pactera EDGE

AZURE MLOPS:
DEVOPS FOR MACHINE LEARNING

SOLUTION OVERVIEW:

Design, build, and deploy an end-to-end future-ready Azure MLOps platform that allows for collaboration between both IT Engineers and Data Scientist.

This Solution is also an operational framework that abstracts the underlying infrastructure complexities allowing the ML engineers to focus on what they do best, thus accelerating the ML driven enterprise capabilities that drive measurable business value.

SOLUTION SUMMARY:

- Azure ML is the combination of Azure DevOps and Azure ML.
- MLOps is **DevOps** for building and commercializing Machine Learning Models.
- It embodies a shift from an **"ARTISAN-LIKE"** approach to model building and deployment, to an **"Institutionalized"** way of infusing Intelligence into the Enterprise.

ARCHITECTURE FLOW FOR MLOPS

An End-to-End MLOps pipeline on Azure can be architected using set ML Build and ML Release Pipelines

ML Build Pipeline

ML Release Pipeline

TYPICAL CHALLENGES WITH SCALING ML AT ENTERPRISE SCALE

People

- Data scientists lack ML engineering and DevOps skills requiring overreliance on IT Support
- IT Team lacks capacity to meet the demand given that the current process is inefficient
- IT team is overburdened by operationalization needs and unable to focus on maturing the organization

Process

- Asynchronous model building process that is not scalable at enterprise level
- Disjointed process leading to hurdles towards collaboration between data scientists and engineers
- Technical and operational challenges towards model management and deployment

Technology

- Heterogenous sets of tools and solutions for model building across different ML Teams
- Disparate model visualization solutions with hurdles in scaling to enterprise capabilities
- Current tech stack lacks sophistication required for a future-ready AI/ML enabled organization

HOW WE CAN HELP

Architect, build, and operationalize MLOps pipeline on Azure Cloud	Productionalize models built at a rapid pace
Productionalize additional models from different functional groups on the MLOps pipeline	Develop and deploy the MLOps operational framework and provide training to citizen data scientists and ML engineers
Provide ongoing MLOps engineering support	Partner with Enterprise AI/ML teams to co-create new enterprise capabilities

KEY SUCCESS FACTORS

Collaborative Development

- Reduction of rework for IT Engineers in productionalizing models
- Reduce the burden on data scientists for enterprise scaling of value driving models
- Data scientists focused on experimentation while IT Team focused on engineering solutions for end users
- Reliance on IT should decrease wherein the Data Scientists become self-sufficient in MLOps over time

Accelerate Enterprise Capabilities

- Speed of model development, deployment, and business adoption must increase
- Enterprise scaling of proven models from all functional groups be seamless and fast tracked
- The AI/ML teams should be able to rapidly prototype and scale AI/ML enterprise solutions with minimum reliance on IT support
- Operationalize CI/CD and ML DevOps functionality across the enterprise
- Cohesive model consumption solutions with UI/UX designed to drive adoption as needed

Efficient AI/ML Product Portfolio Management

- Model management and deployment is critical to the success
- Governance and Monitoring of the models become streamlined and intuitive
- Triage of incidents (data, model output, front end, back end) to be managed by the IT Team with ease allowing the data scientists to focus on model building and driving adoption

TYPICAL ENVIRONMENT BEFORE MLOPS?

- AI/ML Engineers spend considerable time on data preparation and feature engineering to build the model
- Seeking of help from IT team to train and deploy models every time
  - Data Scientist losing time in coordination and un-productive work
  - Over-burdened IT team supporting ad hoc requests from Data Scientists for compute
  - Resources like GPU's for Experimentation and Model Training
  - Overall increased cost of operations due to delays, lot of manual and unproductive work
- Lack of mapping between feature sets and the many sandbox trial runs
- Lack of logging, evaluation metrics and other relevant meta data needed for reporting
- Missing Governance and cadence metrics around model performance monitoring, feature set ingestion
- mapped to individual deployment commits by the data scientists
- Difficult to plan the re-training of models on pertinent data sets at regular cadence
- Standing up the infrastructure needs like GPU's for model retraining requires extensive support from IT team
- Data scientists need Application Developers to expose the models as service for Enterprise wide consumption
- Lack of ability to scale the use of Machine Learning and infusion of intelligence into the Enterprise, resulting in loss of business opportunities

