName: 'Run a multi-line script'
```

MLOps の実現イメージ

Level 0 – No MLOps

- ・ インタラクティブかつ実験的に有益なモデルを探す
- ・ 環境構築、データの取得、デプロイ、テストは手動



課題・チャレンジ

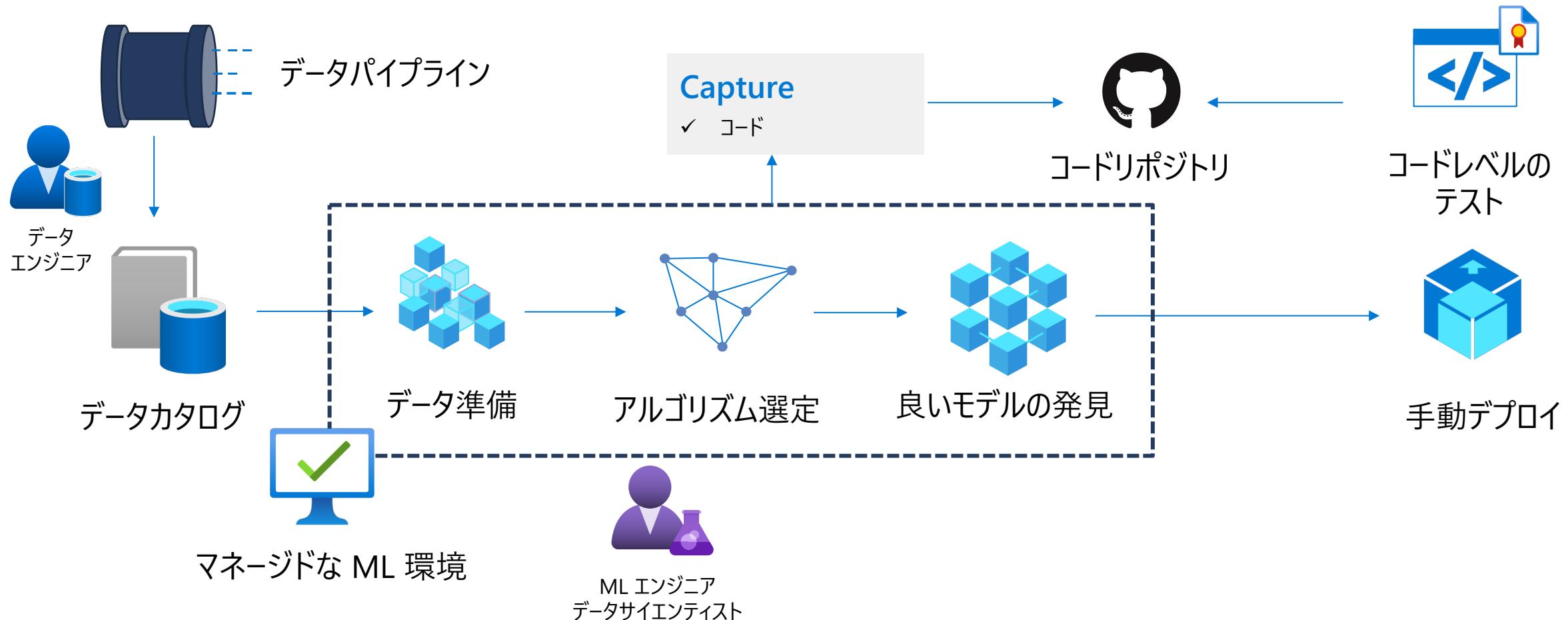
- ・ 再現性
 - ・ データサイエンティストが独自で調達したツールを使い、コードやライブラリの情報が共有されていないため、実験を再現するのが難しい。
- ・ 品質&セキュリティ
 - ・ テストが実施されていなかったり、設計・考慮されていない。
 - ・ IT 部門でツールを管理していないためセキュリティの懸念がある。
- ・ スケーラビリティ
 - ・ 計算リソースのパワー不足でジョブがスケールしない。
 - ・ 高性能の計算リソースが組織で共有されず、無駄なコストが発生している。
- ・ その他
 - ・ 機械学習で用いるデータソースがメンテナンスされておらず、手動でデータを収集する必要がある。

Lv0 → Lv1

- ・組織で標準的な機械学習プラットフォームを準備し、高性能な計算リソースを共有で利用する。
- ・コードリポジトリを準備し、データサイエンティストや関係者間でコードを共有する。
- ・Notebook を Python コードに変換し、リファクタリングする。
- ・学習コード & 推論コードに対するテストとその自動化を設定する。
