How to Setup Azure DevOps CICD Repository

# Create CICD for an organization

* Go to <https://dev.azure.com/>, create a new account
* Login to <https://dev.azure.com/> using that account, in our case it’s [lijunleejohn@gmail.com](mailto:lijunleejohn@gmail.com)
* Create a new Organization. E.g. lijunleejohndevops, then create a new project called CICDSandbox:

Graphical user interface, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

* Select Repos and use the information to clone or push a local repo to remote

Graphical user interface, text, application, email

Description automatically generated

* Install [Git for Windows](https://gitforwindows.org/)
* Assume local repo location is: C:\Users\john\source\lijunleejohn\lijunleejohndevops\repo
* Setup the Git Repository

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

* Commit and Push project “WebAppDockerTest”

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

* Check Azure DevOps

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Working!!

# Create New Repostory

Graphical user interface, text, application, email

Description automatically generated

# Create Web Deployment Package

* Right click Web project and select Publish 🡪 pick Web Deploy Package

Graphical user interface, application

Description automatically generated

* Make sure that the .gitignore file won’t ignore the Property folder

Graphical user interface, text, application

Description automatically generated

* Make sure that the .gitignore file should ignore the publish folder like p2p\_task\_web\_deploy

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

# CI (Build) pipeline template, should include

* NuGet
* VSBuild with generating Publish Profile Output
* Publish Artifacts from drop folder
* The Artifacts published will be the source of CD (Delivery) pipeline

|  |
| --- |
| # ASP.NET  # Build and test ASP.NET projects.  # Add steps that publish symbols, save build artifacts, deploy, and more:  # https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4  trigger:  - dev  pool:  name: 'P2P'  variables:  solution: '\*\*/OSOP.sln'  buildPlatform: 'Any CPU'  buildConfiguration: 'Release'  DefaultPackageFileName: 'OSOP.Presentation.zip'  PackageFilesOutputDir: '$(Build.ArtifactStagingDirectory)\output'  PublishProfileName: 'OSOPPackage'  steps:  - task: NuGetToolInstaller@1  - task: NuGetCommand@2  inputs:  restoreSolution: '$(solution)'  - task: VSBuild@1  inputs:  solution: '$(solution)'  msbuildArgs: '/p:DeployOnBuild=true;PublishProfile=$(PublishProfileName);DefaultPackageFileName=$(DefaultPackageFileName)'  platform: '$(buildPlatform)'  configuration: '$(buildConfiguration)'  - task: CopyFiles@2  displayName: 'Copy Installation Scripts to: $(PackageFilesOutputDir)'  inputs:  SourceFolder: $(build.sourcesDirectory)\  Contents: '\*.ps1'  TargetFolder: $(PackageFilesOutputDir)  OverWrite: true  flattenFolders: true  - task: CopyFiles@2  displayName: Copy webdeploy files to $(PackageFilesOutputDir)  inputs:  SourceFolder: $(build.sourcesDirectory)\osop\_web\_deploy\  TargetFolder: $(PackageFilesOutputDir)\osop\_web\_deploy\  flattenFolders: true  - task: PublishBuildArtifacts@1  displayName: 'Publish Artifact: drop'  inputs:  PathtoPublish: '$(PackageFilesOutputDir)'  ArtifactName: 'drop'  publishLocation: 'Container' |