**Devops Test**

**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.

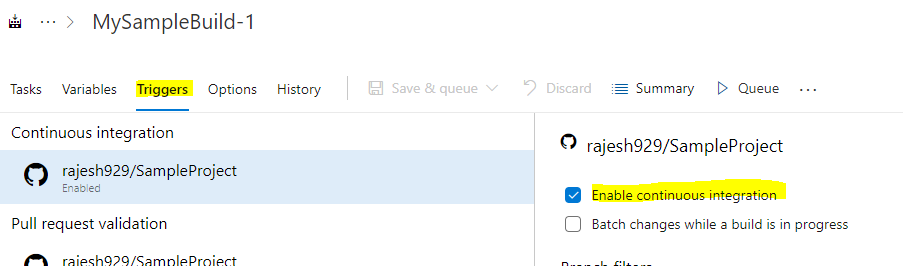
The on-premises system had 3 environments Dev, QA and Prod.

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 Pipeline service for below requirements:

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

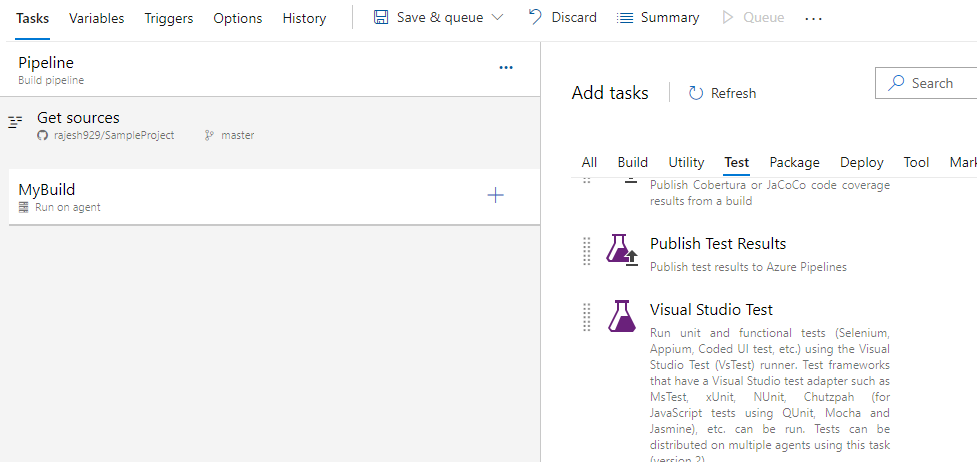
* This can be achieved by configuring Continuous integration trigger in the build on the master branch.



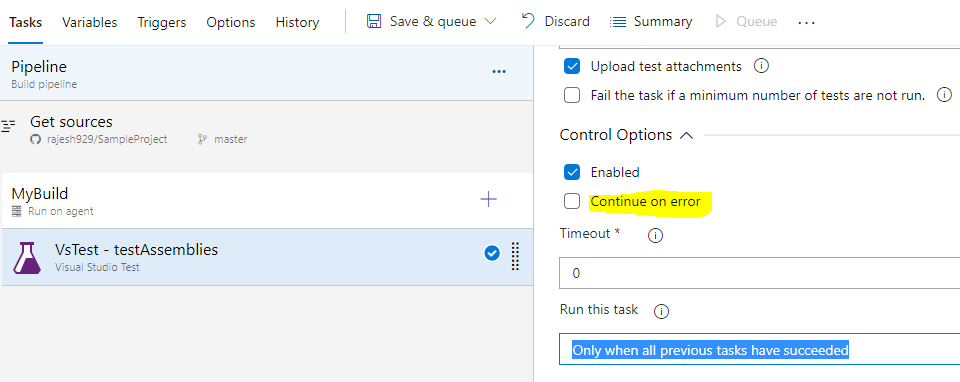
1. There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.

The build should not be successful if any test fails.

* This can be done by using visuals studio test tasks in the pipeline.

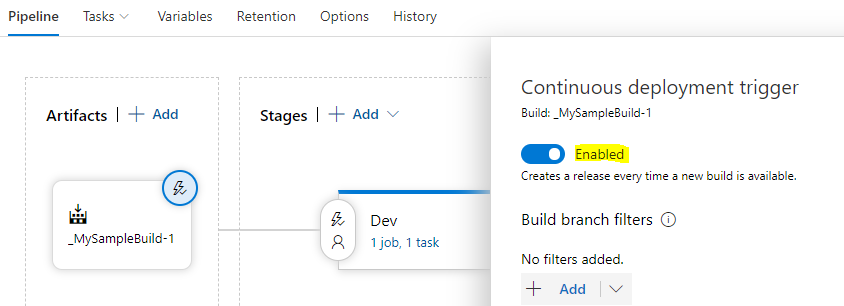


* We need to have three build tasks to build each project for Web, API and Test
* If any of the task fails the build will be failed if we want to stop the we can disable error on continue condition or stop the build for any task failures.



