Azure Function App

Graphical user interface, application, Word

Description automatically generated

When create Function App from Azure Portal, you need to specify the Storage Account Connection String

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

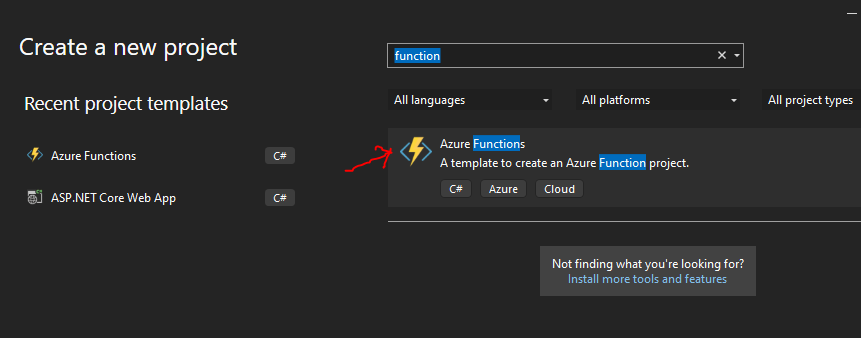
Description automatically generated

* + - * Storage Account Connection String can be found here:

Graphical user interface, text, application, email

Description automatically generated

* There is a project template called “Azure Functions”



You can pick Authentication Level

Graphical user interface, application

Description automatically generated

* Azure Function usually has a triger (by HTTP Request or Message Queue etc.)

A screenshot of a computer

Description automatically generated with medium confidence

Or

Chart

Description automatically generated with low confidence

* Publish through Visual Studio
* Create Azure Function App first

Graphical user interface, text, application, email

Description automatically generated

* Then use Publish to Azure directly

Graphical user interface, text, application

Description automatically generated

* Publish through Power Shell
* Go to Function App’ s bin folder where it contains the .dll and host.json file

Text

Description automatically generated

A picture containing text

Description automatically generated

* Use CICD pipeline to deploy

Graphical user interface, text, application

Description automatically generated

CI (Build) YAML file

|  |
| --- |
| pool:    name: 'Default'  steps:  - script: |      dotnet restore      dotnet build --configuration Release  - task: DotNetCoreCLI@2    inputs:      command: publish      arguments: '--configuration Release --output publish\_output'      projects: '\*.csproj'      publishWebProjects: false      modifyOutputPath: false      zipAfterPublish: false  - task: ArchiveFiles@2    displayName: "Archive files"    inputs:      rootFolderOrFile: "$(System.DefaultWorkingDirectory)/publish\_output"      includeRootFolder: false      archiveFile: "$(System.DefaultWorkingDirectory)/build$(Build.BuildId).zip"  - task: PublishBuildArtifacts@1    inputs:      PathtoPublish: '$(System.DefaultWorkingDirectory)/build$(Build.BuildId).zip'      artifactName: 'drop' |