**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.

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.

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

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

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

**Implementation:**

I am assuming we have web app created in Azure, with ASP .NET runtime stack.

Steps to Create Web app.

Azure portal:

1. Create a new resource search for Web app.

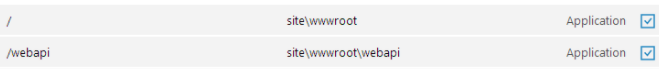
2. In create web app window select the subscription, resource group, Enter a name ex: FastCarz, publish type code, Runtime stack ASP .NET, Region, Select plan as per need, Operating system windows.

3. Enable up App insights monitoring.

4. Create web app.

5. Go to Configuration of webapp, go to path mapping add additional mapping for API under virtual application and directory.

site\wwwroot\webapi

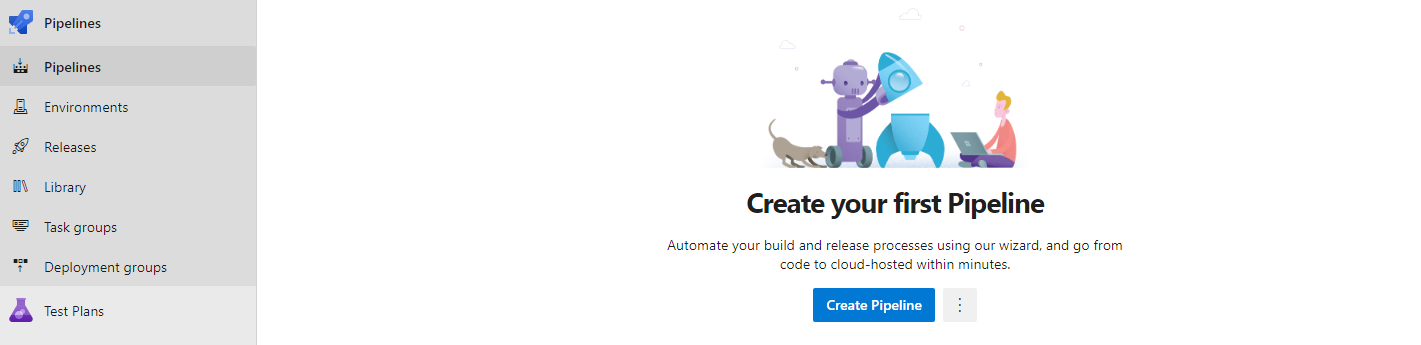


Setting Up build pipeline:

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

Explain the steps with configuration details.

Firstly we need to create new pipeline in azue devops.



Name the pipeline and In the Get sources I will select the project, repo and branch from the Azure Repos git source, will check Report build status so that we enable badge on source repository to indicate whether the build succeeded or failed.

Setup a Agent job, will select the option inherit from pipeline.

under the Agent will start adding jobs

1. First task will be Nuget Restore, will select the solution file and set the nuget path .nuget/nuget.config

2. Will include git version job to set the semantic versioning for jobs.

3. Visual studio solution to build select the solution file and select latest from visual studio version. if you additional arguments for MS build you can pass and platform win32, win 64 also you can specify the build config

4. **Add a task to run visual studio test with test assembly test file with below config**

**\*\*\\*test\*.dll**

**!\*\*\obj\\*\***

you can keep the serach folder ot default working directory

and also you can push results to testresult folder if needed.

5. And we can add the versioning for build that you get from sem version using powershell.

sample code is below

[string]$Year = get-date -format yy

[string]$DayOfYear = (Get-Date).DayofYear

$version = @"

$(GitVersion.SemVer)

"@

$subVersion = $version.Remove(1,1)

$newVersion = $subVersion.Replace('0-unstable.','dev.')

