**Automated Deployment of ASP.NET Web App to Azure App Service using Azure DevOps Pipelines**

Automate the CI/CD deployment of an ASP.NET web application to Azure App Service using Azure DevOps YAML pipeline. The pipeline handles restoring dependencies, building, publishing artifacts, and deploying to staging or production slots.

 **Create ASP.NET Project**  
Use the dotnet CLI to scaffold a sample ASP.NET Core Razor Pages app locally.

 **Initialize Git Repository**  
Initialize a Git repository inside the project and push it to GitHub or Azure Repos.

 **Create Azure App Service**  
Create an App Service Plan and Web App in Azure to host the ASP.NET app.

 **Create Azure DevOps Project and Pipeline**  
In Azure DevOps, create a YAML pipeline to automate:

* dotnet restore – restore NuGet packages
* dotnet build – compile the app
* dotnet publish – generate build artifacts
* Publish artifacts to DevOps
* Deploy to Azure App Service using the task AzureWebApp

 **Configure Deployment Slot (Optional)**  
Add a staging slot and deploy to it for validation before pushing to production.

 **Secure App Settings and Secrets**  
Manage secrets (like connection strings) using Azure DevOps library or App Service settings.

 **Verify Deployment**  
Open the Azure-hosted web app in the browser and confirm successful deployment.

Install dotnet 9.0 SDK

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Create the ASP.NET App**

A screenshot of a computer

AI-generated content may be incorrect.

dotnet new webapp: creates a Razor Pages web app template (recommended for Azure App Service).

-n WebAppAspNet: names the project folder as WebAppAspNet.

cd WebAppAspNet: moves into the newly created project directory.

**Initialize Git & Push to GitHub**

“git init

git add .

git commit -m "Initial commit - ASP.NET Core app"

A screenshot of a computer

AI-generated content may be incorrect.

**Create a new GitHub repo**: aspnet-azureappservice-deployment

A screenshot of a computer

AI-generated content may be incorrect.

**Connect local repo to GitHub and push**

A screenshot of a computer

AI-generated content may be incorrect.

**The code is live on GitHub**

A screenshot of a computer

AI-generated content may be incorrect.

**Create Azure Resources (App Service + App Service Plan)**

Before setting up the CI/CD pipeline, we need the target environment ready in Azure.

**Option 1: Azure Portal (Manual — beginner-friendly)**

1. **Create Resource Group**
   * Name: rg-aspnet-deploy
2. **Create App Service Plan**
   * Name: aspnet-serviceplan
   * Region: your nearest Azure region
   * Pricing tier: B1 (Basic) or F1 (Free if available)
3. **Create App Service (Web App)**
   * Name: must be globally unique, e.g., webapp-aspnet-pradeep
   * Runtime stack: .NET 8 (LTS) or .NET 7 (as your SDK is .NET 9 preview, but App Service supports .NET 8 latest)
   * Region: same as App Service Plan

A screenshot of a computer

AI-generated content may be incorrect.

Create App Service Plan

A screenshot of a computer

AI-generated content may be incorrect.

Create Azure Resources (App Service + App Service Plan)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.**Create the Azure DevOps YAML Pipeline**

* Restore NuGet packages
* Build the app
* Publish output to DevOps as an artifact
* Deploy to your Azure App Service

Create a File Named azure-pipelines.yml in Your Project Root

A screenshot of a computer

AI-generated content may be incorrect.

Commit and Push the azure-pipelines.yml to GitHub

A screenshot of a computer screen

AI-generated content may be incorrect.

Create the Pipeline in Azure DevOps (GitHub as Source)