

Day 10 Agenda

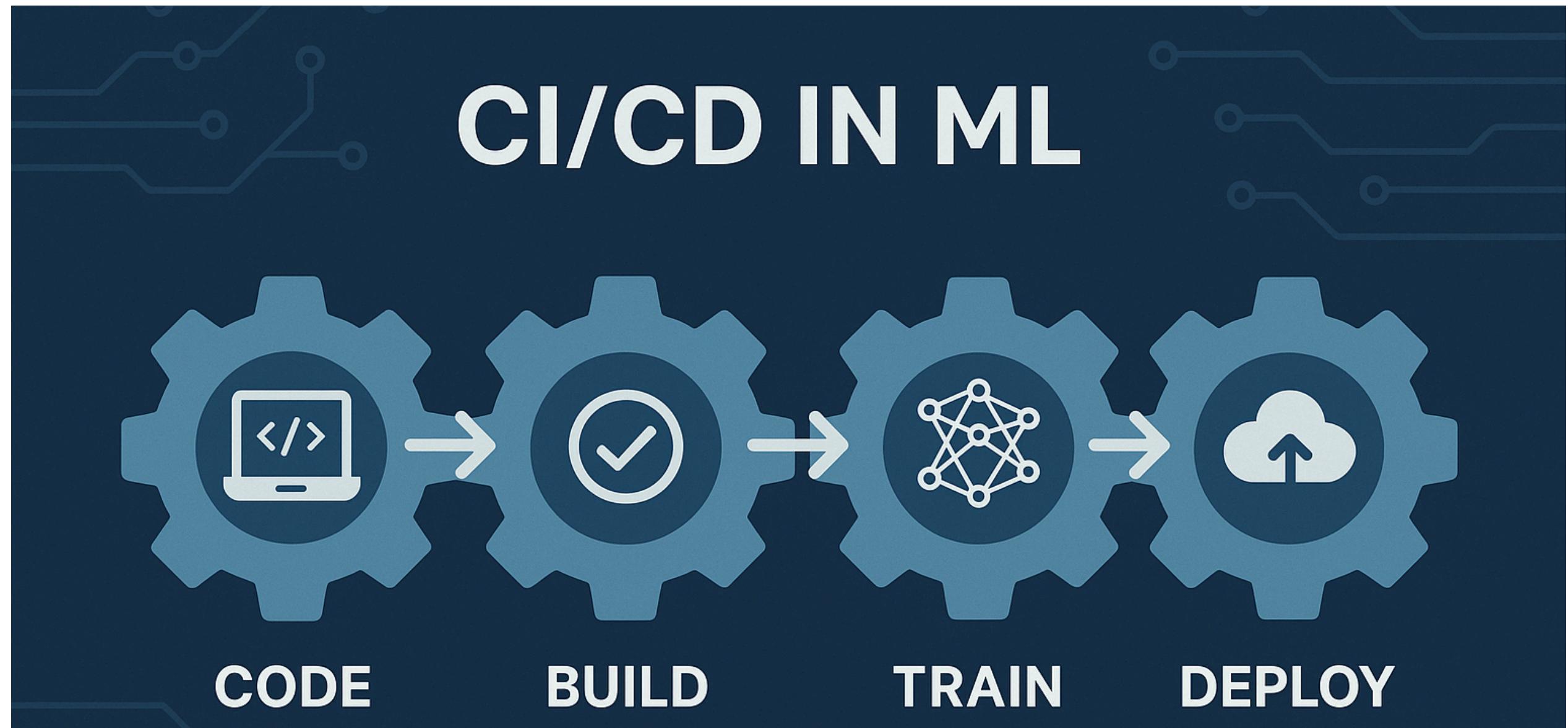
CI/CD + End-to-End Project

- Recap
- Final Quiz
- Building CI/CD workflows
- GitHub Actions for ML
- Deploying a complete ML pipeline
- Showcase of student/real-world projects
- Quiz

CI/CD



CI/CD



CI/CD

CI (Continuous Integration): Automatically testing and validating code changes (unit tests, linting, data checks).

CD (Continuous Delivery/Deployment): Automatically deploying changes to a staging or production environment (models, APIs, dashboards).

CI/CD in MLOps

In MLOps, continuous integration (CI) focuses on automating the building, testing, and packaging of machine learning pipelines whenever there are changes to the codebase in the version control system (like Git).