$format = $Year+"."+$DayOfYear+"."+$newVersion

#$format =$version+"."+'02'+"."+$DayOfYear+"."+$Year

Write-Host $format

Write-Verbose -Verbose "##vso[build.updatebuildnumber]$format"

get the date and day of the year store string variable get the git version form the Semver whichc is created from step two format it to it with year.dayofyear.newversion

6. If you have any dependent file that are not part of build config you can copy them using

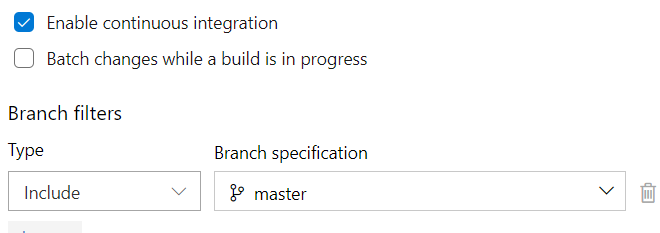
copy files task. you can specify target folder.

7. Finally publish the build artifacts using Publish build artifacts tasks you can set the path by setting the path under variables or you can use the default location from template as well.

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

- you can set the trigger on master branch in Triggers section where you can enable

continuous integration and select master under branch filters.



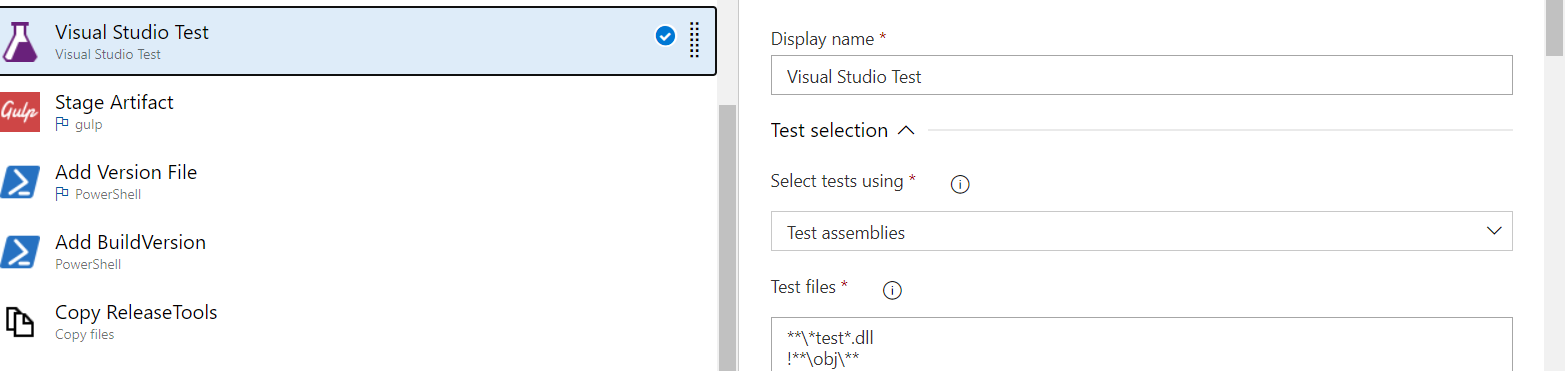
2) There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.

The build should not be successful if any test fails.

-**Add a task to run visual studio test with test assembly test file with below config**

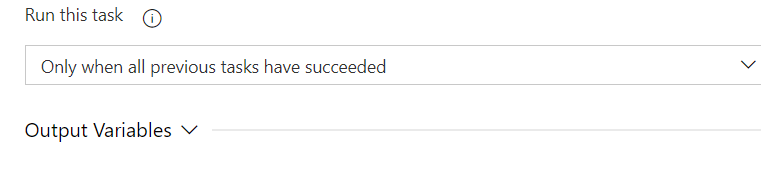
**\*\*\\*test\*.dll**

**!\*\*\obj\\*\***



and in next task select control action to only when all previous task have succeeded should work.

Also you can set custom condition.



Set UP Release pipeline for deployment.

