**What is MLOps**

* **MLOps** - Combination of DevOps-style automation methods and ML best practices focused on deployment and operations of ML models into production systems.
* **DevOps** - Culture of collaboration and automation of developer operations to continuously improve deployment and reduce friction between business units. Includes automation tools and principles like CI/CD.
* **CI/CD** - Continuous integration and continuous delivery - key practices in DevOps workflows. CI involves **regular automated testing and validation** of incrementally - built software while CD focuses on **automatically releasing updates** to environments like production.
* **Maturity Model** - Concept in MLOps defining ascending levels, typically 4-5, of MLOps sophistication and effectiveness, from manual, siloed, unreliable to scalable, autonomous, and resilient systems.
* **Data Ops** - Specific focus on use of DevOps concepts and automation principles to **manage data** workloads like aggregation, transformation, storage, analysis, etc.
* **Feature Store** - **Central repository** to manage, store, and serve ML features for model building retraining. Optimizes data pipelines and reuse.

| **Reading Title** | **Brief Description** |
| --- | --- |
| [MLOps: Continuous delivery and automation pipelines in machine learning](https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning#mlops_level_1_ml_pipeline_automation) | This article serves as a primer article to different approaches for an MLOps pipeline for a given application. It leverages the knowledge and experience of Google's history of ML services to illustrate some common pitfalls of MLOps projects. |
| [What is DevOps?](https://www.atlassian.com/devops) | This article provides an overview of the history, definition, and benefits of DevOps, as well as linking out to a number of useful resources for applying DevOps principles to your organization or workflow. |
| [Sowing the seeds of ethical AI: 4 tasks to stay on track](https://cloud.google.com/blog/transform/4-tasks-to-ensure-your-companys-ai-is-ethical) | This article outlines four guiding principles that should be considered and implemented during the development and delivery of any public-facing AI process. |
| [Build, Test and Deploy ETL solutions using AWS Glue and AWS CDK](https://aws.amazon.com/blogs/big-data/build-test-and-deploy-etl-solutions-using-aws-glue-and-aws-cdk-based-ci-cd-pipelines/) | This article provides a step-by-step process to create a continuous integration/continuous delivery solution using AWS Glue and AWS CDK. It also provides all of the example code to create this workflow in your own AWS account, and encourages you to use this process as a model to create your own AWS solutions. |

# Key Concepts in MLOps

**MLOps Platform:** A specialized software solution and workflow for operationalizing machine learning models. MLOps platforms have capabilities like data labeling, model monitoring, feature stores, and optimized model serving.

**Continuous Integration (CI):** An automated software development practice where developers frequently merge code changes into a shared repository. Changes are then automatically built and tested to catch issues early.

**Continuous Delivery:** A software engineering approach where teams produce software in short cycles, ensuring software can be reliably released at any time. It depends on automation like infrastructure as code to replicate test/prod environments.

**Infrastructure as Code:** Managing and provisioning infrastructure through code instead of manual processes. This allows environment configuration, deployment, and management to be consistent and repeatable across development stages.

**Feature Store:** A centralized repository that stores curated features for machine learning model training. This helps manage data for model creation, storage, and discovery while preventing drift.

| **Reading Title** | **Brief Description** |
| --- | --- |
| [Increase reliability in data science and machine learning projects with CircleCI](https://circleci.com/blog/increase-reliability-in-data-science-and-machine-learning-projects-with-circleci/) | This article goes over the importance of automated testing, and setting up CircleCI to help accommodate that process. Additionally it also provides the starter code to get you up and running with CircleCI, and links to useful documentation about further utilization of the service. |
| [Machine Learning operations maturity model](https://learn.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model?WT.mc_id=academic-0000-alfredodeza) | This article outlines the MLOps maturity model, which clarifies the important processes, technologies, and people that are present in progressively more mature MLOps environments. It's beneficial to understand what might be required to successfully implement and maintain an MLOps environment of a given maturity. |
| [DevOps tech: Continuous delivery](https://cloud.google.com/architecture/devops/devops-tech-continuous-delivery) | This article explains the organizational and technical benefits of implementing a Continuous Delivery workflow. Additionally, it illustrates some of the typical obstacles that organizations have to implementing a CD workflow, and proposes practical approaches to overcome them. |
| [Detect data drift on datasets](https://learn.microsoft.com/en-us/azure/machine-learning/v1/how-to-monitor-datasets?tabs=python&view=azureml-api-1) | This article explains the concept of data drift, and some workflows to help analyze and mitigate it. Additionally, it gives some example code and step by step instructions for utilizing some Azure Machine Learning workflows to help mitigate data drift in your own models. |
| [Automate MLOps with SageMaker Projects](https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-projects.html) | This is a link to the Amazon Sagemaker projects page, which is a subsection of the Amazon Sagemaker developer guide that can walk you through step by step instructions on using Sagemaker to create any number of MLOps projects. More than anything, this is useful documentation to come back to when you'd like to create a Sagemaker MLOps project. |
| [Databricks Feature Store](https://learn.microsoft.com/en-us/azure/databricks/machine-learning/feature-store/) | This short article explains what exactly a feature store is, and some of the benefits of Microsoft's Databricks feature store in particular. Additionally, it links to a number of useful APIs and example notebooks that can give you some hands-on practice to illustrate the value of feature stores. |

