

MLOps with Azure AutoML And Azure DevOps



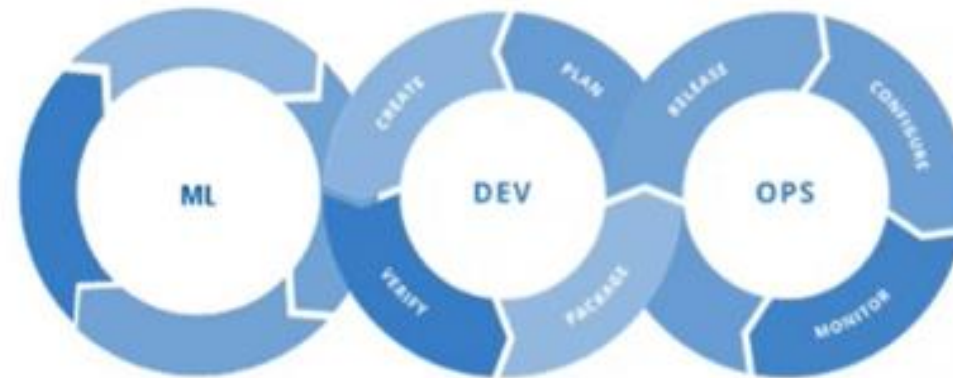
Technology we use to run this project

Technology	Objective/Reason
Azure DevOps	The platform to help you implement DevOps practices on our scenario
Azure Machine Learning Service (Automated ML)	Manage Machine Learning models with the power of Azure
Azure Container Instance(ACI) and Azure Kubernetes Service(AKS)	Deploy Machine Learning models as Docker containers

What is MLOps?

MLOps (a compound of Machine Learning and “information technology OPerationS”) is new discipline/focus/practice for collaboration and communication between data scientists and information technology (IT) professionals while automating and productizing machine learning algorithms.

MLOps = ML + DEV + OPS



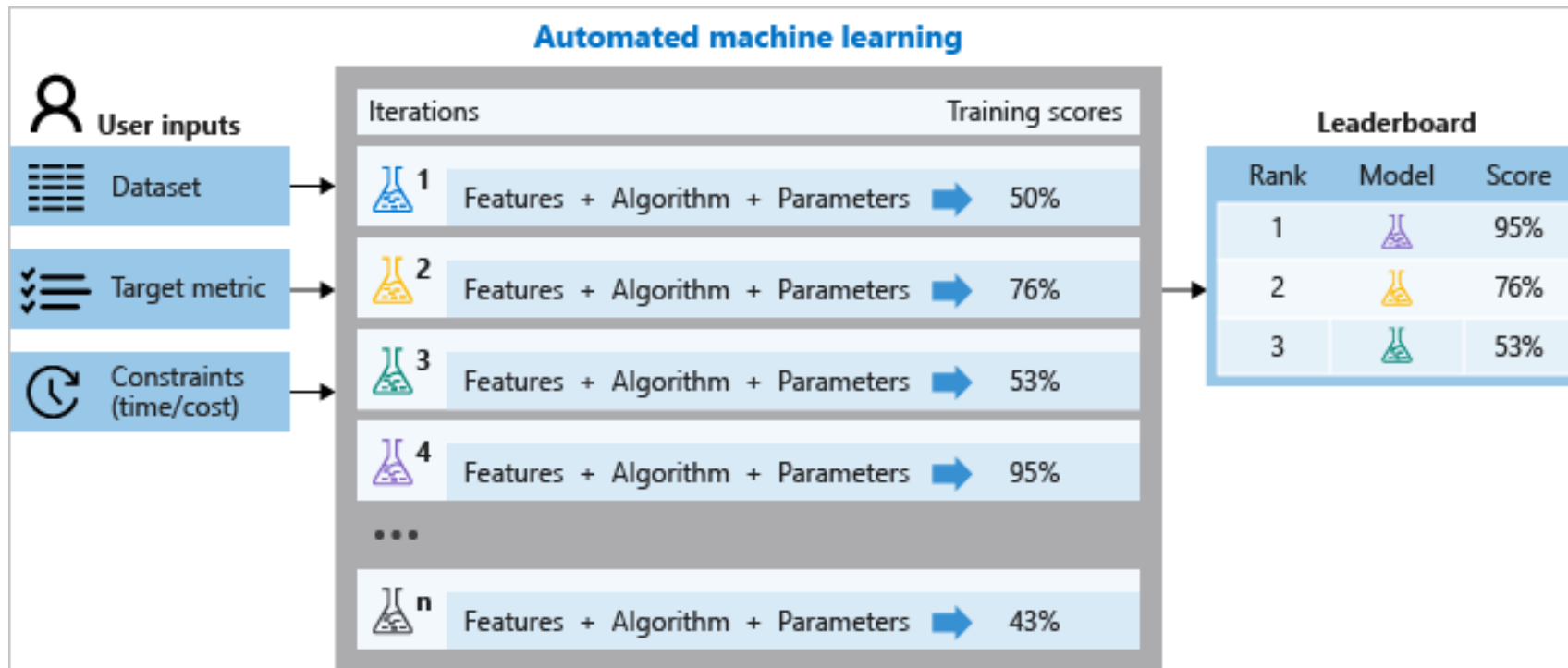
Experiment
Data Acquisition
Business Understanding
Initial Modeling

