AzureDevOps Pipelines

Basic structure and essential parts of a pipeline

Templates?

trigger:

  - main

parameters:

- name: buildConfig

  displayName:

  type:

  default:

  values:

    - debug

    - release

jobs:

  - job: job1

    displayName:

    pool:

      name:

      demands:

    steps:

    - checkout: self

    - task:

      displayName:

      inputs:

        command:

        projects:

        arguments:

    - task:

      displayName:

      inputs:

  - job: job2

  - deployment: run\_test

    dependsOn: job2

    pool:

      name:

      demands:

    environment:

    strategy:

      runOnce:

        deploy:

          steps:

          - task: DownloadBuildArtifacts@0

            inputs:

              buildType:

              downloadType:

              artifactName:

              downloadPath:

          - task:

          - task:

Pipeline to run specflow tests on a Windows agent (without docker) and publish to Living Doc

trigger:

  - main

pool:

  name: QuantitativeTrading

  demands: agent.os -equals Windows\_NT

parameters:

- name: BuildConfiguration

  displayName: Build configuration

  type: string

  default: Release

  values:

    - Debug

    - Release

jobs:

- job: Build\_E2E\_tests

  displayName: Build E2E tests

  pool:

    name: QuantitativeTrading

    demands: agent.os -equals Windows\_NT

  steps:

  - checkout: self

    clean: true

  - task: DotNetCoreCLI@2

    displayName: Restore

    inputs:

      command: restore

      projects: '\*\*/Compass.E2E.Spec.csproj'

  - task: DotNetCoreCLI@2

    displayName: Build

    inputs:

      command: build

      projects: '\*\*/Compass.E2E.Spec.csproj'

      arguments: --configuration ${{parameters.BuildConfiguration}}

  - task: DotNetCoreCLI@2

    displayName: Publish

    inputs:

     command: publish

     publishWebProjects: false

     projects: '\*\*/Compass.E2E.Spec.csproj'

     arguments: --configuration ${{parameters.BuildConfiguration}} --output $(build.artifactstagingdirectory)  -r win10-x64

     zipAfterPublish: false

     modifyOutputPath: false

  - task: PublishBuildArtifacts@1

    displayName: Publish Artifact

    inputs:

      ArtifactName: 'compass\_e2e\_build\_artifact'

      PathtoPublish: $(Build.ArtifactStagingDirectory)

- job: Run\_E2E\_Tests

  dependsOn: Build\_E2E\_tests

  displayName: Run E2E Tests

  pool:

    name: QuantitativeTrading

    demands: agent.os -equals Windows\_NT

  steps:

  - task: DownloadBuildArtifacts@0

    displayName: Download Build Artifacts

    inputs:

      buildType: 'current'

      downloadType: 'single'

      artifactName: 'compass\_e2e\_build\_artifact'

      downloadPath: 'D:\Applications\E2E\'

  - task: PowerShell@2

    displayName: Show folder contents

    inputs:

      targetType: inline

      script: |

        echo 'pwd below:'

        pwd

        echo 'current folder content'

        ls .

        echo 'list D:\Applications\E2E\compass\_e2e\_build\_artifact below:'

        ls D:\Applications\E2E\compass\_e2e\_build\_artifact

  - task: PowerShell@2

    displayName: install Playwright

    inputs:

      targetType: inline

      script: |

        cd D:/Applications/E2E/compass\_e2e\_build\_artifact/.playwright/node/win32\_x64/

        ./playwright.cmd install

  - task: DotNetCoreCLI@2

    displayName: Run E2E tests

    inputs:

      command: 'test'

      workingDirectory: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\'

      projects: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\Compass.E2E.Spec.dll'

  - task: SpecFlowPlus@0

    inputs:

      generatorSource: 'TestAssembly'

      testAssemblyFilePath: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\Compass.E2E.Spec.dll'

      testExecutionJson: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\TestExecution.json'

How to specify repo?

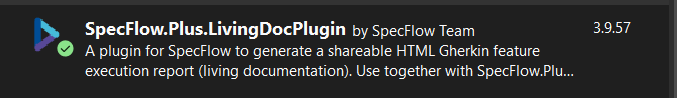
When creating a new pipeline, AzureDevOps requires the user to choose a repo to start with.