# Key Concepts in Microservices

**Microservice** - Encapsulated, reusable logic that is deployed into production environments.

**Continuous Integration (CI)** - The practice of frequently merging code changes into a shared repository and automatically building and testing changes to catch issues early.

**Continuous Delivery** - A development practice where incremental software changes can be reliably released at any time through automated deployments.

**End-to-End MLOps** - Fully automating the machine learning lifecycle from model development through deployment and hosting via platforms like Hugging Face Spaces.

**AWS App Runner** - A fully managed service for deploying containerized web services and APIs.

**Flask** - A popular, lightweight Python web application framework.

**Makefile** - A file containing a set of directives used to automate building and managing a project.

**Requirements File** - A text file containing a list of Python package dependencies used by an application.

| **Title** | **Brief Description** |
| --- | --- |
| [Build a Web Application with Flask](https://paiml.com/docs/home/books/minimal-python/chapter06-build-web-apps-flask/) | This is a chapter from a book authored by this course's instructors! It walks you through using the Flask web framework to create a web application that can use your python code. If you follow along with these instructions, you'll wind up with a functional web app. |
| [MLOps for Azure](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/manage/mlops-machine-learning) | MLOps with Azure Machine Learning offers an efficient way to manage the entire machine learning lifecycle. By following this reading, students will learn how to create reproducible models, reusable training pipelines, and simplify model packaging, validation, and deployment. Moreover, they will gain insights into best practices for quality control, A/B testing, and automating the retraining process. This knowledge will help students improve the quality and consistency of their machine learning solutions, making them better prepared for real-world applications. |
| [Introduction to Microservices](https://www.nginx.com/blog/introduction-to-microservices/) | This article introduces the concept of Microservices, as well as the benefits and drawbacks of using a microsystem-based approach. Understanding these kinds of systems gives a lot of insight into how many large web applications function, and helps solve the problem of scalability within systems you may design. |
| [Building a Go App on App Engine](https://cloud.google.com/appengine/docs/standard/go/building-app) | This tutorial will take you through the process of making a bare-bones Go App using google's AppEngine and Google Cloud. Using this free service will give you experience in deploying an application and using AppEngine to manage your deployments. |
| [Getting Started with App Runner](https://docs.aws.amazon.com/apprunner/latest/dg/getting-started.html) | This documentation is an introduction to Amazon's App Runner, which is a system that allows you to turn a container or source code into a web service using AWS. This is just a cursory overview of using App Runner to deploy whatever application you've made, and when you're creating web applications it will be important for you to consider which delivery method or product best suits your use case. |
| [Learn Github Actions](https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions) | This is the full documentation for Github actions, so you can choose to dive in as deeply as you desire. Github actions is a platform that can help you manage your continuous integration/continuous delivery pipelines, and configure workflows to trigger whenever specific events occur in a given repository, among other advanced functionality. |

**Summary of Lesson:** This lesson covered operationalizing microservices using CI/CD, building end-to-end MLOps solutions with Hugging Face models, deploying apps with AWS App Runner and Flask, leveraging makefiles and important Python files, and quizzed understanding of these key concepts.

**Top 3 Key Points**

* Microservices encapsulate reusable logic pushed to production
* End-to-end MLOps automates model deployment and hosting
* Structured Python projects enable testability and reliability