CD extends CI by automatically delivering or deploying the integrated code to production. In ML, this could involve deploying updated models or datasets to a production environment.

Building CI/CD workflows

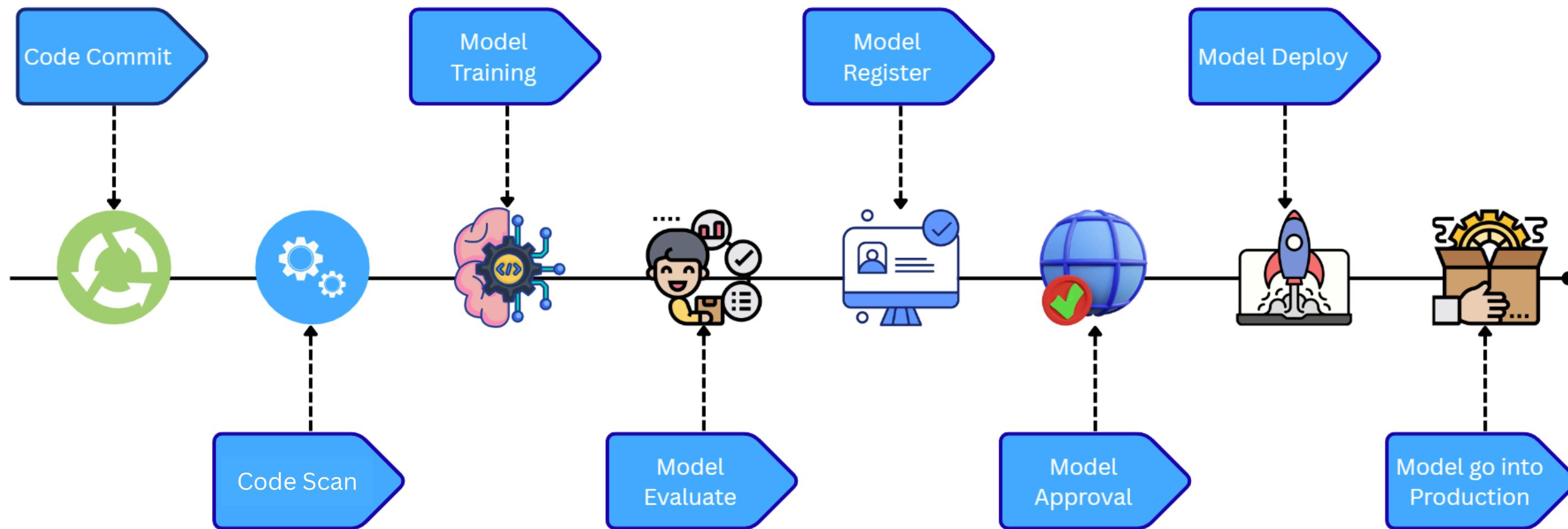
Core Components

- Source Control: GitHub, GitLab
- Build Triggers: On push, pull request, schedule
- Steps:
 - Code formatting & linting
 - Dependency installation
 - Model training/testing
 - Pushing model artifacts
 - Deployment to inference server