- ・機械学習に必要なデータを取得するパイプラインを構築する。

Level 1 – DevOps no MLOps

- ・データパイプラインとマネージドな ML 環境が整備される
- ・テストが整備され始める



課題・チャレンジ

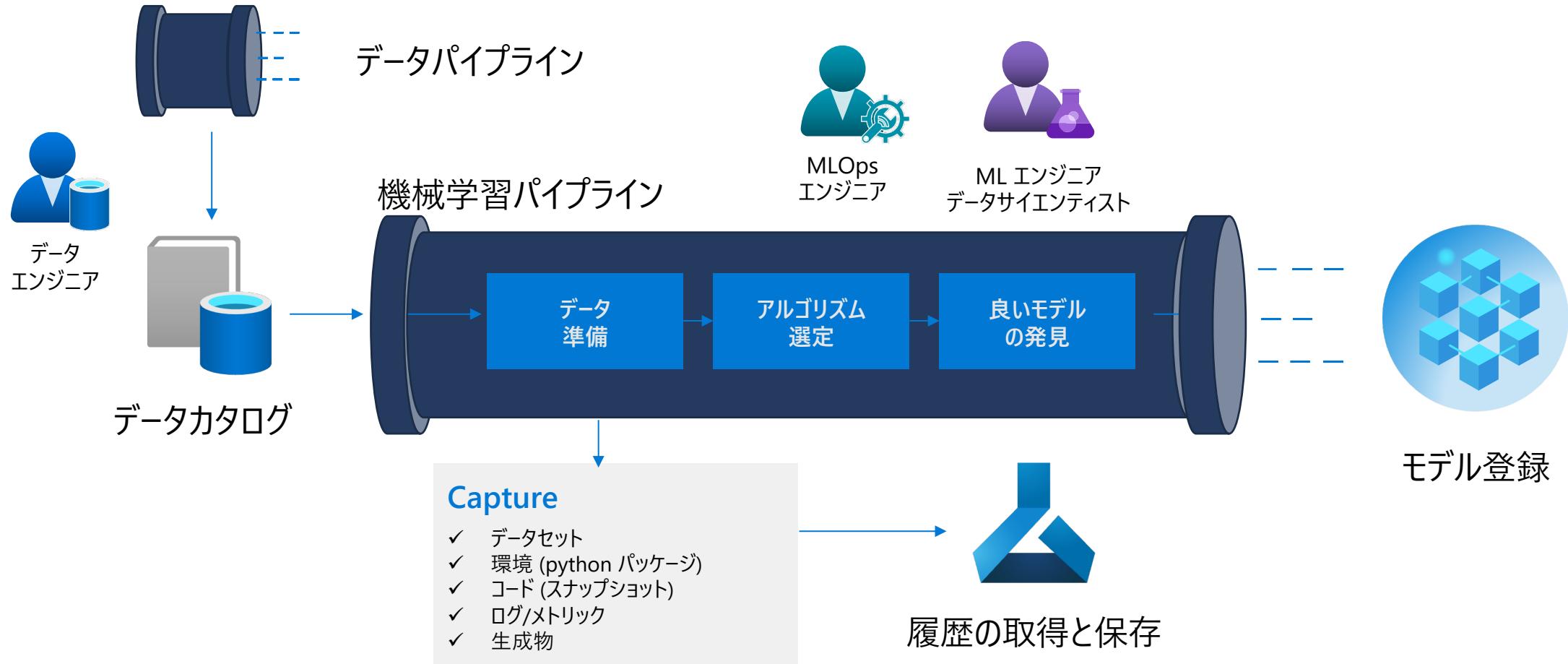
- ・ 再現性
 - ・ 学習を実行した際のアセット (データセット、ライブラリ情報、メトリック・パラメータ、モデルファイル etc) が組織で共有されていない。
 - ・ 確立したモデル学習やデプロイの手順がパイプラインとしてまとめられ実行可能な状態になっている

Lv1 → Lv2

- ・ モデル学習の再現性確保
 - ・ パイプラインを作成し、モデル学習のワークフローを自動化する。
 - ・ コードだけでなくデータ、メトリック、ログ、モデルなどのアセットを保存し、実験を紐づける。
- ・ モデルの運用管理
 - ・ 学習済みモデルを登録し、実験やエンドポイントと紐づける。

Level 2 – Automated Training

- ・コード、データ、モデルがバージョン管理され、実験が記録される
- ・モデルの再作成が自動化される



課題・チャレンジ

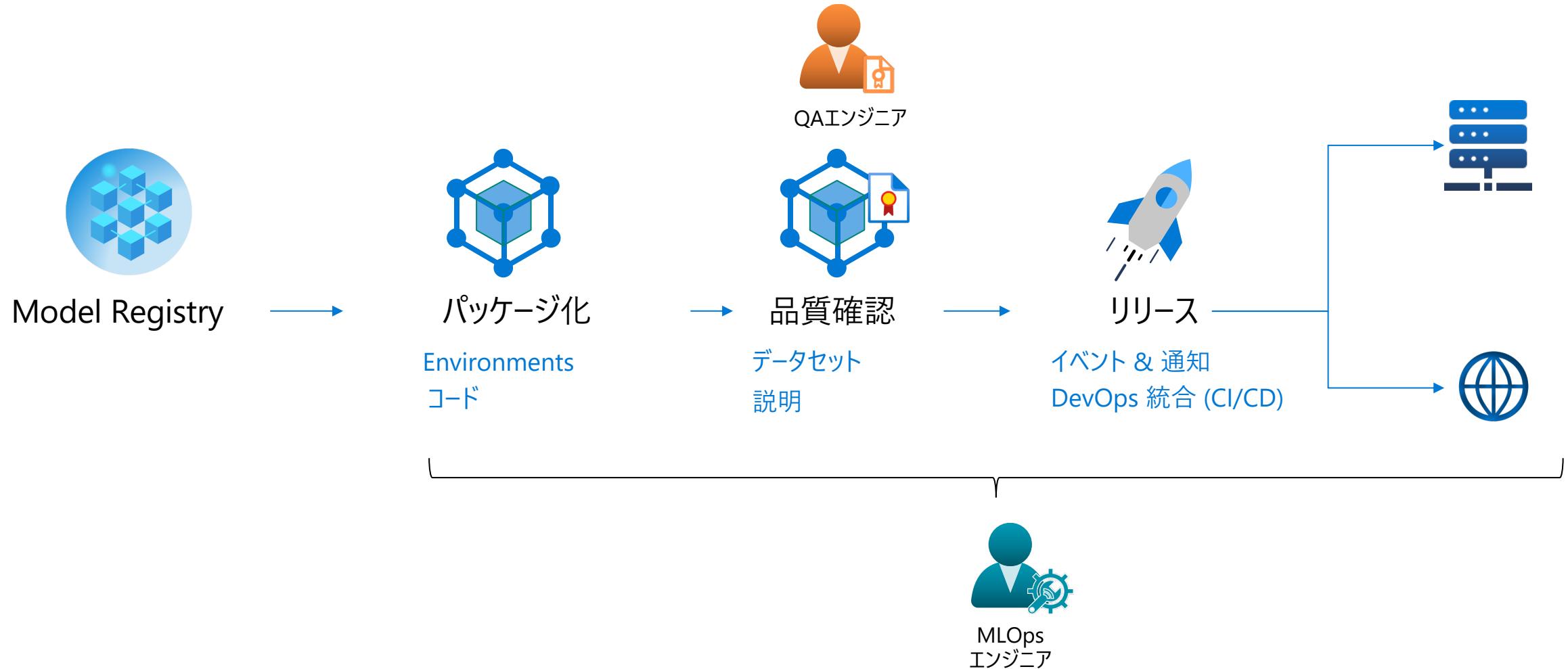
- ・ デプロイ
 - ・ モデルを推論エンドポイントにデプロイする手順が確立しておらず、時間がかかる。
- ・ モデル品質の担保
 - ・ モデルのテストが実施されていないため品質が悪い懸念がある。
