🚀 Azure DevOps CI/CD Pipelines with Azure Repos – Practice Lab

---

Prerequisites

- Azure account (with App Service enabled)
- Azure DevOps account (https://dev.azure.com)
- Visual Studio or VS Code installed with .NET SDK
- Git installed

Step 1: Create a Simple Web API Project

1. Open a terminal (cmd/PowerShell/bash).
2. Run the following commands:
3. `dotnet new webapi -o MyWebApiApp`
4. `cd MyWebApiApp`
5. `dotnet run`
    - Open http://localhost:5000/swagger to check the API.

Step 2: Push the Project to Azure Repos

1. Go to Azure DevOps → Create a new **Project** (e.g., *WebApiDemo*).
2. Inside the project → Go to **Repos** → Copy the Git clone URL.
3. In your terminal:
4. `git init`
5. `git remote add origin <Azure-Repo-URL>`
6. `git add .`
7. `git commit -m "Initial commit - Web API"`
8. `git push -u origin main`

✅ Now your Web API project is in **Azure Repos**.

Step 3: Create a CI Pipeline (Build & Test)

1. In Azure DevOps → Go to **Pipelines** → **Create Pipeline**.
2. Choose **Azure Repos Git** → Select your repo.
3. Choose **Starter Pipeline** → Replace YAML with this:

```
trigger:
  branches:
    include:
      - main

pool:
  vmImage: 'windows-latest'

steps:
- task: UseDotNet@2
  inputs:
    packageType: 'sdk'
    version: '9.x'

- script: dotnet restore
  displayName: 'Restore dependencies'

- script: dotnet build --configuration Release
  displayName: 'Build project'

- script: dotnet test --configuration Release --no-build
  displayName: 'Run tests'

- task: PublishBuildArtifacts@1
  inputs:
    pathToPublish: '$(Build.ArtifactStagingDirectory)'
    artifactName: 'drop'
```

31. Save & Run → The pipeline will:
    - Restore packages
    - Build solution
    - Run tests
    - Publish build artifacts

✓ CI Pipeline ready.

Step 4: Create an Azure Web App with Deployment Slots

1. Go to **Azure Portal → App Services → Create App Service**.
    - Runtime: **.NET 9 (or latest)**
    - OS: **Windows/Linux**
    - Name: `mywebapidemo`
    - Resource Group: `WebApiDemoRG`
2. After creating, go to **Deployment slots** → Add:
    - **UAT**
    - **Staging**
    - (Production slot is the default one)

✓ App Service with **3 slots** is ready.

Step 5: Create a CD Release Pipeline

1. In Azure DevOps → Go to **Pipelines** → **Releases**.
2. Click **New Pipeline** → Choose **Empty Job**.
3. **Add an artifact** → Select the **Build Artifact (drop)** from Step 3.
4. Add **3 stages**:
   - **UAT Deployment**
   - **Staging Deployment**
   - **Production Deployment**
5. For each stage:
   - Add a task → **Azure App Service Deploy**.
   - Configure:
     - Azure Subscription (service connection)
     - App Service name: `mywebapidemo`
     - Deployment slot: (UAT / Staging / Production)
     - Package or folder: `$(System.DefaultWorkingDirectory)/_MyWebApiApp/drop`
6. Add **automatic triggers**:
   - UAT: Continuous deployment trigger ON
   - Staging: After UAT succeeds (auto-approval)
   - Production: After Staging succeeds (manual approval before deployment)

✅ CD Release pipeline ready.

Step 6: Test the End-to-End CI/CD

1. Make a change in the Web API (e.g., update WeatherForecast controller).
2. Commit & push to **main branch**.
3. CI pipeline runs → Build + Test → Artifacts published.
4. Release pipeline starts:
   - Deploys to **UAT slot**
   - Auto deploys to **Staging slot**
   - Waits for approval → Deploys to **Production slot**
5. Verify deployment:
   - UAT URL → `https://mywebapidemo-uat.azurewebsites.net/swagger`
   - Staging URL → `https://mywebapidemo-staging.azurewebsites.net/swagger`
   - Production URL → `https://mywebapidemo.azurewebsites.net/swagger`

✅ Congratulations! You've built a **complete CI/CD pipeline with Azure DevOps, Azure Repos, and Azure App Service with slots** 🎉