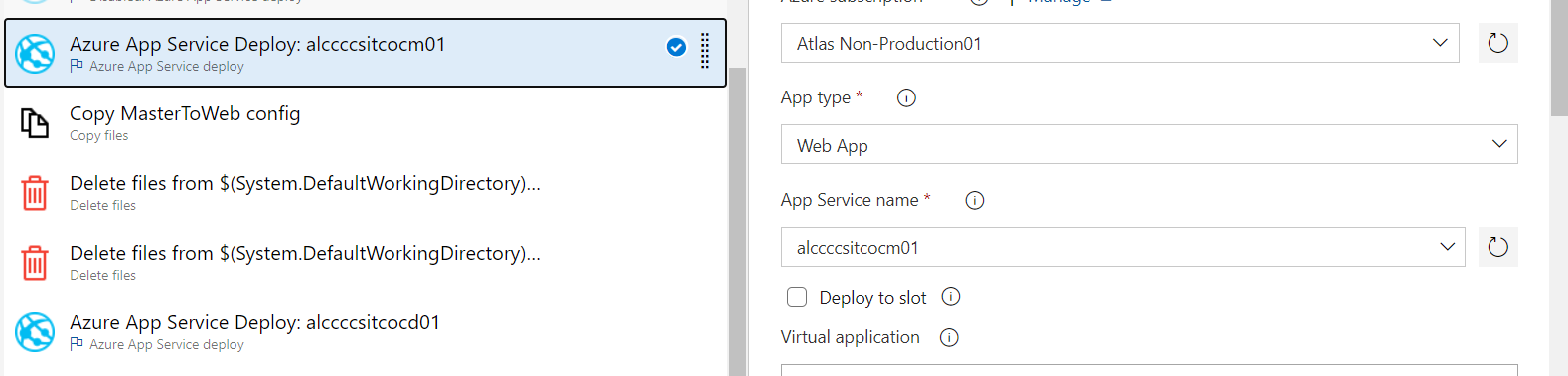
1. Go to Release section under pipeline, Create new pipeline.

2. In Artifact section click empty job and select source type as build select project and the build pipeline created, select version to latest

3.Add Stage DEV under pre-condition select after release.

4. Start adding task in side the DEV stage created you can select Azure pipeline agent or if you have any Microsoft agent you can choose.

5. From the task list select **Azure App Service deploy Azure subscription, App type , App service name which is created. also specify packaging path where the artifact is present**



i.e $(System.DefaultWorkingDirectory)/source\_alias/folder.

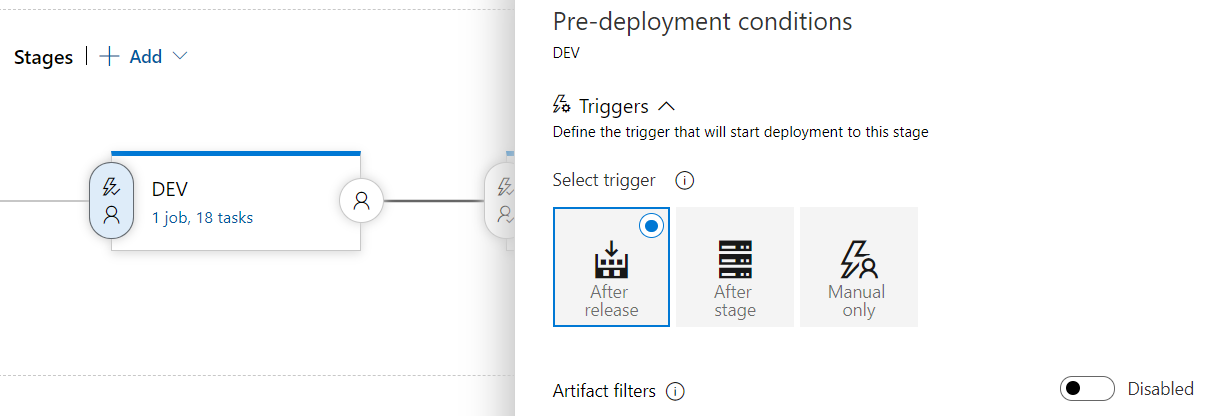
6. Add additional task any thing required.