 - ・ モデルがブラックボックスで説明が難しい。
 - ・ モデルがある属性に対して不公平な動きをする可能性があり、本番展開のリスクがある。

Lv2 → Lv3

- ・ デプロイパイプラインの実装
 - ・ モデルを推論エンドポイントにデプロイする自動化ワークフローを実装する。
- ・ モデルの品質向上
 - ・ テストデータによるモデル精度・挙動のテストを自動化する。
 - ・ 解釈可能性の高いモデル or 学習済みモデルに説明性を付与することによってモデルの透明性を高める。
 - ・ モデルの公平性を評価し、必要に応じて不公平性を軽減する。

Level 3 – Automated Model Deployment

- ・モデルのパッケージ化、品質確認、デプロイが半自動的に行われる



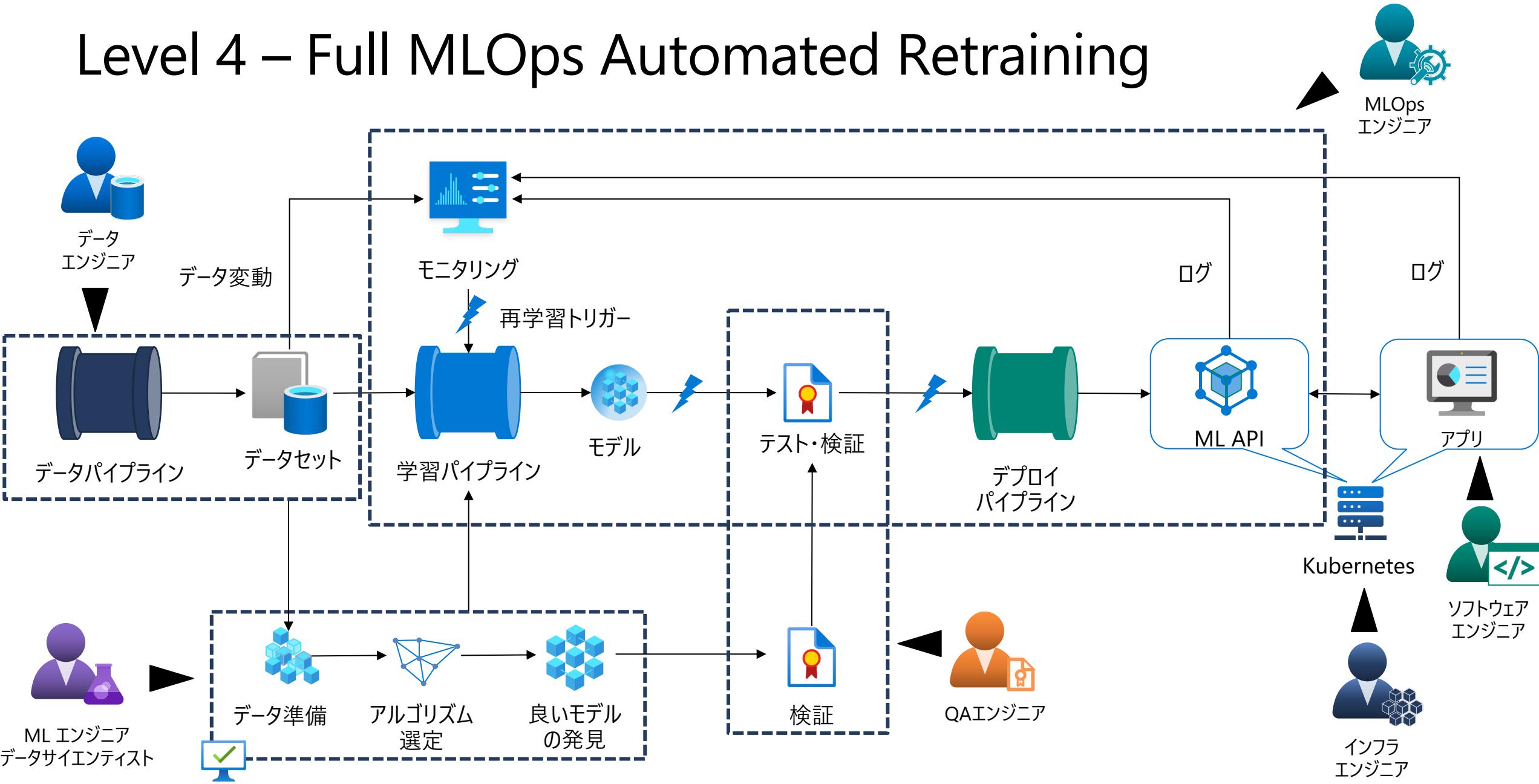
課題・チャレンジ

- ・ モデルの劣化
 - ・ デプロイ後にモデルがデプロイされないため、モデルのパフォーマンスが低下しユーザー・エクスペリエンスを悪化させている。
- ・ モデルの再学習
 - ・ 推論エンドポイントが十分にモニタリングされておらず、再学習を実施するタイミングが分からない。
 - ・ ユーザーへの影響を最小限度抑えながら、新しいモデルをデプロイするのが難しい。

Lv3 → Lv4

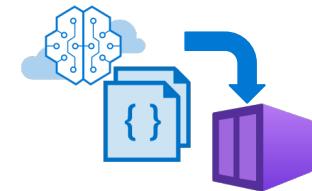
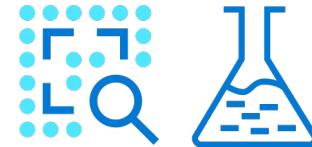
