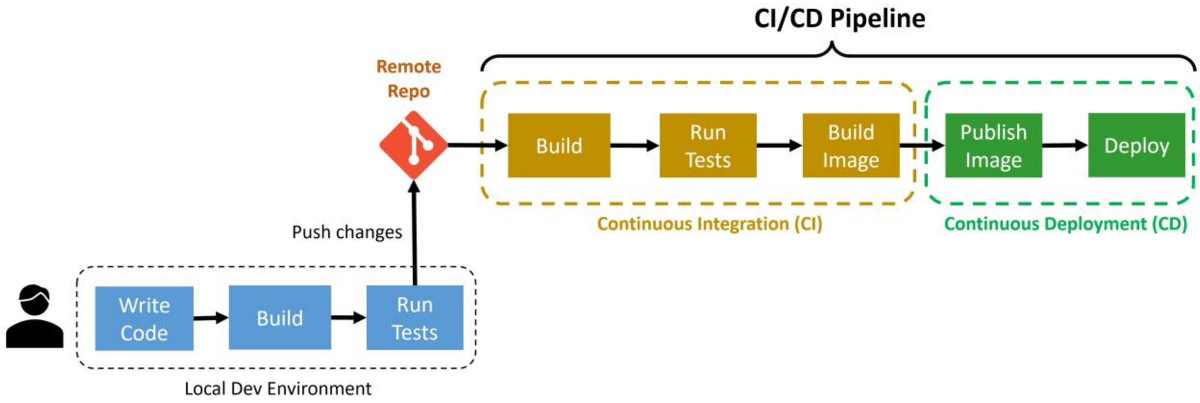
CI/CD

Introduction

Continuous Integration (CI) and Continuous Delivery/Deployment (CD) are modern software engineering practices that automate the process of building, testing, and releasing applications.
Azure DevOps, provided by Microsoft, offers a complete DevOps toolchain that supports CI/CD pipelines, helping teams deliver high-quality software faster and more reliably.

Key benefits of CI/CD in Azure DevOps:

- Faster and more frequent releases
- Early detection of bugs
- Consistency in builds and deployments
- Reduced manual effort with automation



What is Continuous Integration (CI)?

Continuous Integration is the practice of frequently merging code changes into a shared repository. Each commit triggers an automated build and test process to ensure the new code works well with the existing codebase.

CI in Azure DevOps:

- **Azure Repos** – Git-based repository to store code
- **Azure Pipelines** – Automates builds and runs unit tests
- **Build Validation** – Ensures that only tested, working code is merged
- **Pipeline Triggers** – Builds can be triggered on every code commit

Example CI Flow in Azure DevOps:

1. Developer pushes code to Azure Repos
2. Pipeline triggers an automated build
3. Code is compiled and unit tests run
4. Build artifacts (packages, executables, etc.) are generated

What is Continuous Delivery/Deployment (CD)?

Continuous Delivery extends CI by automatically preparing the code for release. Continuous Deployment goes one step further by deploying every successful build to production without manual intervention.

CD in Azure DevOps:

- **Release Pipelines** – Automates deployment to environments (Dev, Test, Staging, Production)
- **Approvals & Gates** – Add manual approval steps or quality checks before moving to production
- **Infrastructure as Code (IaC)** – Use ARM templates, Bicep, or Terraform for automated environment setup
- **Multi-stage Pipelines** – Define different stages (build, test, deploy) in YAML or Classic view