TYPICAL ENVIRONMENT AFTER MLOPS

- **AI/ML Engineers now spend their valuable time on data preparation and feature engineering to build the model**

EVERYTHING ELSE IS TAKEN CARE OF AS AN INSTITUTIONAL STANDARD:

- **Versioning Model Code on Git Hub** for easy collaboration between ML Engineers and ML Ops Engineers
- **Data Sanity** check routines
- **Established DevOps process** that gets triggered for changes to model code
- **Reusable assets** such as “Model Training and Inference Environments thus saving time
- **Configure and consume** high performing compute environments without waiting on IT teams
- **Easy setup** of “Dynamically Scalable Infrastructure” to address high demand
- **Automated Services:**
  - **Model** training, validation and registration of the Models
  - **Creation of wrapper REST API** Service for the published Models for Enterprise Consumption
  - **Monitoring** of the Model’s performance
  - **Data Drift** Analysis
  - **Security** through Multifactor Authentication (MFA)

OUR RAPID IMPLEMENTATION MLOPS SOLUTION AT A GLANCE

Attribute	Description
Service Offering	**MLOps pipeline rapid implementation tool for Microsoft Azure Cloud**
Description	This solution is aimed at serving as a comprehensive framework for rapid implementation of an enterprise scale MLOps pipeline using assets on Azure cloud. MLOps pipeline is brought to life through a combination of Azure DevOps and Azure ML.
Enterprise Framework	Ensures an elegant hand-off between data engineering and the ML Team. Supports all steps for cradling the model from development to production passing through the following steps: ▪ Data Ingestion ▪ Security Checks ▪ Feature Engineering ▪ Model Training ▪ Model Registering ▪ Model Scoring ▪ Model Re-training
Environment	Azure Cloud- A Combination of Azure DevOps and Azure ML
License Type	Basic or Pay-As-You-Go Azure Subscription
Data Storage	Azure Blob Store,
Security	Active Directory Authentication , Service Principal Authentication, RBAC and NSG Groups
Model Deployment Environment	Azure Kubernetes Cluster exposed as REST API service inside the enterprise VNET (behind a secured firewall)

SKILL PRE-REQUISITES

- Expertise with Azure Cloud Environment.
- Expertise with CI/CD using Azure DevOps.
- Familiarity on Azure Machine Learning.
- Familiarity with Azure MLOps.

HIGH LEVEL FLOW FOR MLOPS ON AZURE CLOUD

Resource Creation:
- All resources that are used for MLOps in Azure can be created using ARM templates in IaC Pipelines

CI Pipeline:
- Create repository on the Azure project to host the source code. Enable pipelines to perform CI/CD operations
- Create CI pipelines to perform code quality tests and unit testing on every commit
- Create tasks in CI pipeline to build, publish and trigger ML Pipeline to Azure ML and CI pipeline would wait until the tasks have finished their job