- ・ 推論エンドポイントのモニタリング
 - ・ データを定期的にスキャンしてデータドリフトを検知する。
 - ・ デプロイしたモデルやそのシステムに関するメトリックやログをモニタリングする。
- ・ 自動化
 - ・ メトリックやログのモニタリング結果に基づいてモデルを再学習するパイプラインをトリガーする。
 - ・ blue-green デプロイメントにより、新しいモデルを安全に本番環境にデプロイする。

Level 4 – Full MLOps Automated Retraining



Azure Machine Learning における MLOps

Azure Machine Learning の基本的な使い方



従来

- GPU 搭載のサーバーを調達
- 電源確保 & ネットワークに接続
- Python 環境と Jupyter をセットアップ
- GPU のドライバ/ライブラリをインストール
- データを吸出して CSV や Parquet 等のファイル形式で貯う
- データの前処理
- アルゴリズムの試行とパラメーター探索
- Dockerfile を書いてモデルを収めたコンテナを作成
- Flask / fastAPI などで HTTP エンドポイントをセット
- Kubernetes 等にコンテナをデプロイ

計算リソースの用意

環境構築

データの吸出し

前処理/モデルの作成

推論コンテナの作成

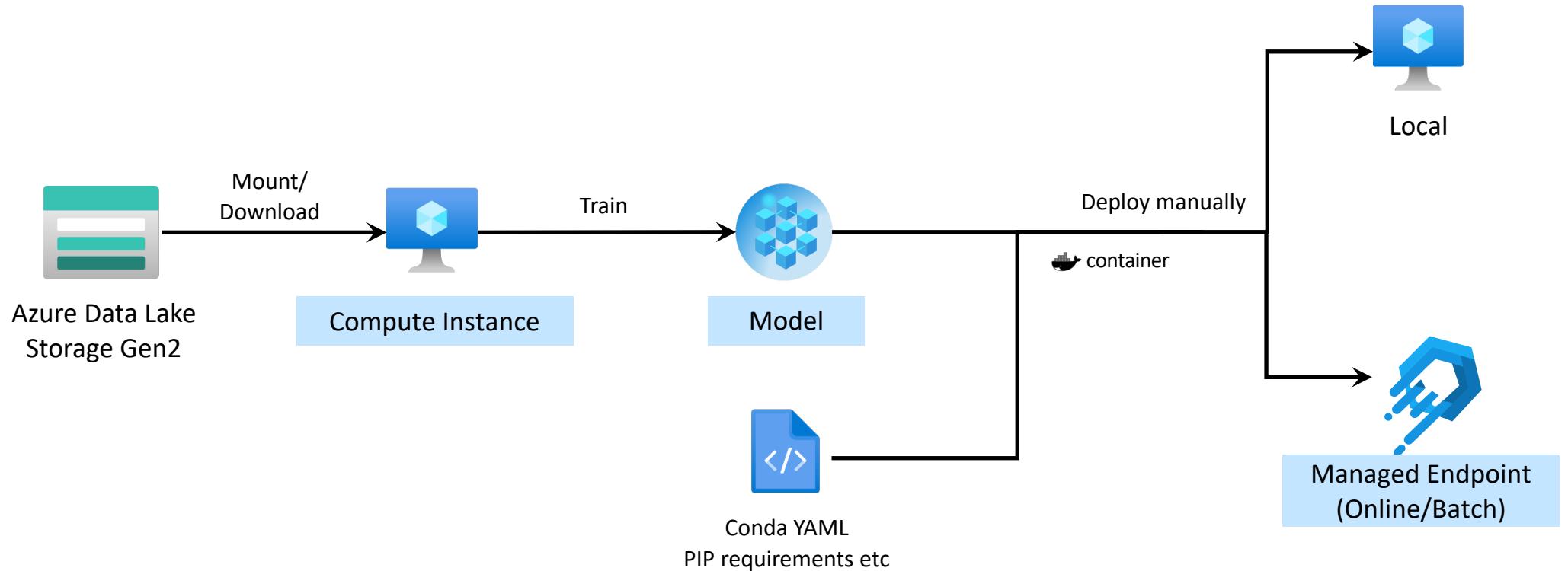
デプロイ