Develop
Modeling + Testing
Continuous Integration
Continuous Deployment

Operate
Continuous Delivery
Data Feedback Loop
System + Model Monitoring

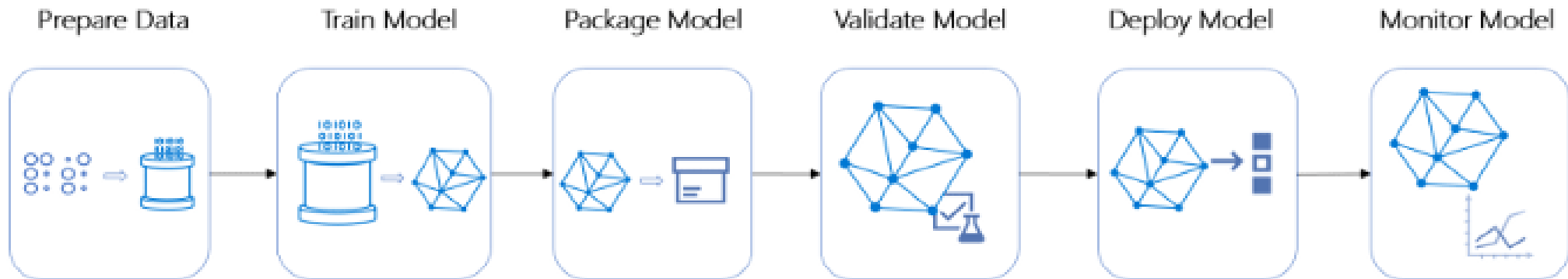
How AutoML works

- ▶ Azure Machine Learning creates a few pipelines in parallel that try different algorithms and parameters for us.
- ▶ The service iterates through ML algorithms paired with feature selections, where each iteration produces a model with a training score.
- ▶ The higher the score, the better the model is considered to "fit" our data. It will stop once it hits the exit criteria defined in the experiment.

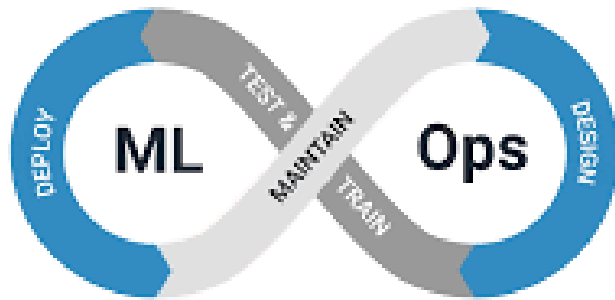


Azure Pipelines

- ▶ Azure Pipelines are cloud-hosted pipelines that are fully integrated with Azure DevOps.
- ▶ Azure DevOps allows us to frequently update models, test new models, and continuously roll out new ML models alongside our other applications and services.
- ▶ The end-to-end Machine learning pipeline includes data-prep, training, packaging and validating model, deploying model and continuous testing. It looks like below

