**Q1 - SCENARIO**

A car rental company called FastCarz has a .net Web Application and Web API which are recently migrated from on-premise system to Azure cloud using Azure Web App Service

and Web API Service.

The on-premises system had 3 environments Dev, QA and Prod.

The code repository was maintained in TFS and moved to Azure GIT now. The TFS has daily builds which triggers every night which build the solution and copy the build package to drop folder.

deployments were done to the respective environment manually. The customer is planning to setup Azure DevOps service for below requirements:

1. The build should trigger as soon as anyone in the dev team checks in code to master branch.

trigger:  
  branches:  
    include:  
      - master

* Once the pull request is completed and code is merged into the master branch, above steps in the yaml file will trigger the build, if we want to trigger the build for another branch, we need to *add the branch name.*

2) There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.

The build should not be successful if any test fails.

* The below task used to install the version of .net core project in the agent

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '3.0.x'

* + The below steps used to restore all the packages available in the .csproj files, this step helps us to build the solution

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

projects: '\*\*/\*.csproj'

* + The below tasks builds the solution

- task: DotNetCoreCLI@2

inputs:

command: build  
 projects: '\*\*/web-api.csproj'

* + The below task unit tests the solution, this task will only complete once all the unit tests are passed

- task: DotNetCoreCLI@2

inputs:

command: 'test'

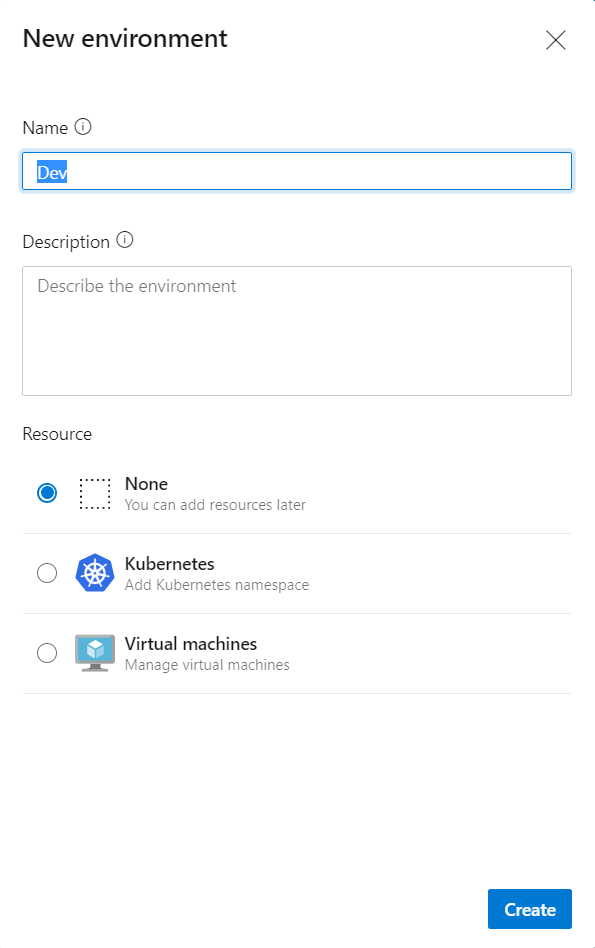
projects: '\*\*/web-api-tests.csproj’

if want to continue the next steps incase of failure at any task, we should set continueonError is set to true, by default it is set as false.