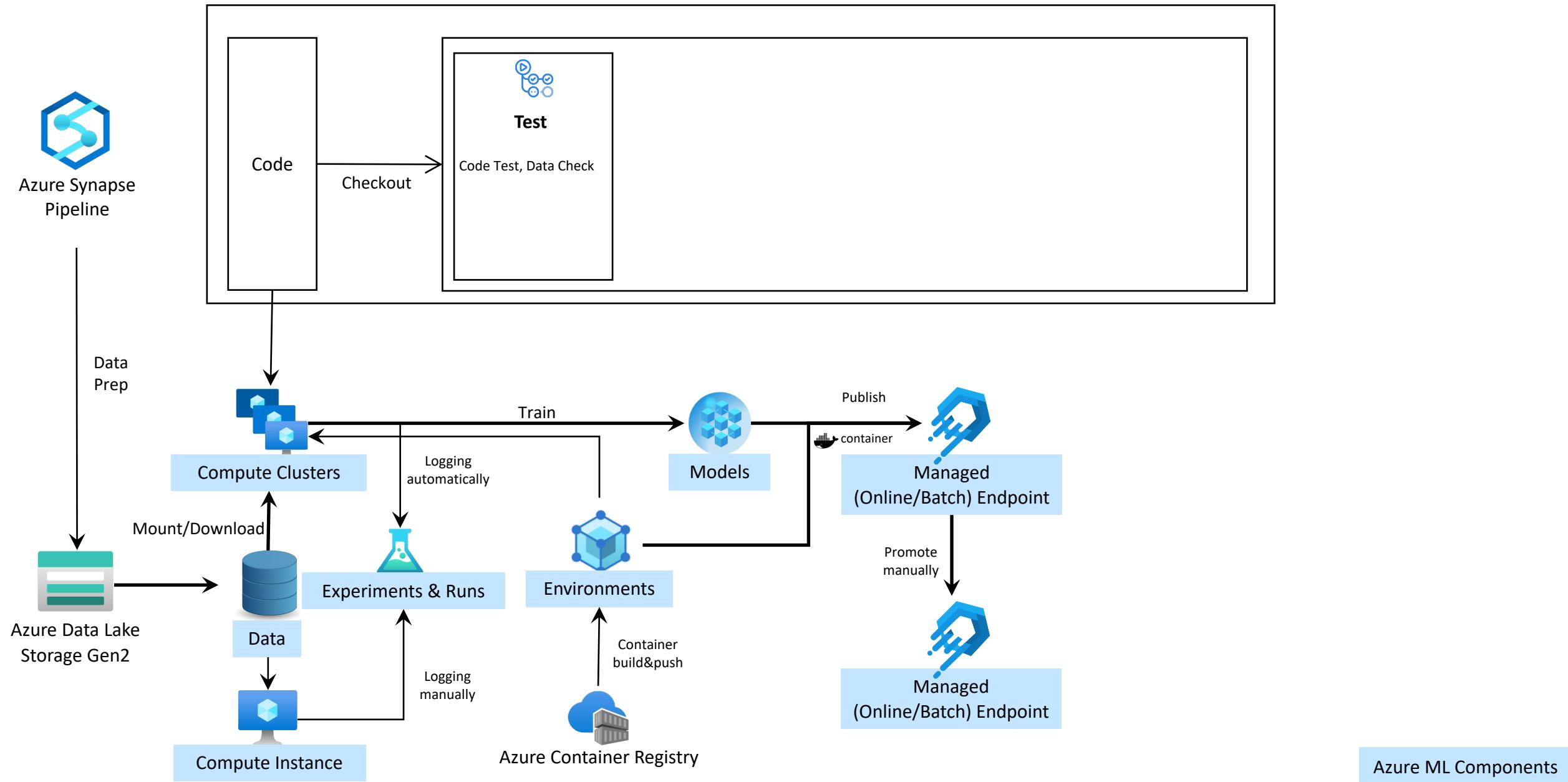
Azure
Machine
Learning

- 計算リソースをアタッチ
- Azure ML Compute Instance/Cluster を展開
- Environment を読み込む
- Compute Instance をそのまま使う
- Datastores としてデータソースをアタッチして、Dataset を作成
- Auto ML でモデルを自動生成
- 大規模計算リソースで効率的に探索
- 実験/モデルを記録
- 保存したモデルにメタデータを付与する
- エントリスクリプトの作成
- アタッチした推論用のリソースにデプロイ

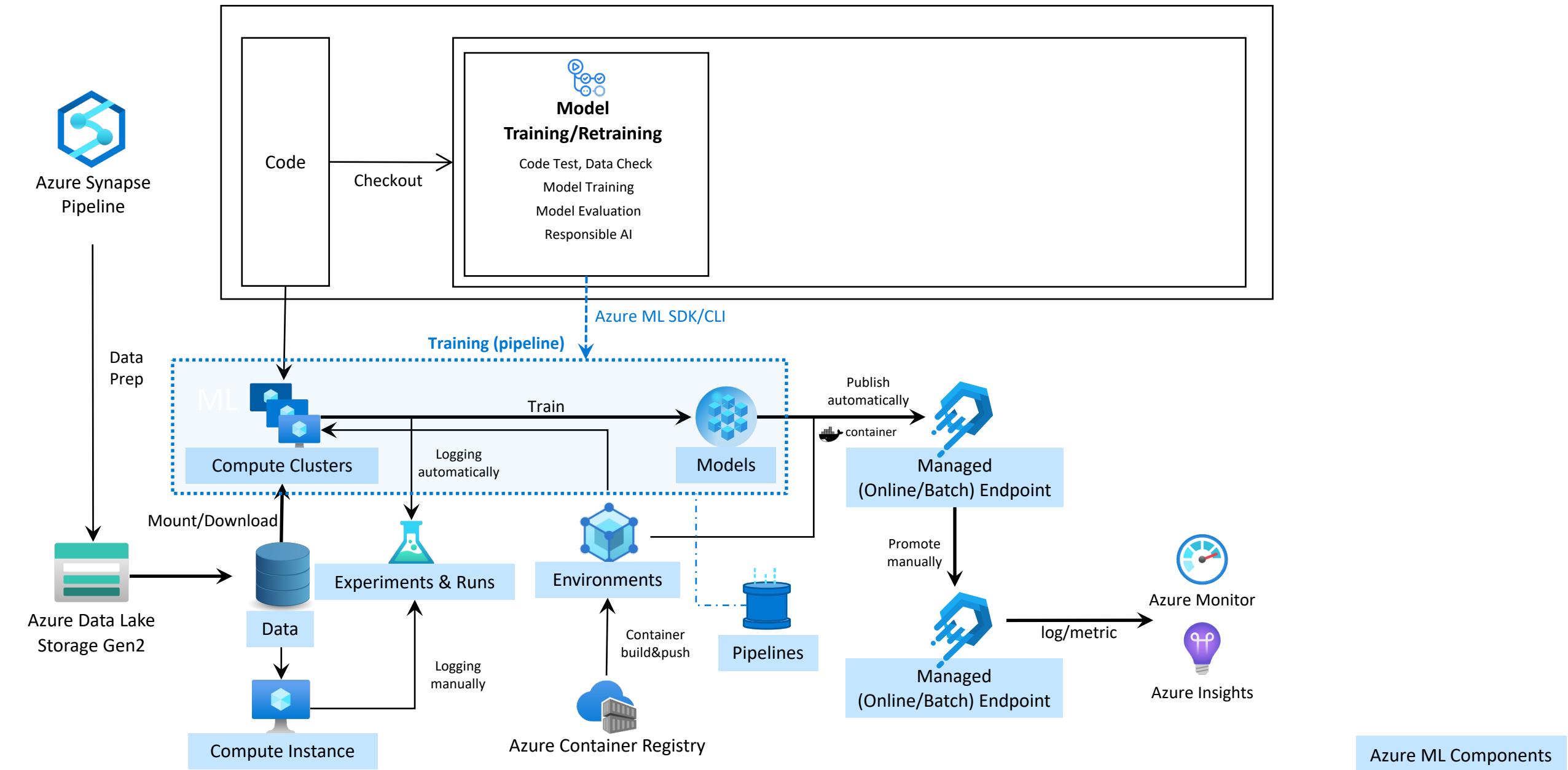
Level 0 – No MLOps



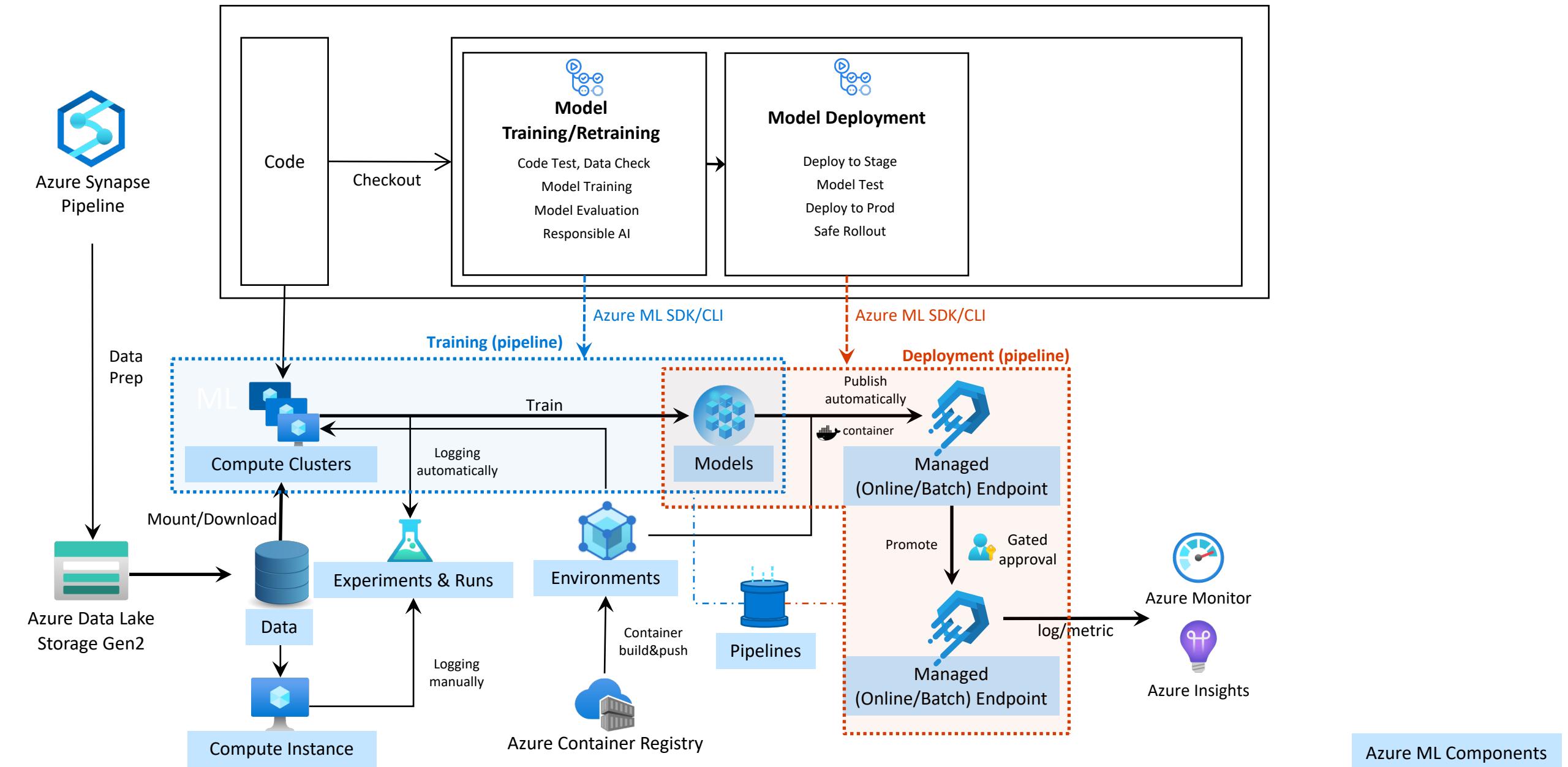
Level 1 – DevOps no MLOps



Level 2 – Automated Training



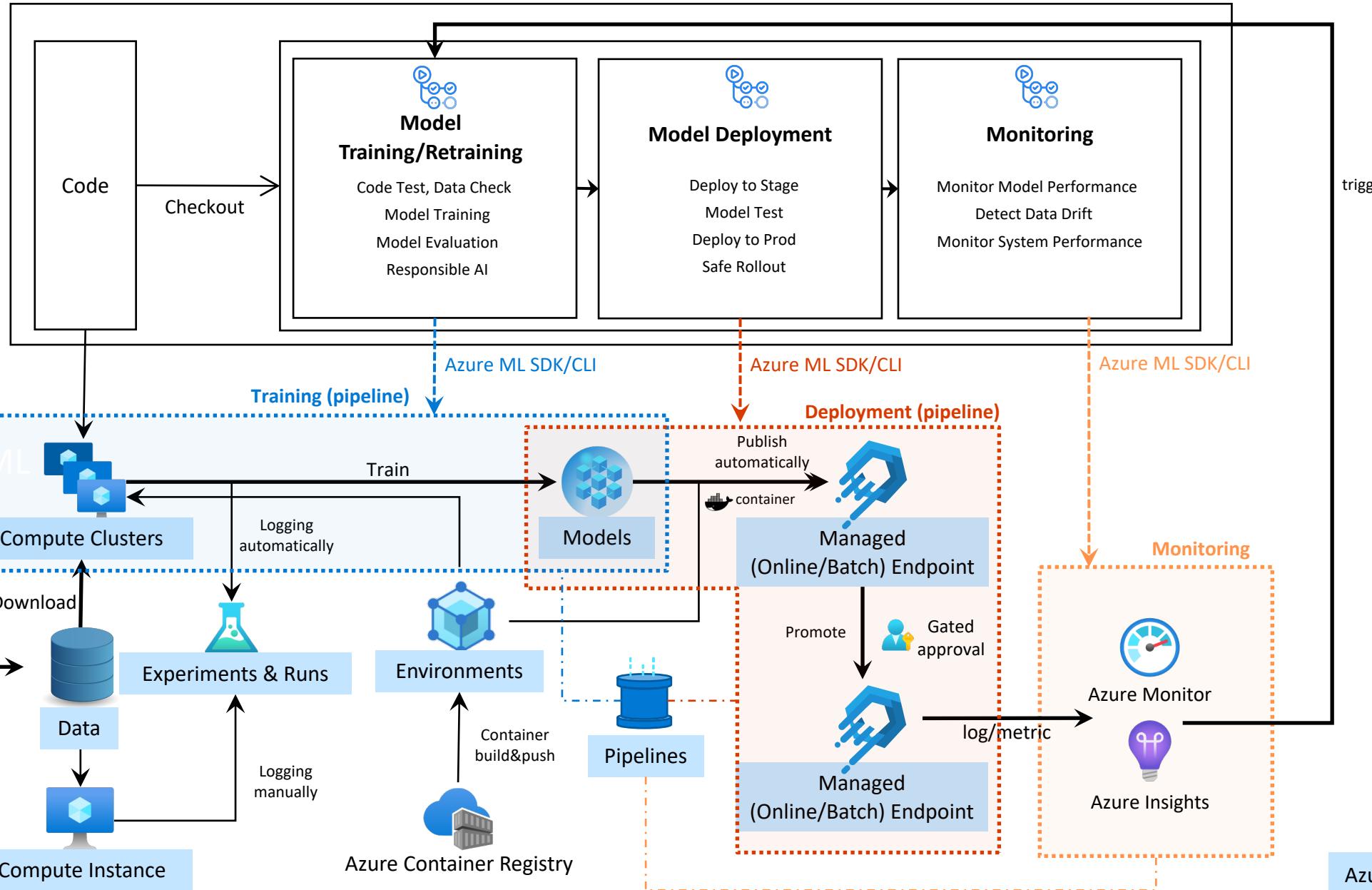
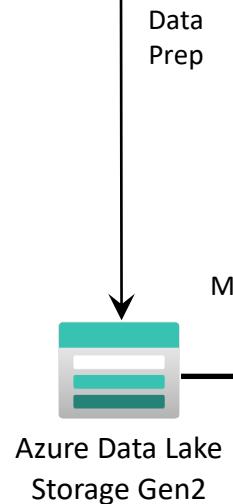
Level 3 – Automated Model Deployment



Level 4 – Full MLOps Automated Retraining



Azure Synapse
Pipeline



trigger

Azure Container Registry

Azure ML Components

参考資料

Category	Link
Microsoft Docs	機械学習とは - Cloud Adoption Framework Microsoft Docs
Microsoft Docs	Azure Machine Learning を使用して機械学習のライフサイクルをアップスケールするための機械学習運用 (MLOps) フレームワーク - Azure Architecture Center Microsoft Docs
Microsoft Docs	Team Data Science Process とは - Azure Architecture Center Microsoft Docs
Github Repository	microsoft/MLOps: MLOps examples (github.com)
Github Repository	Azure/mlops-v2: Azure MLOps (v2) solution accelerators. (github.com)
Microsoft Docs	Azure Machine Learning を使用して機械学習のライフサイクルをアップスケールするための機械学習運用 (MLOps) フレームワーク - Azure Architecture Center Microsoft Docs



Microsoft AI



FastTrack for Azure とは?