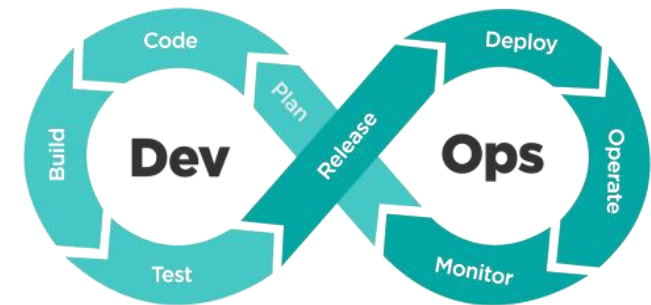


MLOps and DevOps Aims

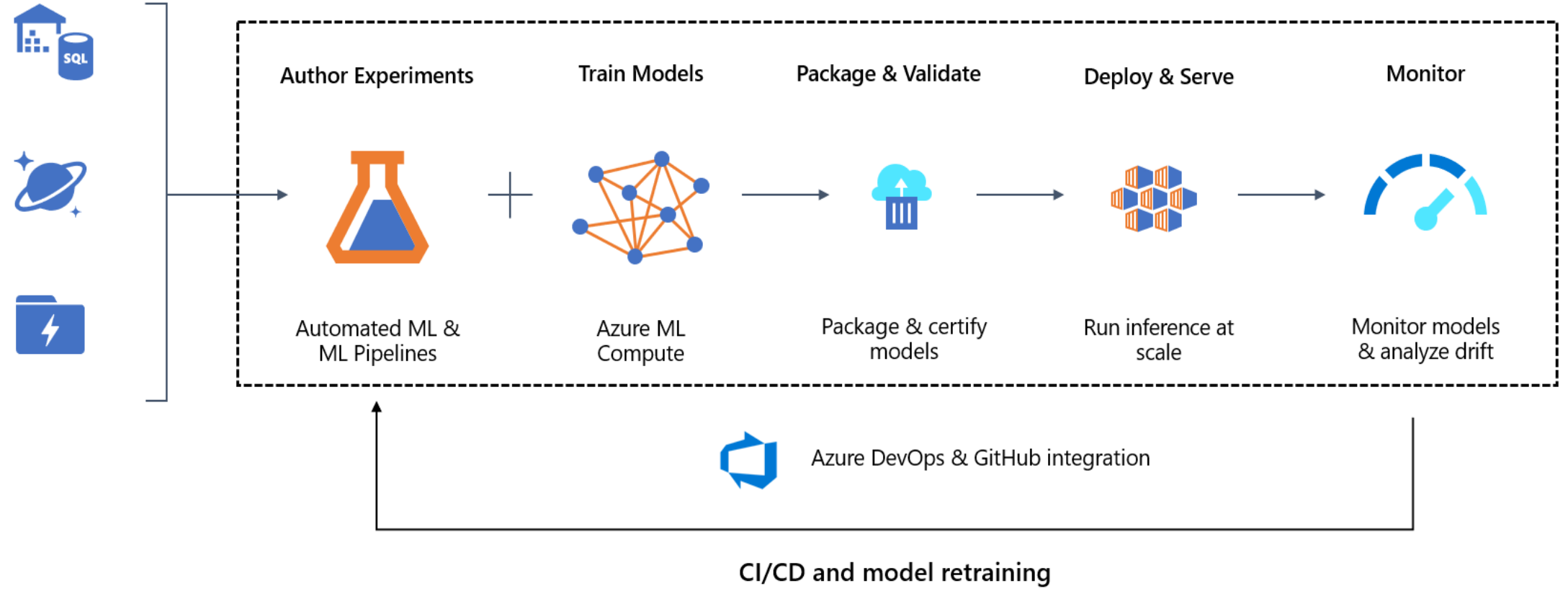


MLOps aims to establish a culture and environment where ML technologies can generate business benefits by rapidly, frequently and reliably building, testing, and releasing ML technology into production.

