



A quick explanation of what you got there:

* **trigger**: This is the event that will trigger the pipeline. In this case, it’s set to trigger on any push to the main branch.
* **pool**: A pool is a collection of agents that can run your pipeline tasks. In this case, it’s set to use agents based on the latest Ubuntu version in the hosted pool provided by Azure Pipelines.
* **variables**: This is where you can define variables that will be used across the pipeline. The build configuration is defined as a variable by default.
* **steps**: This is where you define the steps that will be run in the pipeline. By default, you get a single script step that builds your application in the Release configuration.

# NAME THE PIPELINE SAME AS FILE (WITHOUT ".yml")

resources:

  repositories:

    - repository: self

      trigger:

       - main

variables:

  azureServiceConnection: 'azure subs'

  subscriptionId: '86fcde80-267c-4d3b-8f28-bec95924273f'

  resourceGroup: 'AZ400-RG153681980'

  location: 'westus2'

stages:

- stage: Build

  jobs:

  - job: Build

    pool:

      vmImage: 'ubuntu-latest'

    steps:

    - task: PublishPipelineArtifact@1

      inputs:

        targetPath: 'AzureAppAccess/infra'

        artifact: 'infra'

    - task: AzureResourceManagerTemplateDeployment@3

      displayName: Deploy ACR using Bicep

      inputs:

        deploymentScope: 'Resource Group'

        azureResourceManagerConnection: $(azureServiceConnection)

        subscriptionId: $(subscriptionId)

        action: 'Create Or Update Resource Group'

        resourceGroupName: '$(resourceGroup)'

        location: '$(location)'

        templateLocation: 'Linked artifact'

        csmFile: 'AzureAppAccess/infra/acr.bicep'

        deploymentArtifact: 'infra'

        deploymentMode: 'Incremental'

        deploymentOutputs: 'outputJson'

    - task: PowerShell@2

      displayName: Parse Bicep Output

      inputs:

        targetType: 'inline'

        script: |

          ls -l $(Build.SourcesDirectory)

          ls -l $(Build.SourcesDirectory)/AzureAppAccess

          $var=ConvertFrom-Json '$(outputJson)'

          $value=$var.acrLoginServer.value

          Write-Host "##vso[task.setvariable variable=acrLoginServer;]$value"

    - task: Docker@1

      displayName: 'Build the docker image'

      inputs:

        azureSubscription: $(azureServiceConnection)

        azureContainerRegistry: $(acrLoginServer)

        dockerFile: 'AzureAppAccess/Dockerfile'

        defaultContext: false

        context: $(Build.SourcesDirectory)

        includeLatestTag: true

        imageName: eshoponweb/web:$(Build.BuildId)

    - task: PowerShell@2

      displayName: 'Log image name before push'

      inputs:

        targetType: 'inline'

        script: |

          echo "Pushing image: $(acrLoginServer)/eshoponweb/web:$(Build.BuildId)"

    - task: Docker@1

      displayName: 'Push the docker images'

      inputs:

        azureSubscription: $(azureServiceConnection)

        azureContainerRegistry: $(acrLoginServer)

        action: 'Push an image'

        imageName: eshoponweb/web:$(Build.BuildId)

        includeLatestTag: true