Agile Devops Tools – Azure Pipelines CICD

[**✅ Task 4: Azure Pipelines (CI/CD) – Automation** 1](#_Toc205556694)

[**1️⃣ Purpose** 1](#_Toc205556695)

[**2️⃣ Theory** 1](#_Toc205556696)

[**3️⃣ Prerequisites** 2](#_Toc205556697)

[**4️⃣ Code Example – YAML Pipeline for .NET Core App** 2](#_Toc205556698)

[**5️⃣ Project Snapshot** 3](#_Toc205556699)

[**6️⃣ Summary** 3](#_Toc205556700)

**✅ Task 4: Azure Pipelines (CI/CD) – Automation**

**1️⃣ Purpose**

Azure Pipelines is a service in Azure DevOps that automates the process of **building, testing, and deploying** applications. It supports **Continuous Integration (CI)** and **Continuous Deployment (CD)**, ensuring your code moves from **source to production** reliably and efficiently.

**2️⃣ Theory**

**🔹 What is Azure Pipelines?**

Azure Pipelines is a CI/CD service that integrates with:

* Azure Repos, GitHub, etc.
* Any language/platform (e.g., .NET, Java, Node.js).
* Multiple environments: on-prem, cloud (Azure, AWS, GCP).

**🔹 Key Concepts:**

| **Concept** | **Description** |
| --- | --- |
| CI (Continuous Integration) | Automatically builds and tests code on every commit/pull request. |
| CD (Continuous Deployment) | Automatically deploys code to staging/production after build & tests. |
| Pipeline | Workflow that defines build/deploy steps (YAML or Classic UI). |
| Agent | VM that runs the pipeline jobs (Microsoft-hosted or self-hosted). |
| Stages, Jobs, Steps | Stages contain jobs; jobs contain steps (script/command/tool execution). |

**🔹 Types of Pipelines:**

* **Classic (UI-based):** Drag-and-drop visual editor.
* **YAML (Pipeline as Code):** Text-based, stored in repo for versioning.

**3️⃣ Prerequisites**

* Azure DevOps project created.
* Code repository in Azure Repos or GitHub.
* Basic knowledge of CI/CD flow.
* For .NET: Build tools and SDKs installed.

**4️⃣ Code Example – YAML Pipeline for .NET Core App**

# azure-pipelines.yml at the root of your repo

trigger:

- main

pool:

vmImage: 'windows-latest'

variables:

buildConfiguration: 'Release'

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '7.x.x'

installationPath: $(Agent.ToolsDirectory)/dotnet

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

projects: '\*\*/\*.csproj'

- task: DotNetCoreCLI@2

inputs:

command: 'build'

projects: '\*\*/\*.csproj'

arguments: '--configuration $(buildConfiguration)'

- task: DotNetCoreCLI@2

inputs:

command: 'test'

projects: '\*\*/\*Tests/\*.csproj'

arguments: '--configuration $(buildConfiguration)'

**5️⃣ Project Snapshot**

📂 **Repo Structure**

student-course-app/

│

├── Controllers/

├── Models/

├── Views/

├── student-course-app.csproj

├── azure-pipelines.yml ← Pipeline config

└── ...

✅ Once you commit this file to your main branch, the pipeline will trigger automatically.

**6️⃣ Summary**

* Azure Pipelines provides **CI/CD automation** using YAML or Classic UI.
* Pipelines support multiple environments and languages.
* YAML Pipelines offer **version-controlled**, reusable build logic.
* Key steps: Restore → Build → Test → Publish → Deploy.