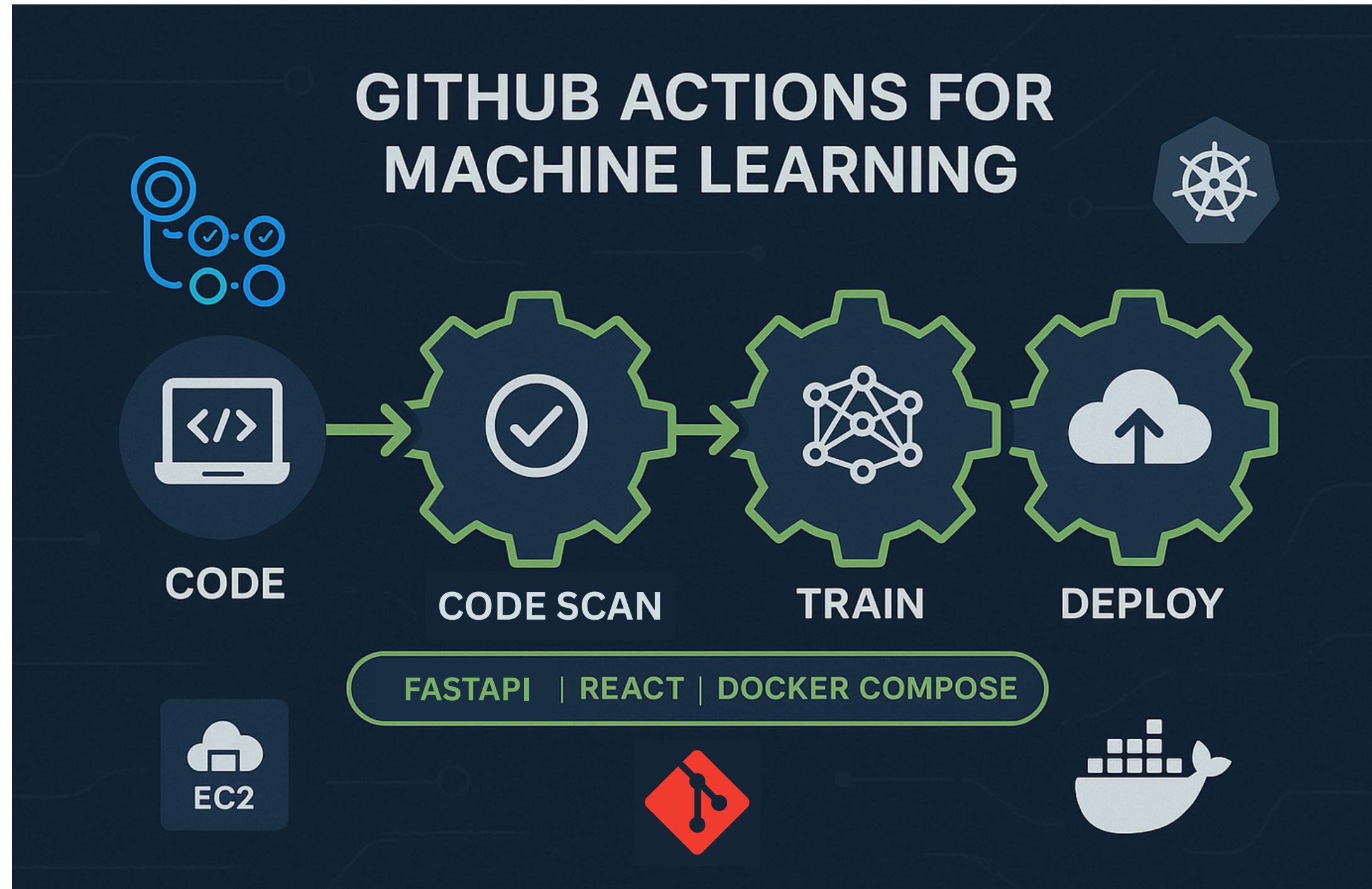
CI/CD Best Practices

Best Practices

- Keep environments reproducible (Docker/venv)
- Use environment variables/secrets for keys
- Version everything: code, data, models
- Isolate test, staging, and production environments



What Is GitHub Actions?





What Is GitHub Actions?

GitHub Actions lets you automate tasks like training models or deploying APIs, directly from your GitHub rep

How to perform CI/CD ?

Performing CI/CD using GitHub Actions involves creating a workflow YAML file that defines the automated stages of your machine learning pipeline.

Here's a general outline:

- **Set Up Your Repository:**
 - Make sure your MLOps project is hosted on a GitHub repository.

Feedback





How to perform CI/CD ?

- **Create a .github/workflows Directory:**
 - Inside your repository, create a new directory named .github/workflows. This is where you'll store your workflow YAML files.
- **Define Your CI/CD Workflow YAML File:**
 - Create a new YAML file within the .github/workflows directory. This file will define the stages of your CI/CD pipeline.

What is GitHub Runner ?

- In GitHub Actions, a “runner” is essentially a virtual machine or container instance that executes your workflows. When you create a workflow in GitHub Actions, it consists of one or more jobs, and these jobs run on a runner.
- Runners can be hosted by GitHub (GitHub-hosted runners) or by yourself (self-hosted runners).

What is GitHub Runner ?

- **GitHub-hosted runners:** These are virtual machines managed by GitHub. GitHub provides several types of hosted runners, each with different specifications for CPU, memory, and operating system. You can use these runners for free, but they have usage limits depending on your GitHub plan.
- **Self-hosted runners:** These are machines that you set up and manage yourself. They can be physical machines, virtual machines, or containers in your own infrastructure or a cloud provider. Self-hosted runners give you more control over the execution environment, and you can use them to run workflows on your own hardware or in environments that GitHub doesn't provide.