Lab#4 – Azure DevOps CI/CD

**Due Date:** Midnight of Week#12 Sunday (Dec 1, 2024)

**Purpose:** The purpose of this lab is to help you become familiar with:

* Azure DevOps Artifacts
* Azure DevOps CI/CD

**Instructions**: Be sure to read the following general instructions carefully:

This lab should be completed individually by all students. Submit your solution **through the dropbox**, name your submission as **studentID(yourlastname)\_Labnumber.zip**.

e.g., **300123456**(**smith)\_Lab#4**.zip

**Rubric**

|  |  |
| --- | --- |
| Question | Marks |
| 1. Artifacts in Azure DevOps | 5 |
| 1. Continuous Integration/Continuous Delivery | 10 |

Question 1 – Artifacts in Azure DevOps [ 5 marks]

1. Implement a C# class library project which includes an extension method for class string to check whether a string starts with a upper case letter or not (refer to <https://learn.microsoft.com/en-us/dotnet/core/tutorials/library-with-visual-studio?pivots=dotnet-8-0> ), please make sure to use .NET 8
2. Use GitHub as SCM to manage your project
3. Create a public Azure DevOps project for the GitHub repo created in step 2
4. Define a CI pipeline to build Nuget package, pack and push it to Azure Artifacts Feed
5. Store your CI pipeline definition to GitHub repo so that it can be tracked and versioned
6. Please make sure that Self-hosted agent is used in your CI pipeline
7. Implement a console app to consume your own NuGet package
8. Commit all source code changes to GitHub

After you finish all the steps above, include the **URL of your GitHub repo** and **Azure DevOps project, please make sure that the project is public**. Include screenshots for following into your submission doc file

1. CI definition yaml file in GitHub repo [0.5 mark]
2. Screenshot of pipeline run shows that the pipeline successfully. In other words, every step succeeds [1 mark]
3. Artifact Feed UI showing NuGet package [0.5 mark]
4. Visual Studio 2022 showing NuGet library is downloaded from Azure DevOps Artifact Feed [2 mark]
5. Command window showing successful run of the Console App with or without uppercase string [0.5\*2 mark]

Question 2 – Continuous Integration & Continuous Delivery in Azure DevOps [10 marks]

1. ***Continue Q2 of Lab 3***, and finish first 3 parts (i.e., coding), and put code to ***Azure repo***
2. Create Azure DevOps project for your Web app
3. Create CD release pipeline based on the artifact generated by your CI pipeline, and your CD pipeline should be automatically triggered whenever there is new build is available
4. Upon deployment, your asp.net core web application should be able to run on your machine

After you finish all of those steps above, include **Azure DevOps project** (**please make sure that the project is public**) as well as **screenshots** for following into your submission doc file

1. Self-hosted agent has been set up [0.5 mark]
2. A Deployment group has been set up properly [1 mark]
3. Azure DevOps public project [0.5 mark]
4. Azure repo for C# code, and corresponding work item(s)’ status in Azure DevOps Boards [2 marks]
5. Store Azure DevOps CI pipeline definition using self-hosted agent (yaml file) [0.5 mark]
6. Azure DevOps release pipeline (yaml file) (2 marks)
7. Pipeline log showing that the application has been successfully deployed to the deployment group [1 mark]
8. Create a release [0.5 mark]
9. Provide screenshot to show that the released application can run properly [2 marks]