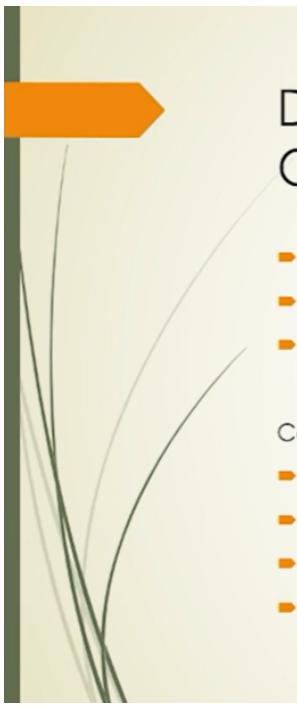




# Deploying Pipelines through Azure DevOps



## DevOps – Development and Operations

- ▶ Cultural Philosophies
- ▶ Practices
- ▶ Tools

Core principles of DevOps:

- ▶ Continuous integration and Continuous delivery (CI/CD)
- ▶ Real-time monitoring
- ▶ Incident response systems
- ▶ Collaboration platforms

## Benefits of DevOps

- ▶ Accelerating time to market
- ▶ Adapting to market and competition
- ▶ Maintaining system stability and reliability
- ▶ Improving mean time to recovery

## DevOps and the Application Life-cycle

- ▶ Plan
- ▶ Develop
- ▶ Deliver
- ▶ Operate



## Azure DevOps

- Plan
- Collaborate
- Build
- Deploy



## Azure DevOps service

- Quick set-up
- Maintenance-free operations
- Easy collaboration across domains
- Elastic scale
- Rock-solid security



## DevOps and the Cloud

- Cloud Agility
- Orchestration
- Serverless Computing



## Features of Azure DevOps

- Flexible
- Platform agnostic
- Cloud agnostic



## Azure DevOps services

- ▶ Azure repos
- ▶ Azure pipelines
- ▶ Azure Boards
- ▶ Azure Test Plans
- ▶ Azure Artifacts



## Azure DevOps services

- ▶ Collaboration tools
- ▶ Wikis
- ▶ Notification tools
- ▶ Extension support

## Problem Statement

### Objective

- To assess if a customer's license should be issued (AAI), renewed (REV) or cancelled(AAC) depending on various parameters

### Machine Learning Problem

- Develop a machine learning model to learn relation of the target variable with the set of features available from the training data

### Technology

- Python, h2O, Scikit-learn, tensorflow, keras, Pandas, Numpy

### Deployment

- Deploy model in a scalable way so that business decisions can be taken in near real time in assessing a customer's license status

## Solution System

- Code
- API
- Docker
- Container registry
- Web App

## Predictive classification problem

- ▶ Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data
- ▶ A classification can have real-valued or discrete input variables
- ▶ Several types: binary classification problem, multi-class classification problem and multi-label classification problems
- ▶ Class labels are often string values and must be mapped to numeric values before being provided to an algorithm for modeling.
- ▶ Classification accuracy is a popular metric used to evaluate the performance of a model

## Git

- ▶ Distributed Version Control System
- ▶ Platform and Environment friendly
- ▶ Enables code review with pull requests
- ▶ Branch protection with policies
- ▶ Extend pull request with pull statuses
- ▶ Enables code isolation with forks



## TFVC

- ▶ Centralized version control system
- ▶ Server Workspaces and Local workspaces
- ▶ Not good practice to use it anymore



## Cloning Azure Repos

- ▶ HTTPS
- ▶ SSH endpoints
- ▶ VSCode



## Perquisites

- Azure DevOps account
- Git



## FastAPI

- Modern, fast, high-performance web framework for building APIs with Python
- Uses a web server called ASGI or Asynchronous Server Gateway Interface
- Starlette for the web parts.
- Pydantic for the data parts.



## Features of FastAPI

- Fast
  - Fast to code
  - Fewer bugs
  - Intuitive
  - Easy
  - Short
  - Robust
  - Standards-based
- 



## Routing methods

- **Routing** refers to how an application's endpoints (URLs) respond to client requests.
  - GET() Method
  - POST() Method
- 

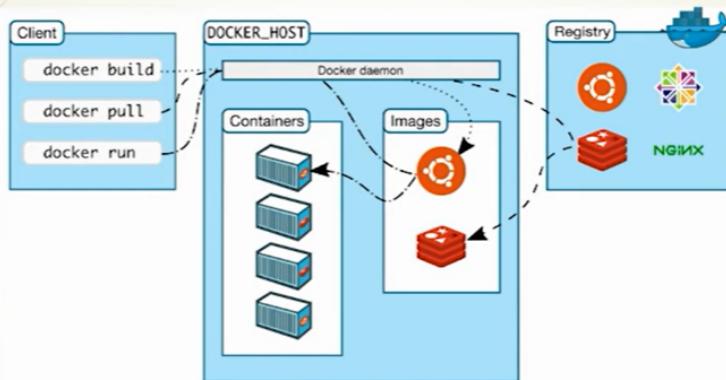
## Docker

- Docker is an open platform for developing, shipping, and running applications
- Manage infrastructure and applications
- Ability to package and run an application in a loosely isolated environment called a container
- Containers are lightweight and contain everything needed to run the application

