# CI/CD & Azure DevOps Documentation

## 1. Introduction

This document outlines the steps used to upload an ASP.NET Core project to Azure DevOps, configure Continuous Integration (CI) and Continuous Deployment (CD) pipelines, and automate deployment using Azure DevOps Pipelines.

## 2. SOLID Principles Overview

S - Single Responsibility Principle

O - Open/Closed Principle

L - Liskov Substitution Principle

I - Interface Segregation Principle

D - Dependency Inversion Principle

## 3. Git Version Control

Git is used to track project code changes.

Key commands include git add, git commit, git push, and git pull.

## 4. Uploading Code to Azure DevOps Repo

1. Create a project in Azure DevOps

2. Create/Use a repository

3. Clone repository using HTTPS

4. Add project files and push using Git

## 5. Creating CI Pipeline in Azure DevOps

Steps:

- Navigate to Pipelines → Create Pipeline

- Select Azure Repos Git

- Choose a starter pipeline

- Add YAML steps for restore, build, publish, and artifact upload

### 6. YAML Pipeline Example

The following tasks are typically included:

- dotnet restore

- dotnet build

- dotnet publish

- PublishBuildArtifacts

### 7. Enabling Continuous Deployment (CD)

A release pipeline or YAML deployment step can automatically push published artifacts to an Azure Web App or chosen environment.

### 8. Conclusion

This documentation summarizes how CI/CD automation is implemented using Azure DevOps for ASP.NET Core applications.