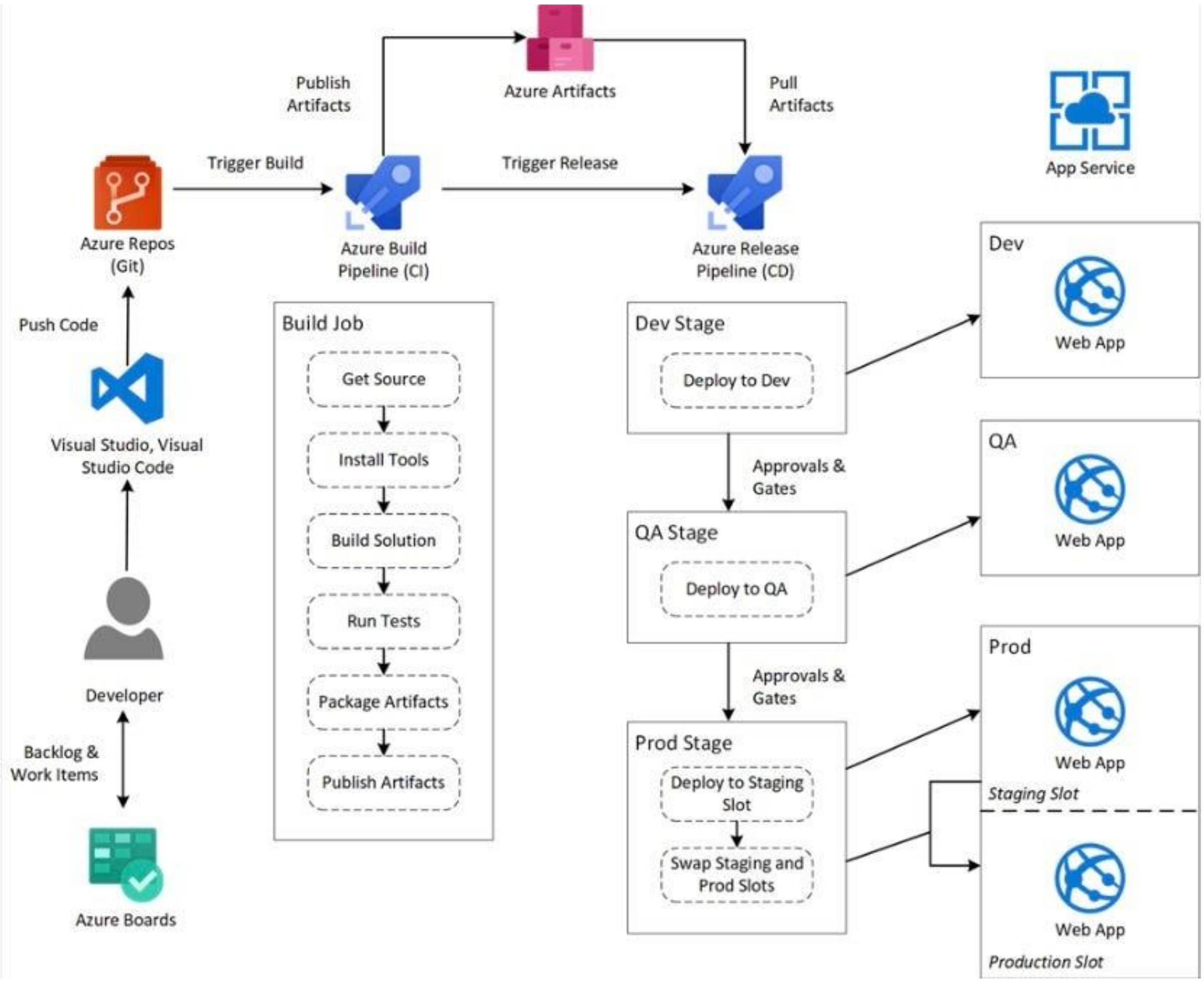
Example CD Flow in Azure DevOps:

1. Build artifact from CI pipeline is passed to release pipeline
2. Code is deployed to Dev/Test environment
3. Automated integration tests are run
4. After approval, code is deployed to production

Key Features of CI/CD in Azure DevOps

- **Scalability** – Works with any programming language and platform
- **Flexibility** – Supports YAML pipelines and visual designer
- **Security** – Secrets and credentials managed via Azure Key Vault integration

- **Monitoring** – Deployment logs and real-time insights available
- **Integration** – Works with GitHub, Docker, Kubernetes, Jenkins, and more


Publish Artifacts
Azure Artifacts
Pull Artifacts
App Service
Trigger Build
Trigger Release
Azure Repos (Git)
Azure Build Pipeline (CI)
Azure Release Pipeline (CD)
Push Code
Build Job
Get Source
Install Tools
Build Solution
Run Tests
Package Artifacts
Publish Artifacts
Visual Studio, Visual Studio Code
Developer
Backlog & Work Items
Azure Boards
Dev Stage
Deploy to Dev
Approvals & Gates
QA Stage
Deploy to QA
Approvals & Gates
Prod Stage
Deploy to Staging Slot
Swap Staging and Prod Slots
Dev
Web App
QA
Web App
Prod
Web App
Staging Slot
Web App
Production Slot