**Response to VeriPark Azure DevOps(TFS) Test:**

**Tools used:**Azure DevOps (Aka VSTS) Pipelines for Build & Deploy (yaml script)

GitHub as source control

Powershell (optional for publishing to artifactory)

**Procedure:**

**Pre-requisites:**

* Created an Azure DevOps free organisation
* Created a “Test” Project
* Cloned the sample correct source code to github.
* Updated SampleSurvery.csproj file for “MSBuildExtensionsPath” to reflect visual studio version as 16.0 to make it compatible with Hosted Agents while executing pipelines from Azure DevOps.

**Test VP Pipeline set up:**

* Defining azure-pipelines.yml with the following :
  + Automated pipeline trigger for master branch whenever there is a check-in
  + Pool, variables for build platform, build configuration and solution
  + Tasks include
    - Nuget Installer,
    - Nuget Restore,
    - Building the solution,
    - Running test and
    - Publishing Artifacts(or we can add copy files task)
* Notifications can be subscribed on Azure DevOps as shown in the screenshots document.
* A Service connection to github account was configured.

**Attachments:**

* Yaml Script



* Pipeline Output logs



* Compiled output



* Powershell script to publish to jfrog artifactory



* Email Notifications

 

**URL’s used for this Test:**

<https://dev.azure.com/ChaitanyaKadiri/Test/>

<https://github.com/Chaits358/TestVP/>

I can give access to above URL’s if email address is provided.

**Improvements:**

* Enhanced security by using Azure DevOps Git
* Use powershell script to publish the release drop to Jfrog artifactory to store binaries.
* Use powershell scripts to deploy on to different environments by setting up release pipelines specific to environment.
* Scheduling the pipeline instead of triggering for every check-in can be helpful in some cases.
* Approvals for pull requests.
* Usage of gated check-in builds.