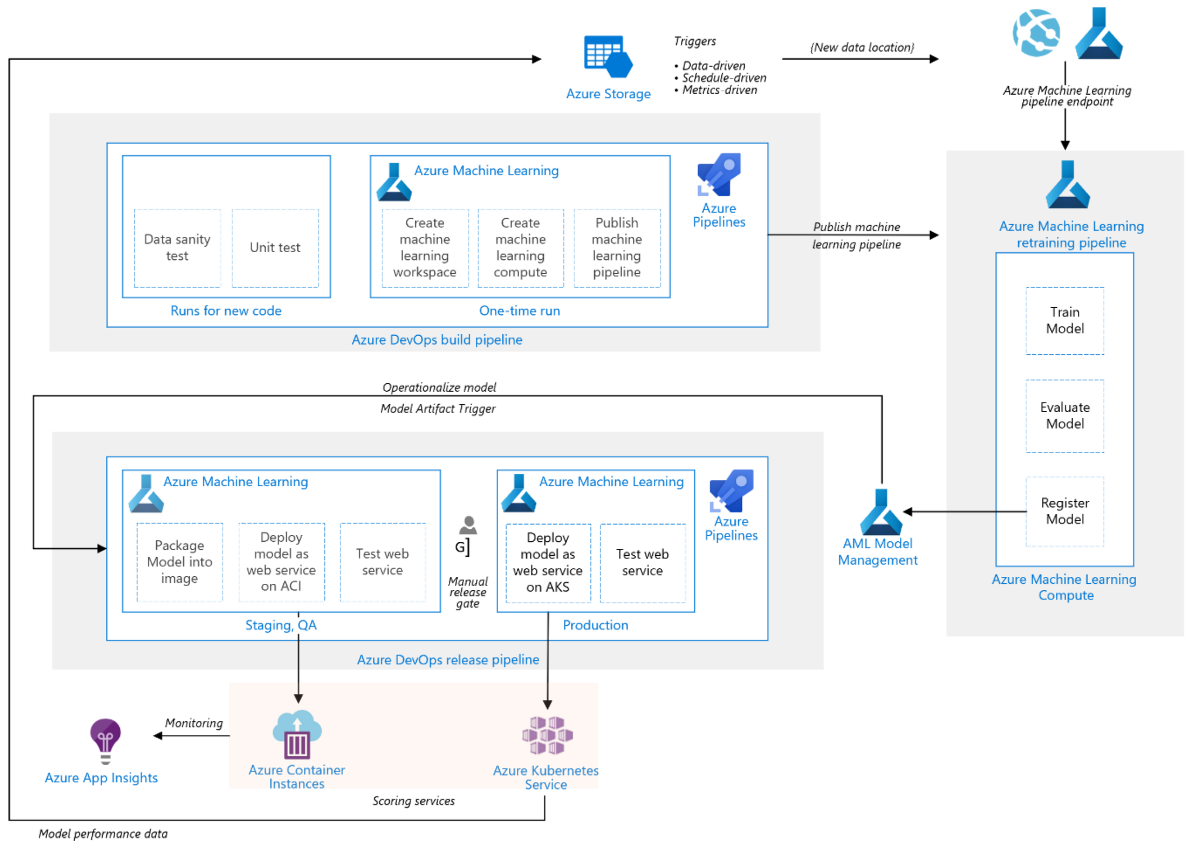
Introduction:

This document explains the necessary steps to deploy a machine learning application in a cloud. For scope limitations, we will be using Azure cloud and AHClassifier application.

Architecture:



The architecture using Azure Machine Learning workspace along with Azure Devops for CI, CD, monitoring and deployment of model as a web service. Each component is stated below:

Azure Devops: This is used for creating Build and Release pipelines

Azure Machine Learning: Used for accessing data for the model, registering the model and triggering pipelines for e.g. model training pipeline.

Azure Container Instance: This can be used for deploying the containers containing the model. The container can be deployed on Azure ML as a webservice for real time scoring

Azure Kubernetes Service: This can be used for container orchestration and also for running GPU for the model.

The following items list down the steps needed to deploy this code on Azure cloud:

1) In order to deploy the AHClassifier application , we need:
   - An Azure subscription
   - An Azure devops organization and a project
   Make sure these are present and the devops project has access as owner or contributor to provision resource in the subscription. To connect the azure devops project with azure portal, create a virtual machine and install a Microsoft agent on it. Once done, add a service connection in the azure devops project to connect.

2) This step involves creation of resources. This is a one time infrastructure activity and can be done in 2 ways
   - Manually: Create a resource group in your subscription and create an Azure ML workspace, a keyvault, storage account, azure container registry and a vnet.
   -Automatic: A YAML pipeline can be created using the AHClassifier\env_setup\iac-create-environment-pipeline-arm.yml pipeline. Login to azure devops project and create a new branch. Clone the AHClassifier code in the branch and create a build pipeline using the YAML file. Once created, run the pipeline to deploy the resources.
3) Once the resources are created, we need to build an artifact of our code. This is done so to make sure that there are no bugs in our code before we deploy the code to an environment. For this, we can create a CI (continuous integration) pipeline. This can be done in 2 ways:
   - Manually: Create a pipeline with initialization of a docker file, a python lint task, running unit test, publish coverage report and publishing the AHClassifier\src\training\train.py code into a training pipeline in Azure ML.
   -Automatic: Create a model_CI pipeline using the AHClassifier\pipelines\AHImgClassifier-ci.yml pipeline.
4) Once the code has passes the unit tests and the train.py file has been added in Azure ML studio as a training pipeline, we can now run the evaluation of the model in Azure ML using the src/evaluate/evaluate_model.py. Once the model has been evaluated and registered, we can move the model on higher environments. This is done automatically when a new model version is available in Azure ML
5) On the higher Test, Acceptance environments, we can deploy our model in a container image via a Azure Devops. This will be the AHClassifier\src\model\pickle folder. Since the requirement is to deploy the model as a webservice, we can either use the App service or ACI for an endpoint so that real time scoring can be done.
6) Once deployment is done, we can either optionally using App insights to monitor the Kubernetes/Azure Container instances.

To summarize, the deployment steps are:

- Create Azure resources
- Build the code (CI)
- Release code (CD) with model version
- Optionally save scoring metrics to a Datalake/Blob storage.