## Pwsh VS Powershell@2

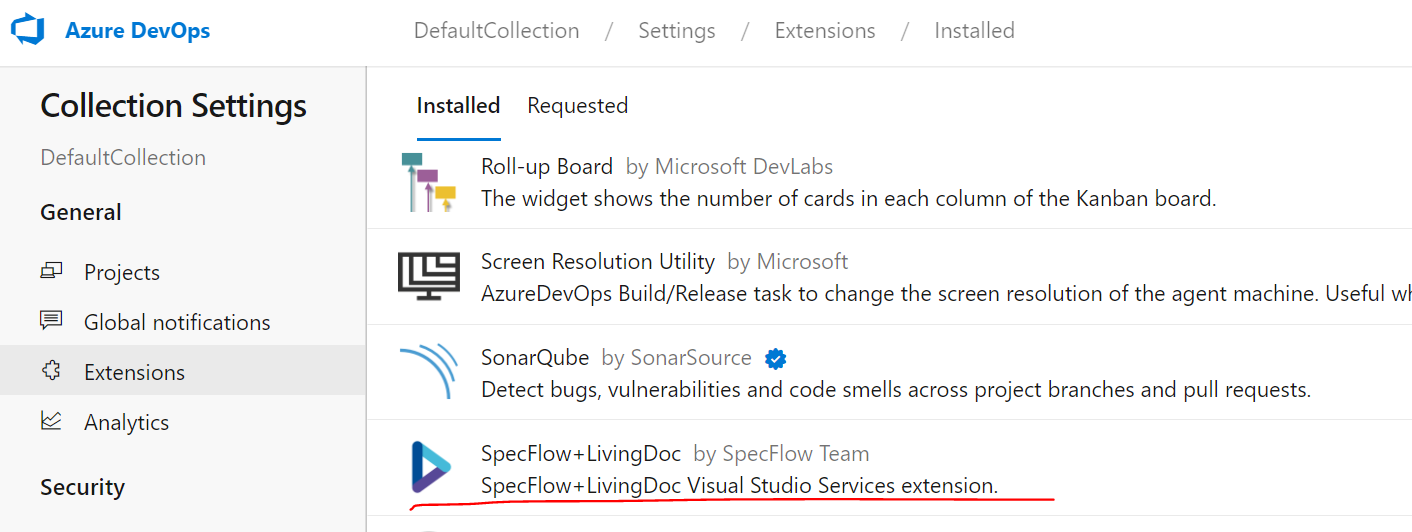
Pwsh is the new Powershell Core, designed to be cross platform (the future 😉), however some build agents may not have it installed., in which case, use Powershell for windows build agent and bash for linux

## Specflow Livingdoc

Needs 3 things:

1. Visual Studio nuget package installed

2. AzureDevOps livingDoc extension installed



3. SpecFlowPlus task in the pipeline:

  - task: SpecFlowPlus@0

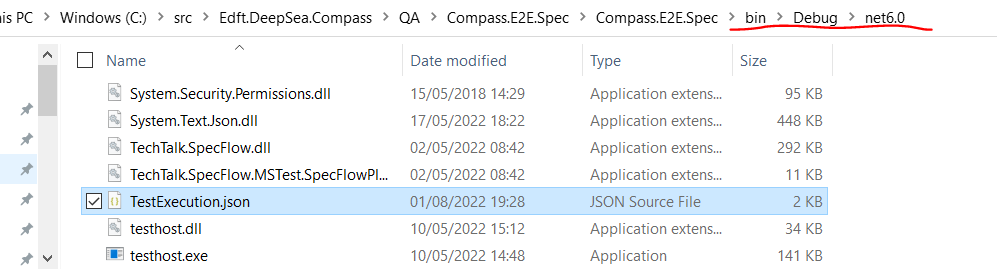
    inputs:

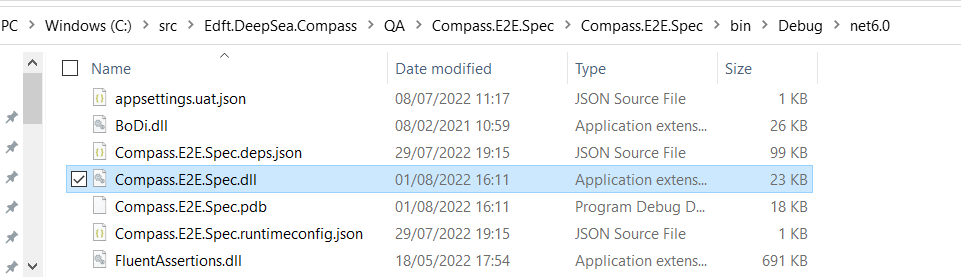
      generatorSource: 'TestAssembly'

      testAssemblyFilePath: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\Compass.E2E.Spec.dll'

      testExecutionJson: 'D:\Applications\E2E\compass\_e2e\_build\_artifact\TestExecution.json'

need to specify both testAssembly and testExecutionJson

Both of them are in the build artifact folder, i.e.: locally:



In AzureDevOps, the testAssembly is in the artifact folder produced from “dotnet build” process (by default, part of dotnet test). The testExecution is produced by the “dotnet test” after test execution in the working folder, by default, where the testAssembly is.