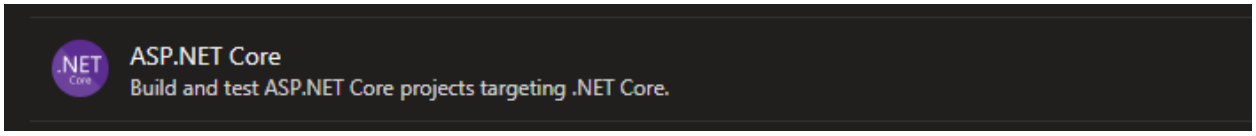
1- Build using

    dotnet run --project DevConf2020

2- Launch Project - localhost:5000
3- Go to Azure DevOps, mark User Story Create Pipeline as Active
4- Add Build Pipeline and select repo DevConf2020
5- Select ASP.NET Core under configuration and build



6- Build application and look for result
7- Pull latest change locally and replace YAML by - Insert **Snippet1**

```yaml
# ASP.NET Core
# Build and test ASP.NET Core projects targeting .NET Core.
# Add steps that run tests, create a NuGet package, deploy, and more:
# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:
- '*'

pool:
  vmImage: 'ubuntu-latest'

variables:
  buildConfiguration: 'Release'

steps:
- task: DotNetCoreCLI@2
  displayName: 'Restore project dependencies'
  inputs:
    command: 'restore'
    projects: '**/*.csproj'

- task: DotNetCoreCLI@2
  displayName: 'Build the project - Release'
  inputs:
    command: 'build'
    arguments: '--no-restore --configuration Release'
    projects: '**/*.csproj'
```

8- Push and look at build on Azure DevOps
9- Insert **Snippet2**

```yaml
- task: DotNetCoreCLI@2
```

```
  displayName: 'Publish the project - Release'
  inputs:
    command: 'publish'
    projects: '**/*.csproj'
    publishWebProjects: false
    arguments: '--no-build --configuration Release --
output $(Build.ArtifactStagingDirectory)/Release'
    zipAfterPublish: true

- task: PublishBuildArtifacts@1
  displayName: 'Publish Artifact: drop'
  condition: succeeded()
```

10- Push and look at build on Azure DevOps

11- Checkout **unit-test**

12- Run **dotnet test --configuration Release**

13- Insert **Snippet3** after Dotnet Publish

```
- task: DotNetCoreCLI@2
  displayName: 'Run unit tests - $(buildConfiguration)'
  inputs:
    command: 'test'
    arguments: '--no-build --configuration $(buildConfiguration)'
    publishTestResults: true
    projects: '**/*.Test.csproj'
```

14- Commit and push – for failed test

15- Edit DevConf2020/Data/WeatherForecastService.cs to map 10 days

16- Commit and Push – Passing test

17- Checkout **Release-pipeline**

18- Yaml – Remove pool section

19- Add below **AFTER** Variables and indent following steps (same line as pool) - Insert **Snippet4**

```
stages:
  - stage: 'Build'
    displayName: 'Build the web application'
    jobs:
    - job: 'Build'
      displayName: 'Build job'
      pool:
        vmImage: 'ubuntu-latest'
```

20- Add below at the end - Insert **Snippet5**

```
  - stage: 'Deploy'

  displayName: 'Deploy the web application'
  dependsOn: Build
```

```
jobs:
- deployment: Deploy
  pool:
    vmImage: 'ubuntu-latest'
  environment: dev
  strategy:
    runOnce:
      deploy:
        steps:
        - download: current
          artifact: drop
```

21- Push and look at build on Azure DevOps

22- Checkout **Release-pipeline-createAZResource**

23- Add below after artifact: drop – Insert **Snippet 6**

```
            - task: AzurePowerShell@5
              inputs:
                azureSubscription: 'DevConf2020Azure'
                ScriptType: 'InlineScript'
                Inline: 'New-AzResourceGroupDeployment -
ResourceGroupName DevConf2020 -
TemplateUri https://raw.githubusercontent.com/whosenbocus/DevConf2020/release
-pipeline-createAZResource/DevConfAzARM.json -
TemplateParameterUri https://raw.githubusercontent.com/whosenbocus/DevConf202
0/release-pipeline-createAZResource/DevConfAzARM.parameters.json'
                azurePowerShellVersion: 'LatestVersion'
```

24- Push and look at build on Azure DevOps

25- Navigate to - https://devconf2020.azurewebsites.net/

26- Add below at end of file – insert **snippet 7**

```
            - task: AzureWebApp@1
              inputs:
                azureSubscription: 'DevConf2020Azure'
                appType: 'webApp'
                appName: 'DevConf2020'
                package: '$(Agent.WorkFolder)/**/DevConf2020.zip'
                deploymentMethod: 'auto'
```

27- Push and look at build on Azure DevOps

28- Checkout **telemetry**

29- Under Azure DevOps. Click on Library -> Click on + Variable Group

   a. Set the Variable group name as "**Telemetry**"
   b. Select "Allow access to all pipelines"
   c. Under Variables, click on "+ Add"
      i. Set Name as "telemetryKey"

ii. Set Value as => (Instrumentation Key from DevConf2020 Application Insights)

d. Click on Save

30- Add below snippet below **environment: dev** – insert **Snippet 8**

```
      variables:
      - group: Telemetry
```

31- Add snippet end of script – insert **Snippet 9**

```
- task: AzureAppServiceSettings@1
              inputs:
                azureSubscription: 'DevConf2020Azure'
                appName: 'DevConf2020'
                resourceGroupName: 'DevConf2020'
                appSettings: |
                  [
                    {
                      "name": "APPINSIGHTS_INSTRUMENTATIONKEY",
                      "value": "$(telemetryKey)",
                      "slotSetting": false
                    }
                  ]
```

32- Push and look at build on Azure DevOps and AI on Azure

33- Checkout **Final**

34- Edit DevConf2020 > Pages > Index.razor. Replace “Hello, World” by “Hello, Dev Conf 2020”

35- Push and look at build on Azure DevOps and refresh page