Continuous Integration with Visual Studio Team Services

Lab Tasks

Task 1: Create a Web Application in VS

1. Complete Developing Microsoft Azure Infrastructure Solutions lab.

Task 2: Connect Web Application in VS[2] to a new Team Project

1. Open VS[3].
2. Open the Project created[2] in previous lab.
3. On the Solution Explorer, right click on the Solution.
4. Click on <Add Solution to Source Control…>.
5. Go to Team Explorer -> Home ( ).
6. Click on <Sync>.
7. Click on <Publish Git Repo>.
8. Select the account name.
9. Click on <Advanced> link.
10. Provide the Project name.
11. Provide the Repository name.
12. Click on <Publish repository>.
13. If prompted, sign in.
14. Click on <See it on web.> link from the yellow message indicating the Project was pushed.
15. It will open a window to the new project Dashboard.

Task 3: Create Work Items

1. Create Epic.
    - Click on Work -> Work Items.
    - Click on New Work Item -> Epic.
    - In the new window, provide the details.
        - Title.
        - Assigned to.
        - Description.
        - The rest of the fields are optional.
        - Click on <Save>.
    - Save the ID for future usage.
2. Create Feature.
    - Click on Work -> Work Items.
    - Click on New Work Item -> Feature.
    - In the new window, provide the details.
        - Title.
        - Assigned to.
        - Description.
        - In Related Work.
            - Click Add link -> Existing Item.
                - Link type: Child.
                - Work item to link: Enter the Epic Id.
                - Comment: Optional.

- Click on <OK>.
- The rest of the fields are optional.
- Click on <Save>.
- Save the ID for future usage.

3. Create User Story.
    - Click on Work-> Work Items.
    - Click on New Work Item -> User Story.
    - In the new window, provide the details.
        - Title.
        - Assigned to.
        - Description.
        - In Related Work.
            - Click Add link -> Existing Item.
                - Link type: Child.
                - Work item to link: Enter the Feature Id.
                - Comment: Optional.
                - Click on <OK>.
        - The rest of the fields are optional.
    - Click on <Save>.
4. Create Task.
    - Go to Work -> Backlogs.
    - Click on <+> beside the User Story created previously.
    - Click on <Task>.
    - In the new window, provide the details.
        - Title.
        - Assigned to.
        - Description.
        - Assign Original Estimate in the Effort (Hours).
        - The rest of the fields are optional.
        - Check the Task is linked to the user story.
        - Click Save & Close.

Task 4. Create Continuous Integration Build

1. Click on Pipelines -> Builds at the top menu of the page.
2. Create a new pipeline.
    - Click on <New pipeline>.
    - Under the Select sources you need to make sure to select Azure Repos Git.
    - In the Team project select the one you created.
    - In the Repository select the project you created.
    - In the Default branch, select Master.
    - Click on <Continue>.
3. Select the <Empty pipeline> template.
    - Click on <Apply>.
4. Once the new empty build has been created.

- Modify the Name[1].
- Select <Hosted VS2017> in the Agent pool.

5. To start adding tasks.
    - Click on <+> beside the Agent job1 on the left panel.
6. Add tasks.
    - Select the Tool tab.
    - Select <NuGet Tool Installer> task.
    - Click on <Add>.
    - Select the Package tab.
    - Select <NuGet> task.
    - Click on <Add>.
    - Select the Build tab.
    - Select <Visual Studio Build> task.
    - Click on <Add>.
    - Select <Index Sources & Publish symbols> task.
    - Click on <Add>.
    - Select the Utility tab.
    - Select <Publish Build Artifacts> task.
    - Click on <Add>.
7. Modify <NuGet restore> task.
    - Select <NuGet restore> task from the left panel.
    - Click on <… > from the Path to solution.
    - Select the .sln solution.
    - Click on <OK>.
8. Modify <Build solucion ***.sln> task.
    - Select <Build solucion ***.sln> task from the left panel.
    - Click on <…> from Solution.
    - Select the .sln solution.
    - Click on <OK>.
    - Add the next arguments in MSBuild Arguments

/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true
/p:SkipInvalidConfigurations=true
/p:DesktopBuildPackageLocation="$(build.artifactstagingdirectory)\WebApp.zip"
/p:DeployIisAppPath="Default Web Site"

    - Add the next variable in Platform

$(BuildPlatform)

    - Add the next variable in Configuration

$(BuildConfiguration)

9. Modify <Publish symbols path> task.
    - Disable Publish symbols
    - Disable Continue on error, on the Control Options section.
10. Modify <Publish Artifact> task.
    - Change the Path to publish.

$(build.ArtifactStagingDirectory)

    - Change the Artifact name to <drop>.
11. Add variables.

- Click on the Variables tab.
- Click on <+Add>.
    - Name: BuildConfiguration.
    - Value: release.
    - Settable at queue time: enabled.
- Click on <+Add>.
    - Name: BuildPlatform.
    - Value: any cpu.
    - Settable at queue time: enabled.
- Enable the Settable at queue time to the variable system.debug

12. Modify triggers.
    - Click on the Triggers tab.
    - Check Enable continuous integration (CI) option for your project to build the solution every time a change is pushed.
    - Make sure the filter includes the appropriate branch, in this case master.
    - Uncheck Batch changes while a build is in progress option.
13. Click <Save & queue> from the top menu of the Pipeline.
14. Click on <Save> from the list.
    - Add a Comment.
    - Click on <Save>.

Task 5: Test the CI trigger in VSTS

1. Go to your solution in VS.
2. Navigate to SolutionName -> ProjectName -> Controllers.
3. Open HomeController.cs.
4. Add after the using section a comment (Example: //CI Test).
5. Save the file (File->Save).
6. In the Solution Explorer right click on the solution.
7. Click on <Commit…>.
8. In the Team Explorer add a comment.
9. Make sure the controller where added the comment is listed in Changes.
10. Click on <Commit All>.
11. From the home in the Team Explorer or the commit message.
    - Click on <Sync> to upload the changes in the server.
    - Click on <Push> from the Outgoing commits list.
12. Go back to VSTS.
    - Click on Pipelines -> Build menu.
    - The previous push should have triggered the build we previously created.
13. Click on <…> from the new build that has been created.
    - Click on <View build results> from the list.
    - You should see progress of the build.
    - Here you can also see the commands being logged to console and the current steps that the build is on.
14. You can see in the Artifacts menu, there is the drop folder with the output configured (the build project).

[1] Optional.
[2] You will use the application created in the <Developing Microsoft Azure Infrastructure Solutions> labs, module 3.
[3] VS is installed in the VM created in the <Developing Microsoft Azure Infrastructure Solutions> lab, module 2.

References

- https://msdn.microsoft.com/en-us/library/ms181374(v=vs.80).aspx
- http://microsoft.github.io/PartsUnlimited/cicd/200.3x-CICD-M01-CIwithVSTS.html