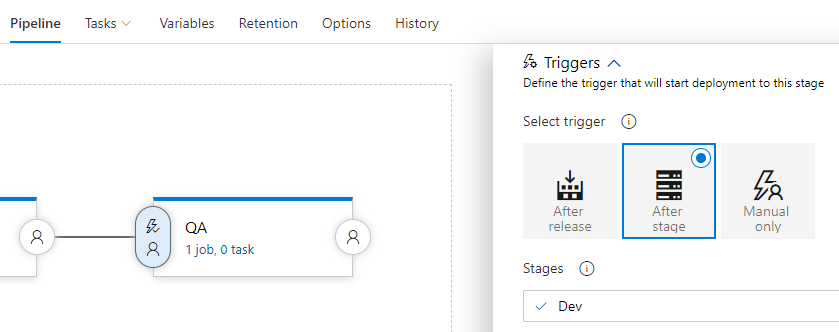
1. The deployment of code and artifacts should be automated to Dev environment.

- For this we need to configure a release pipeline with continuous deployment enabled on the artifacts and the dev stage trigger should be enabled as After Release.



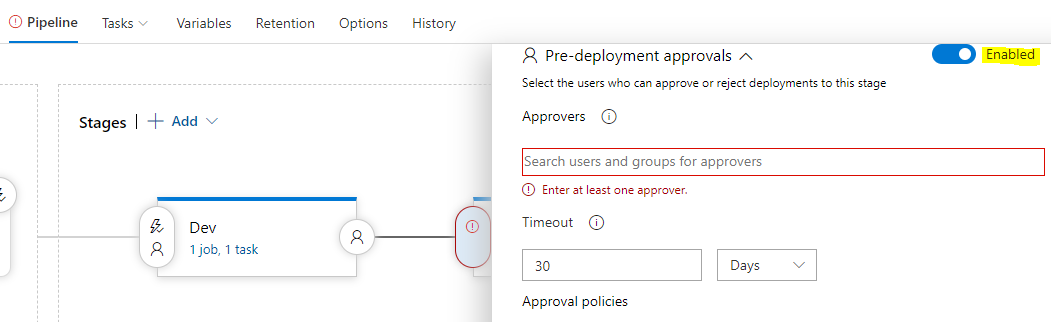
1. Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.

- This can be achieved by enabling the After Stage trigger on the QA and Prod stages.



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

* This can be achieved by enabling pre deployment approval on the stages.



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

Explain the steps with configuration details.

**Q2 - SCENARIO**

Macro Life, a healthcare company has recently setup the entire Network and Infrastructure on Azure.

The infrastructure has different components such as Virtual N/W, Subnets, NIC, IPs, NSG etc.

The IT team currently has developed PowerShell scripts to deploy each component where all the properties of each resource is set using PowerShell commands.

The business has realized that the PowerShell scripts are growing over period of time and difficult to handover when new admin onboards in the IT.

The IT team has now decided to move to Terraform based deployment of all resources to Azure.

All the passwords are stored in an Azure Service known as key Vault. The deployments need to be automated using Azure DevOps using IaC (Infrastructure as Code).

1. What are different artifacts you need to create - name of the artifacts and its purpose

- Terraform main file and variables file to deploy infrastructure

1. List the tools you will to create and store the Terraform templates.

- Visual Studio Code to create terraform template

- Git bash to commit the changes into Azure repos

- Powershell to validate the terraform templates

1. Explain the process and steps to create automated deployment pipeline.

- We will create a build pipeline source as the branch where we have the terraform templates

- In the pipeline we will add terraform task to initiate, plan and deploy the infrastructure.

- To automate the deployment, we will enable Continuous trigger on the pipeline

1. Create a sample Terraform template you will use to deploy Below services:

Vnet

2 Subnet

NSG to open port 80 and 443

1 Window VM in each subnet

1 Storage account

* Attached the terraform template. 

1. Explain how will you access the password stored in Key Vault and use it as Admin Password in the VM Terraform template.

* We will link the Azure Keyvault to the variable groups and link this variable group to the pipeline.
* We will declare variables in terraform template and those variables will be replaced from the pipeline variables.