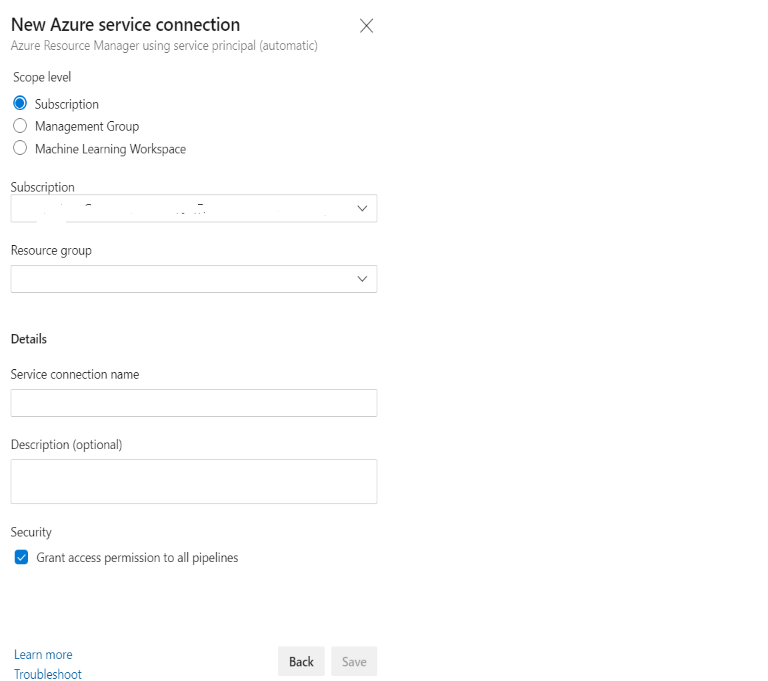
1. *The deployment of code and artifacts should be automated to Dev environment.*

* Upload the Artifacts from the Agents to Azure DevOps using below task  
   - publish: $(Build.ArtifactStagingDirectory)

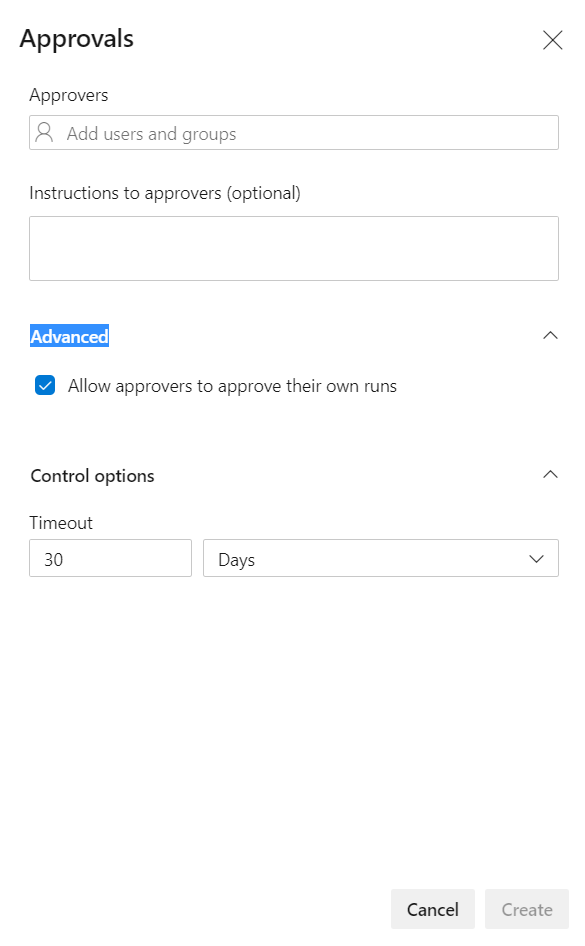
            artifact: drop

* Create an environment in the Azure DevOps, since we are using PAAS services we have to choose the resource as none  
    
  
* As we want continuous deployment to the Dev Environment, we should not set any approvers
* Create Service Connection for Development subscription and resources and use the service connection for deploying to the App Service

1. *Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.*

* Create QA and Prod environments and Stages to deploy the code to the respective environment
* Create QA and Prod Service Connections and use the service connections for the deployment   
  

1. *The deployments to QA and Prod should be enabled with Approvals from approvers only.*

* After Creating the environment, go to approvals and checks 🡪 approvers give the approvers names, in advanced choose the number of approvers is required to approve the deployment to stage.  
    
  

Explain how each of the above the requirements will be met using Azure DevOps configuration.

Explain the steps with configuration details.