## Uses of Docker

- Docker streamlines the development lifecycle by allowing developers to work in standardized environments
- Containers are great for (CI/CD) workflows.
- Responsive deployment and scaling
- Running more workloads on the same hardware

## Docker Architecture



## Docker Images and Containers

- ▶ An *image* is a read-only template with instructions for creating a Docker container.
- ▶ We create a Dockerfile with a simple syntax for defining the steps needed to create the image and run it.
- ▶ A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI.
- ▶ A container is defined by its image as well as any configuration options you provide to it when you create or start it.



## Azure Resource Group

- ▶ Container that holds related resources for an Azure solution
- ▶ Allocate resources to resource groups
- ▶ Stores metadata about the resources



## Benefits of Azure Container Registry

- ▶ Store and manage images for all types of container deployments
- ▶ Automated container builds, testing and security scanning
- ▶ Store your container image in local, network-close storage on Azure
- ▶ Use common Command Line Interface (CLI) to interact with the registry
- ▶ Manage windows and Linux container images in a single registry



## Advantages of Azure Resource Groups

- ▶ Straightforward way to create and manage resources
- ▶ Provides a flexible, customizable, high-level view of available resource
- ▶ Collects metadata from each individual resource to facilitate more granular management than at the subscription level
- ▶ Effective administration, cost management and role-based access controls.



## Azure Pipelines

- ▶ Automatically builds and tests code projects
- ▶ Continuous Integration + Continuous Delivery
- ▶ Continuous Testing



## CI/CD/CT

- Continuous Integration (CI) is the practice used by development teams of automating merging and testing code
- Continuous Delivery (CD) is a process by which code is built, tested, and deployed to one or more test and production environments
- Continuous Testing (CT) is the use of automated build-deploy-test workflows that test your changes continuously



## Features of Azure Pipelines

- Version control systems
- Languages
- Application types
- Deployment targets
- Continuous testing
- Package formats



## Perquisites

- Azure DevOps account
- Source code in Version Control System



## Benefits of Azure Pipelines

- Works with any language or platform
- Deploys to different types of targets at the same time
- Integrates with Azure deployments
- Builds on Windows, Linux, or Mac machines
- Integrates with GitHub
- Works with open-source projects.

## CI vs. CD

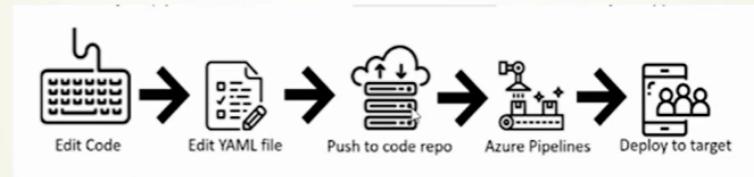
Continuous integration (CI)	Continuous delivery (CD)
<ul style="list-style-type: none"><li>- Increase code coverage</li><li>- Build faster by splitting test and build runs</li><li>- Automatically ensure you don't ship broken code</li><li>- Run tests continually</li></ul>	<ul style="list-style-type: none"><li>- Automatically deploy code to production</li><li>- Ensure deployment targets have latest code</li><li>- Use tested code from CI process</li></ul>

## Defining Pipelines

- YAML Syntax
- Classic Interface

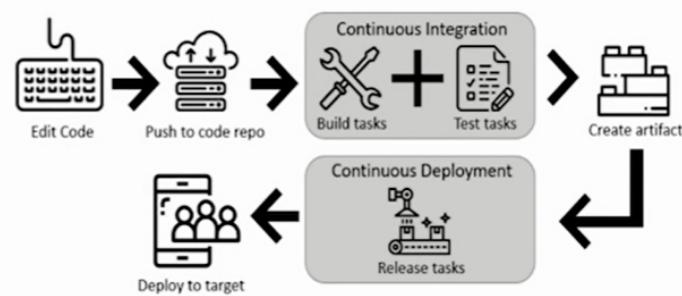
## Defining Pipeline through Yaml

- Define pipeline in a YAML file called `azure-pipelines.yml`



## Defining Pipelines through Classic Interface

- Create and configure pipelines in the Azure DevOps web portal with the Classic user interface editor



## YAML

- ▶ Data serialization language that is used for writing configuration files
- ▶ YAML is for data, not documents
- ▶ Human-readable and easy to understand
- ▶ YAML has features that come from other programming languages
- ▶ The structure of a YAML file is a map or a list

## Customize YAML Pipeline

- ▶ Change the platform for build
- ▶ Add steps
- ▶ Build across multiple platforms
- ▶ Build using multiple versions
- ▶ Customize CI triggers



## Azure Web App

- ▶ Create and deploy scalable web applications
- ▶ End point for deployment of image