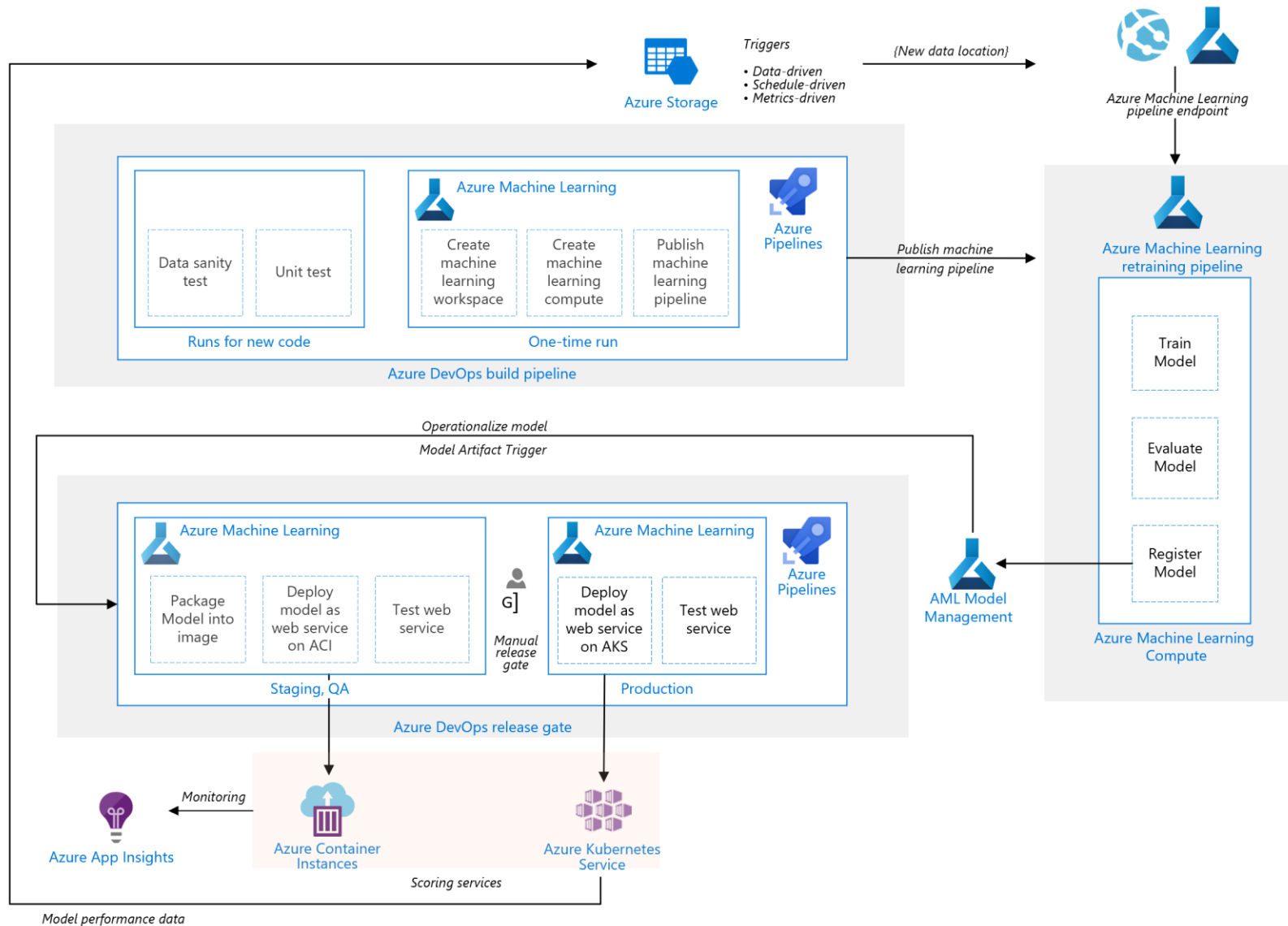
We will be using the Azure DevOps project for build and release pipelines along with Azure AutoML services for ML/AI model management and operationalization.



Azure Machine Learning service



CI/CD pipeline will look like below



Architecture and Features



- Implement continuous integration (**CI**), continuous delivery (**CD**), and training pipeline for an AI application using **Azure DevOps** and **Azure Machine Learning**.
- The solution is built on the **classification** scenario but can be easily adapted for any AI scenario.
- The build pipelines include DevOps tasks for data sanity tests, unit tests, **automated model training** on different compute targets.
- Selecting the **best model** including model version management, **model evaluation/model selection**, model deployment as **real-time web service**, staged deployment to QA/prod and integration testing.

Configure CI pipeline

- ▶ In this step, we will configure CI pipeline for your ML/AI project. This pipeline will include DevOps tasks for
 - ▶ data sanity test
 - ▶ model training on different compute targets
 - ▶ model version management
 - ▶ model evaluation/model selection etc.

Azure DevOps Dashboard

The screenshot shows the Azure DevOps interface for a project named "AutoML_Introact_MLOps". The top navigation bar includes the user name "anishtalukdar", the project name, and a search bar. The left sidebar contains navigation options: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main content area is divided into three sections: "About this project" with a description and an "Add Project Description" button; "Project stats" showing 0 pull requests and 0 commits; and "Members" showing 1 member (AT). The "Project stats" section also includes a dropdown for "Last 7 days" and two progress indicators for "Builds succeeded" and "Deployments succeeded", both at 0%.

Azure DevOps anishtalukdar / AutoML_Introact_MLOps / Overview / Summary

Search

A AutoML_Introact_MLOps Public Invite

About this project

Help others to get on board!
Describe your project and make it easier for other people to understand it.

