**CS 504 – Software Engineering**

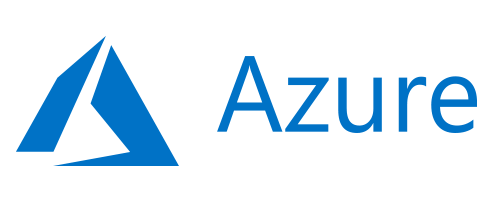
**HOP10C – Blazor Application – Continuous Integration & Continuous Deployment (CI/CD)**

12/04/2020 Developed by Kim Nguyen

12/04/2020 Reviewed by Yu-Che Liu

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

Icon

Description automatically generated 

**Before You Start**

* Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please choose the stable release rather than beta-testing version.
* This tutorial targets Windows users and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgement while going through this cookbook style tutorial to complete each step.
* For your working directory, use your course number. This tutorial may use a different course number as an example.
* The directory path shown in screenshots may be different from yours.
* If you are not sure what to do or confused with any steps:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Implement Continuous Integration & Continuous Deployment (CI/CD) with Azure and Github.

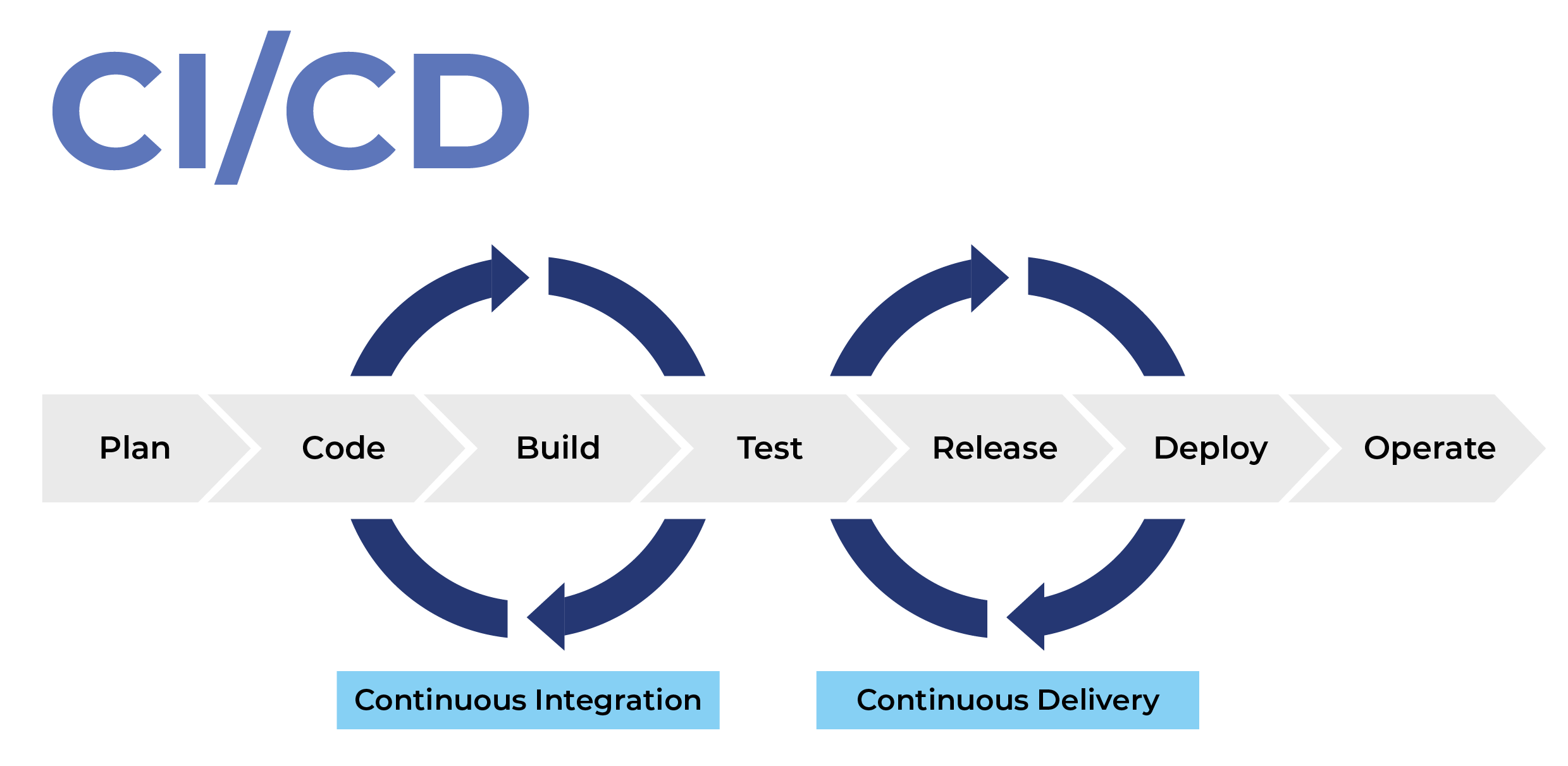
**Resources:**

* Microsoft | ASP .NET - <https://dotnet.microsoft.com/apps/aspnet>

**Continuous Deployment (CI/CD)**

Now our app is live, what happens if we want to make changes to the app and publish them? Do we have to go through the deployment process again? No! We can automate! CI/CD helps us to make this process easy.

In software engineering, CI/CD or CICD generally refers to the combined practices of continuous integration and either continuous delivery or continuous deployment. CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications.



Azure lets us automate CI/CD by allowing us to link the Github repo (that has the source code of our app) to the Azure Deployment Center. This way, whenever we push any changes to Github, they will automatically be synced to the live app, without us having to go through deployment process.

1. On Azure portal, in the left side menu, click on Deployment Center, you should see the following window:

Graphical user interface, application

Description automatically generated

In this class, we will use the GitHub option to intergrade CI/CD.

1. Click on Github option, then click “Authorize”:

Graphical user interface, text, application

Description automatically generated

1. A new window will pop up, click Authorize AzureAppService to grant Azure access to your Github repos:

Graphical user interface, text, application

Description automatically generated

1. Once done, you should see the Azure Deployment Center again, this time, Github option should show your Github username. Click Continue:

Graphical user interface, application

Description automatically generated

1. Choose Github Actions, then click Continue:

Graphical user interface, text, application, email

Description automatically generated

1. Fill out the information to find your repo.

Choose cityuseattle for the Organization

Choose your correct HOP10 repo for the Repository

Choose master for Branch

Then, click Continue

Graphical user interface, text, application, email

Description automatically generated

1. Check the Summary, to see if all the information is correct, then click Finish:

Graphical user interface, text

Description automatically generated

1. It will take couple of minutes to load the repo, once done, you should see the similar window:

Graphical user interface, text, application, email

Description automatically generated

1. Let’s test and see our real time changes. Go back to VSCode, under the project folder, under Pages folder, in Index.razor, add the following line:

Graphical user interface, text

Description automatically generated

1. Save changes, push them to GitHub:

Make sure you are in the right path, for example: /Desktop/CS504/HOP10-YourGitHubUsername/Module 10

Type the following command:

git add . (to copy all changes you have made)

git commit -m “Test CI/CD” (To add a message to your submission)

git push origin master (to upload your work to Github)

1. Go back to Azure Deployment Center, you should see our changes were updated automatically:

Graphical user interface, text, application, email

Description automatically generated

1. Go to your web app, reload, you should see the change shows up:

Graphical user interface, application

Description automatically generated

☺ Hope you enjoyed Full stack development and learned a thing or two! See you in other CityU courses! ☺

**Push your work to GitHub**

Open the terminal from the VSCode by hitting the control + ~ key, make sure you are in the right path, for example: /Desktop/CS504/HOP010-YourGitHubUsername/Module 10

Type the following command:

git add . (to copy all changes you have made)

git commit -m “Submission for Module 10 – Your Name” (To add a message to your submission)

git push origin master (to upload your work to Github)