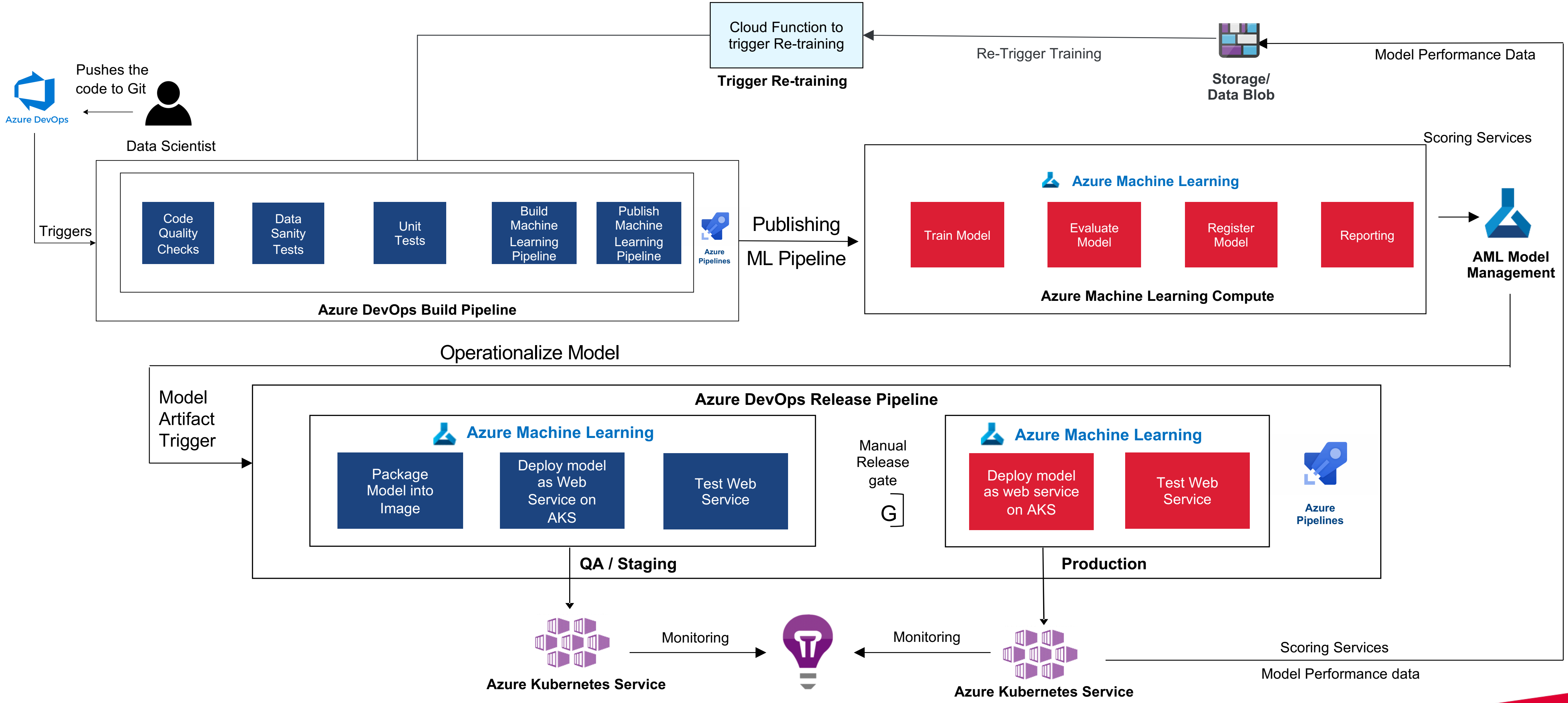
ML Pipeline:
- After the ML pipeline is published and triggered on Azure ML workspace a trained model is given as the output
- Perform model evaluation with existing model in production and register if evaluation succeeds
- As soon as a model is registered the CD Pipeline would be triggered automatically

CD Pipeline:
- The CD pipeline would be configured to pick the latest model from the model registry, package it as a service. Model registry is a version control system for the built models. Any selected previous version of the model can be deployed if desired
- Deploy the packaged model as Webservice onto desired environment (AKS cluster) which provides the REST API Endpoint for consumption of the model

Monitoring:
- Azure App Insights and Azure Monitor Services help us generate and monitor custom logs along with viewing live data in analytics of the deployed cluster and service.
- Azure ML workspace has all the historical runs which provides the complete meta information on items like the data set that has been used for that run, what it’s accuracy is, and if the model was deployed, what the features & significance were, etc. Model evolution is not just confined to code change versions in the version control system

Re-Training:
- Azure Functions are created to trigger retraining of pipelines based on various conditions such as data drift or scheduled time intervals

TYPICAL END-TO-END MLOPS ARCHITECTURE



pactera EDGE