FastTrack for Azure (FTA) は、クラウド プロジェクトを抱える組織向けに、Azure ソリューションの迅速かつ効果的な設計とクラウド上での展開を可能にする技術支援プログラムです。ベスト プラクティスとアーキテクチャ ガイドを活用して、Azure エンジニアにより カスタマイズされたガイダンスが提供されます。



FastTrack for Azure 概要

お客様への利点



Azure エンジニアから直接支援を受ける



Azure のデプロイを成功させる方法を学ぶ



Azure ソリューションの展開を迅速化する

FastTrack for Azure が提供するもの

ディスカバリー

ビジネス ビジョンと目標を定義し、アーキテクチャのニーズを評価する

ソリューションの有効化

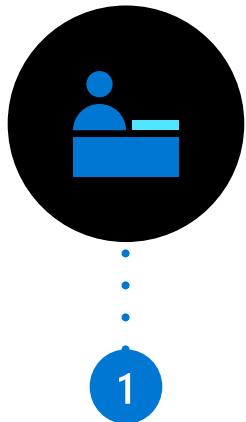
ビジネス アプリケーションに適した実証済みのプラクティス、設計原則、およびツールを提供する

展開

運用環境に Azure ソリューションを正常にデプロイするためにお客様の社内リソースおよびパートナーとのコラボレーションする

FastTrack for Azure プログラムの使用条件

プログラムに参加するには、以下の条件を満たす必要があります



1
Azure サブスクリプションを
持っている



2
サポートされているリージョンに
プロジェクトを実装する

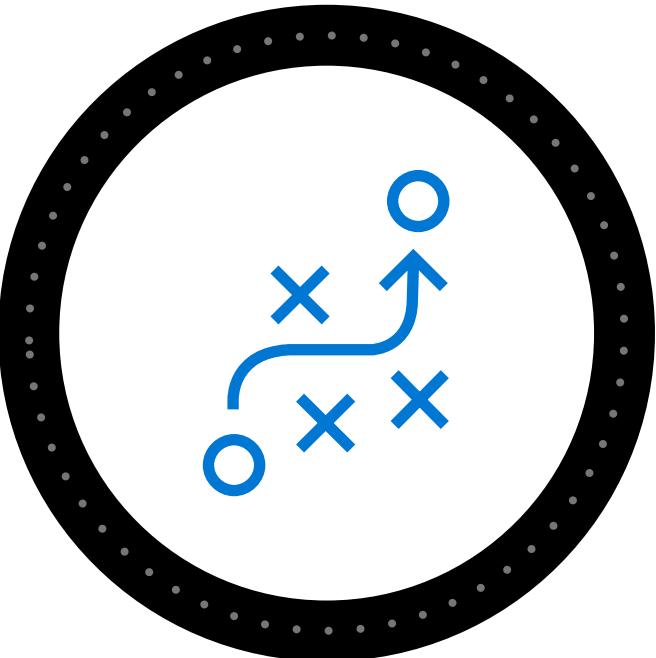


3
FastTrack for Azure でサポート
されるソリューションを実装する
プロジェクトが特定されている



4
特定されたプロジェクトは、Azure
に展開後 12 か月以内に \$5k/月
を消費する可能性がある

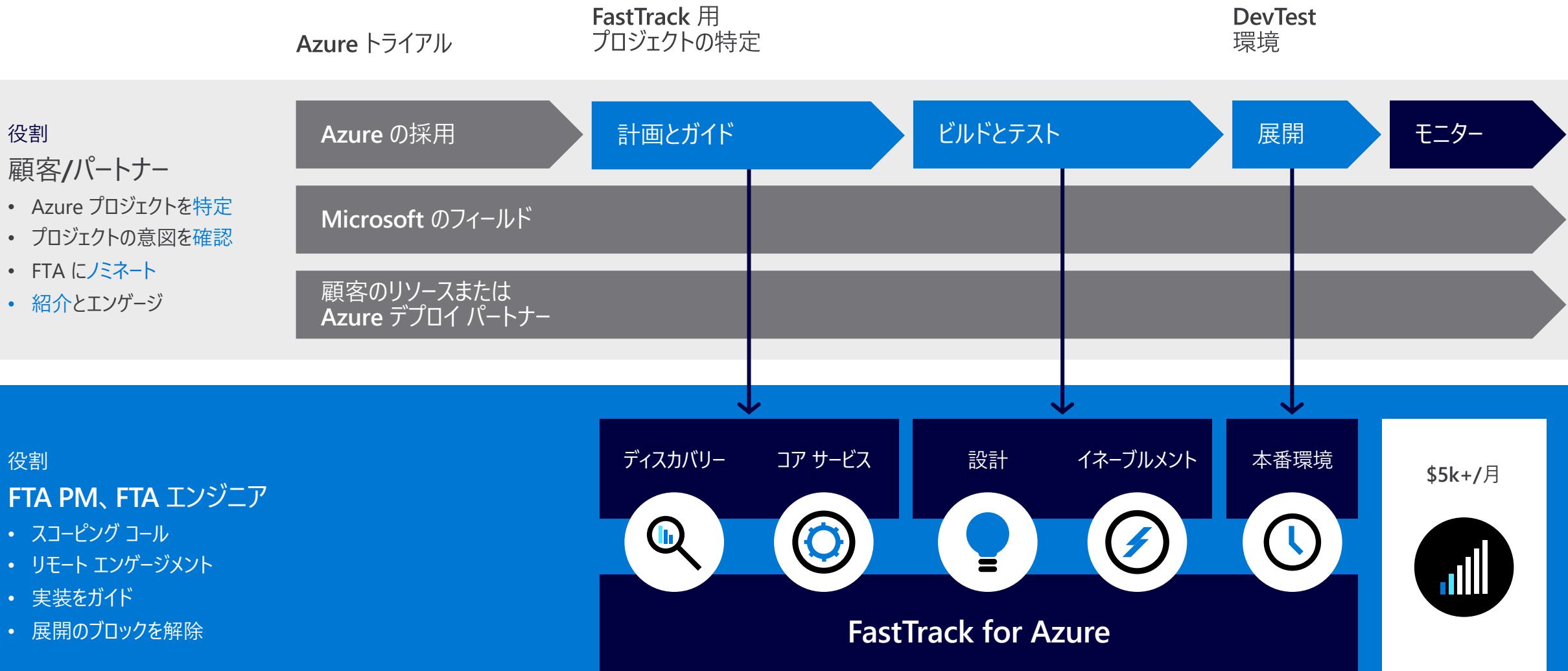
FastTrack for Azure でサポートされるソリューション



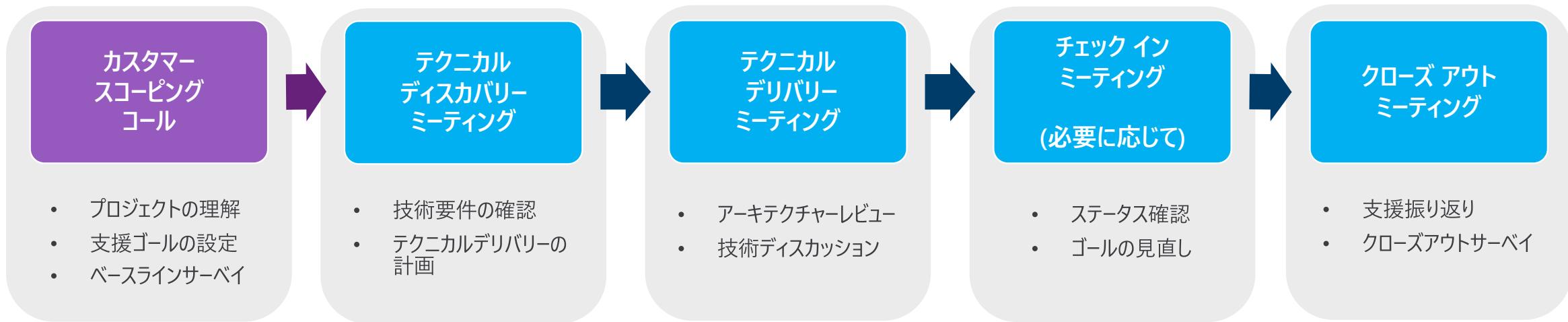
注: HPC と IoT は、現在日本語では提供されていません。

- AI/ML
- AppDev
- ビジネス継続性と災害復旧 (BCDR)
- データ プラットフォーム
- データセンターの移行
- ハイパフォーマンス コンピューティング (HPC)
- SAP
- AVD
- セキュリティ

FastTrack for Azure で支援できるタイミング



FastTrack for Azure のエンゲージメント フロー



FTA PM

FTA
エンジニア

お客様

パートナー

※ 支援はすべてリモートで実施します

※ お客様のみ、またはパートナー様のみへのご支援も可能です

FastTrack for Azure のスコープについて

スコープ内

- ソリューションおよびアーキテクチャ設計のレビュー
- 実証済みの実践技術アドバイザリーと運用ガイダンス (モニター系)
- 一般公開されているドキュメントから引き出されたサンプル/テンプレート/コード/スクリプトを使った技術知識トランスファー
- 本番環境を目指すソリューションの PoC および開発/テスト環境の展開支援
- Microsoft Azure 製品グループへのフィードバック

スコープ外

- ✗ 構想段階の、Azure プロジェクトの特定のための支援
- ✗ ライセンスに関するガイダンス、提案、支援
- ✗ Azure 環境および構成に関わるコストの見積もり
- ✗ お客様向けのコード/スクリプトの開発
- ✗ アーキテクチャドキュメント/成果物の作成
- ✗ アセスメント結果 (生データ) またはログ (パフォーマンス ログも含む) の分析
- ✗ お客様の Azure サブスクリプションの有効化、アクセスおよび操作
- ✗ トラブルシューティング支援
