**AZURE FUNCTION – CICD Pipeline for PowerShell – Azure Portal(*Using ARM template*)**

As we are aware that we can create CD/CD pipeline and deploy Azure function using 2 ways.

1. Using Azure Function created in Azure Portal
2. Using Azure Resource Manager and ARM Template(Created in VS Code)

Here we will be discussing on First approach.

**Note:**In the first method, we have 2 approach to build pipeline

1. We can build Pipeline using default template provided while creating new Pipeline
2. We can build Pipeline using ARM template and can use for deployment (below steps shown).

**Steps:**

1. Navigate to Azure Portal > Type Function App > Create

Graphical user interface, text, application, email

Description automatically generated

1. Provide details for creating new function App

Graphical user interface, text, application, email

Description automatically generated

1. One Function App is created then create function from left navigation

Graphical user interface, text, application, email

Description automatically generated

1. Once HTTP function is created > Click on Overview of the function App.
2. Click on download App content and also check “include app settings”

Graphical user interface, text

Description automatically generated with medium confidence

1. Once downloaded, unzip all files in any folder.
2. Navigate to DevOps site > create a repository and push all your local files just downloaded.

Graphical user interface, text

Description automatically generated

1. Click on Pipeline > New Pipeline > Use Classic Editor
2. Select the Repo we just uploaded > Continue
3. For Select Template > Select Empty Job
4. Add Task as PowerShell

Graphical user interface, application, Teams

Description automatically generated

For the Script, type

if (Test-Path "extensions.csproj") {

     dotnet build extensions.csproj --output ./bin

}

And for Type should be Inline

Graphical user interface, text, application, email

Description automatically generated

1. Add New Task as “Archive Task”,

For root folder >  $(System.DefaultWorkingDirectory)

Uncheck > Prepend

Archive Type >  ZIP

Archive file to create > $(Build.ArtifactStagingDirectory)/$(Build.BuildId).zip

Graphical user interface, text, application, email

Description automatically generated

1. Add new task as , “Publish Artifect: Drop”

Path to Publish > $(Build.ArtifactStagingDirectory)

Artifact name > drop

Artifact publish location > Azure Pipeline

Graphical user interface, text, application, email

Description automatically generated

1. Save and Queue > Run Pipeline
2. If you want to consume in Release, Create new release and select from the build.

**Deployment/Release Pipeline:**

If you want to create release for deployment of Azure function then you should select release template accordingly.

For Azure Function:

Graphical user interface, text, application, email

Description automatically generated

For Web Jobs:

Graphical user interface, text, application, email

Description automatically generated