7. Add two more stages from dev QA and PROD in the pre-condition step select after stage for QA after DEV and for PROD after QA. Enable pre-deployment approval and the users or group.

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

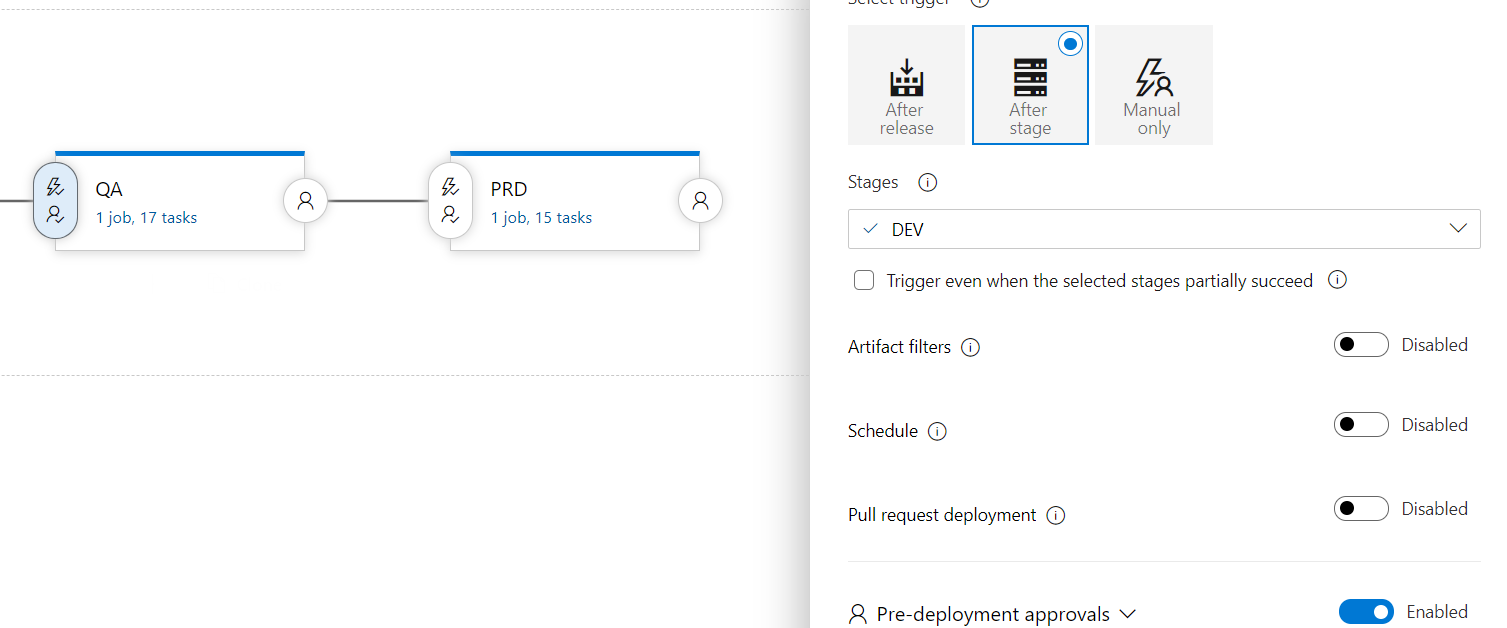
- . In Artifact section click empty job and select source type as build select project and the build pipeline created, select version to latest

Add Stage DEV under pre-condition select after release.



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

- Add two more stages from dev QA and PROD in the pre-condition step select after stage for QA after DEV and for PROD after QA



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

- Enable pre-deployment approval and the users or group.