- ✗ 非公開情報および Azure 以外のサービスに関するガイダンスの提供

FTA プログラムに関するよくあるご質問

Q: 問合せにはどのくらいの時間でご回答をいただけるのでしょうか？

A: FTA 支援では SLA を設けておりません。お客様およびパートナー様からのお問い合わせに関するマイクロソフトからの回答は全てベストエフォートになります。

Q: FTA テクニカルセッションでのレビュー結果やアドバイス結果は、全て採用する必要があるのでしょうか？

A: FTA チームとディスカッションした結果を最終的にソリューション アーキテクチャーに採用するかについては、お客様およびパートナー様にてご判断をお願い致します。

Q: テクニカルセッションの議事録を作成していただけるのでしょうか？

A: テクニカルセッションの議事録作成もお受けできかねますので、お客様およびパートナー様にてご対応をお願い致します。FTA チームとのテクニカル テクニカルセッション終了後、FTA エンジニアがフォローアップ メールを送ることがありますが、その内容はお客様とパートナー様の利便性を考慮した ミーティング メモであり、ディスカッションした内容の正式な議事録およびレポートではありません。

Q: テクニカルセッションは録音可能でしょうか？

A: FTA で開催するミーティングの録音・録画はお断りさせて頂いております。外部ツール・機器による録音・録画やまたライブ中継に関しましてもお断りさせて頂いております。

Q: FTA 支援による納品物はあるのでしょうか？

A: FTA はドキュメントやスクリプトの作成をスコープ外としているプログラムであることから、支援による納品物はございません。

New!

AI/ML 関連のソリューション

ソリューション名	プロダクト	概要	備考
Platform/Security : Modern AI Platform	Azure Machine Learning, Azure Cognitive Services, Azure Cognitive Search, ADF, Synapse Analytics, Power BI, ADLS Gen2	Azure Machine Learning などの Azure AI サービスを用いたセキュアな AI・機械学習プラットフォームのアーキテクチャの設計・実装方法のレクチャー、設計レビュー	※既存環境・ツールからのマイグレーションも含む
MLOps : Asset Management	Azure Machine Learning, GitHub, Azure DevOps	アセットをチームで運用管理するための仕組みづくりの設計・実装方法のレクチャー、設計レビュー	
MLOps : Process Automation	Azure Machine Learning, GitHub, Azure DevOps	パイプライン(モデル学習、デプロイ、監視)を作成し、機械学習プロセスを自動化する設計・実装方法のレクチャー、設計レビュー	
Acceleration : AI at Scale	Azure Machine Learning	大規模データに対するモデル学習、分散学習、ハイパーパラメータチューニングを高速実行するためのアーキテクチャの設計・実装方法のレクチャー、設計レビュー	
Acceleration : Productivity	Azure Machine Learning, Azure Container Registry	VSCode などの IDE 上で PyTorch, Scikit Learn などの OSS フレームワークを用いた開発環境の設計・整備方法のレクチャー、設計レビュー	※ IDE は VSCode, Jupyter, JupyterLab, RStudio の想定
Responsible AI : Model Debug	Azure Machine Learning	機械学習モデルへの説明性の付与、誤差分析の実装方法のレクチャー	