+ Add Project Description

Project stats

Last 7 days

Repos

0 Pull requests opened

0 Commits by 0 authors

Pipelines

0% Builds succeeded

0% Deployments succeeded

Members

1

AT

Git Repository

The screenshot shows the Azure DevOps interface for a Git repository named 'AutoML_Introact_MLOps'. The left sidebar contains navigation options: Overview, Boards, Repos (selected), Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, and Project settings. The main area displays the repository structure with folders like aml_config, data, deployment, media, setup, tests, and training, along with files .gitattributes and azure-pipelines.yml. The right pane shows the 'Files' view for the 'master' branch, displaying a table of file changes.

Search:

Files ✓ succeeded

Contents History

Name ↑	Last change	Commits
aml_config	3 Jul	a2102691 Added run_id,
data	13 Jun	42d7c5bd Deleted config
deployment	6 Jul	b8e013a6 Updated score
media	10 Jun	ebe40955 initial commit
setup	5 Jul	8a8cf015 Updated requi
tests	11 Jul	d4236a06 Updated integ
training	17 Jul	64500554 Updated Evalu
.gitattributes	10 Jun	f992bed6 Initial commit
azure-pipelines.yml	13 Jun	08598097 Set up CI with

Pipeline configs will look like below

The screenshot displays the Azure DevOps interface for configuring a pipeline. The left sidebar shows the navigation menu with 'Pipelines' selected. The main area shows the pipeline configuration for 'AutoML_Introact_MLOps-CI'. The pipeline is currently in the 'Build pipeline' state. The configuration is organized into sections: 'Pipeline' (Build pipeline), 'Agent job 1' (Run on agent), and a list of tasks. The tasks listed are: 'Use Python 3.6' (Use Python version), 'Install Python Requirement' (Bash), 'Data Quality Check' (Bash), 'Publish Test Results **/test-*.xml' (Publish Test Results), 'Add AzureML CLI Extension' (Azure CLI), 'Create AzureML Service Workspace' (Azure CLI), 'Create Azure ML Compute Cluster' (Azure CLI), and 'Upload Data to the Default Datastore' (Disabled: Azure CLI). The right-hand side of the interface shows the configuration details for the pipeline, including the Name (AutoML_Introact_MLOps-CI), Agent pool (Azure Pipelines), Agent Specification (ubuntu-16.04), and Parameters (This pipeline doesn't have any pipeline parameters. Create them to share the most important settings between tasks and change them in one place. Learn more).

Azure DevOps | anishtalukdar / AutoML_Introact_MLOps / Pipelines

AutoML_Introact_MLOps +

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts
- Project settings <<

AutoML_Introact_MLOps-CI

Tasks | Variables | Triggers | Options | Retention | History | Save & queue | Discard | Summary | Queue | ...

Pipeline Build pipeline

Get sources AutoML_Introact_MLOps master

Agent job 1 Run on agent

- Use Python 3.6 Use Python version
- Install Python Requirement Bash
- Data Quality Check Bash
- Publish Test Results **/test-*.xml Publish Test Results
- Add AzureML CLI Extension Azure CLI
- Create AzureML Service Workspace Azure CLI
- Create Azure ML Compute Cluster Azure CLI
- Upload Data to the Default Datastore Disabled: Azure CLI
- Create Metadata and Model folders

Name *
AutoML_Introact_MLOps-CI

Agent pool * | Pool information | Manage | Azure Pipelines

Agent Specification *
ubuntu-16.04

Parameters ⓘ
This pipeline doesn't have any pipeline parameters. Create them to share the most important settings between tasks and change them in one place.
[Learn more](#)

Configure CD pipeline

So, as we are done with CI pipeline let's configure Release pipeline which will deploy the image created from the build pipeline to Azure Container Instance and Azure Kubernetes Services.

The screenshot displays the Azure DevOps interface for configuring a Release pipeline. The breadcrumb navigation shows the path: Pipelines / Releases / New release pipeline / Release-50. The left sidebar contains navigation options: Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main content area shows the 'Release' section with a 'Manually triggered' release by Anish Talukdar on 11/07/2020 at 16:08. Below this, an artifact is listed: '_AutoML_Introact_MLO...' with version 274 and branch 'master'. The 'Stages' section shows two stages: 'Deploye to pre-produ' (note the typo) and 'Deploy to Production'. Both stages are marked as 'Succeeded' with a green checkmark. The 'Deploye to pre-produ' stage has a warning icon and occurred on 11/07/2020 at 16:27. The 'Deploy to Production' stage occurred on 11/07/2020 at 23:12. Both stages show a progress indicator of 100%.

Thank You

