Documentation on Therabill-Feeder

Version Number:1.0

Table of Contents

[Introduction: 2](#_Toc146532906)

[Scope: 2](#_Toc146532907)

[CI/CD Architecture: 3](#_Toc146532908)

[Deployment Group 3](#_Toc146532909)

[Build Pipeline for Therabill-Feeder 5](#_Toc146532910)

[Pipelines 7](#_Toc146532911)

[Trigger: 8](#_Toc146532912)

[GitTag: 8](#_Toc146532913)

[Publish Artifact 9](#_Toc146532914)

[Validation 9](#_Toc146532915)

[Release pipeline for Therabill-Feeder 9](#_Toc146532916)

[Steps 9](#_Toc146532917)

[Continuous deployment trigger: 11](#_Toc146532918)

[Pre-deployment conditions : 13](#_Toc146532919)

[Tasks: 14](#_Toc146532920)

[Agent Job: 15](#_Toc146532921)

[Azure powershell script:Inline Script 15](#_Toc146532922)

[Delete files 22](#_Toc146532923)

[Variables 23](#_Toc146532924)

[Validation 23](#_Toc146532925)

[Revision History  23](#_Toc146532926)

# Introduction:

This is the documentation for the Therabill-Feeder. It includes the build and release pipeline processes. Build pipelines include artifact generation, etc. Once the build is successful, the release pipeline starts triggering it.

# Scope:

* CI/CD is used because code can be easily deployed into a specific environment in a short period of time.
* Through the build pipeline, we can generate the artifacts, and through the release pipeline, we are able to deploy them in a specific location with the required changes.

# CI/CD Architecture:

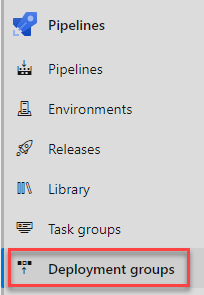
A diagram of a software company

Description automatically generated   
 

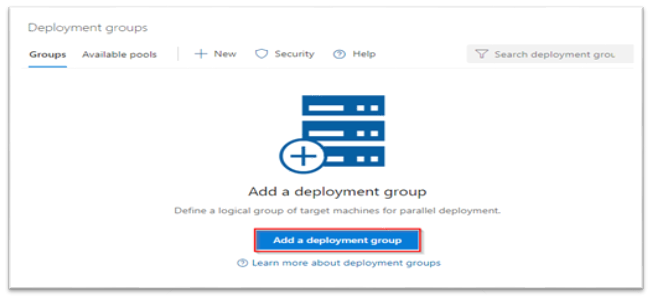
Deployment Group:

A deployment group is a logical set of deployment target machines that have agents installed on each one.

1. Create a deployment group.



1. Select Add a deployment group



1. Enter a Deployment group name and then select Create. A registration script will be generated. Select the Type of target to register and then select Use a personal access token in the script for authentication. Finally, select Copy script to the clipboard.

A screenshot of a computer

Description automatically generated

At this stage our deployment group is created but not online yet.

1. Log onto each of your target machines and run the script from an elevated PowerShell command prompt to register it as a target server. When prompted to enter tags for your agent, press *Y* and enter the tag(s) you will use to filter subsets of the servers.

A blue screen with white text

Description automatically generated

At this stage, our deployment group creation gets completed and now if you go and check the status of the deployment group in the Azure DevOps, it should show as Online.

A screenshot of a computer

Description automatically generated

# Build Pipeline for Therabill-Feeder

Steps:

1) Navigate the page to Pipelines and click on the “New Pipeline”.

A screenshot of a computer

Description automatically generated

2) Click on the “YAML”.

A screenshot of a computer

Description automatically generated

3) Select the Source as “Azure Repos Git” and followed by Project, Repository and branch.

A screenshot of a computer

Description automatically generated

4)Select the template as “YAML”

Graphical user interface, text, application, email, Teams

Description automatically generated

Pipelines:

We are using Azure pipelines (Microsoft hosted agent) as the agent pool for this pipeline. Microsoft-latest OS version will be used in the agent.

A white background with blue text

Description automatically generated

## Trigger:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## GitTag:

A screenshot of a computer code

Description automatically generated

# Publish Artifact

Publish Artifact is necessary for an automated deployment. This step publishes all the files which you want to deploy later in the release pipeline.

A screenshot of a computer code

Description automatically generated

# Validation

So, here “drop’ folder should include the following files:

A white rectangular object with a black border

Description automatically generated

# Release pipeline for Therabill-Feeder

Steps:

1. Navigate to release pipelines and click on “New“so as to create a new release pipeline which will be Therabill-Feeder in this case.

A screenshot of a computer

Description automatically generated

2) Select an empty job because we need to add customized jobs later which will support our project’s needs and functionality.

A screenshot of a computer

Description automatically generated

3)Before customizing the jobs, we will need to add the artifact that was built through our build pipeline. So, click on “Add an artifact” and then select the build pipeline which produced the artifact that is going to get used in this release pipeline.

A screenshot of a computer

Description automatically generated

4) As soon as the artifact is added, we must customize jobs according to the functionality of the project.

Graphical user interface, application, PowerPoint

Description automatically generated

# Continuous deployment trigger:

Continuous deployment triggers allow you to create a release every time a new build artifact is available.

A screenshot of a computer

Description automatically generated

Note -A release will be triggered only for a build that is from one of the branches selected here.  For example, selecting "DEV" will trigger a release for every build from the DEV branch.

Pull request trigger

Enabling this will create a release every time a selected artifact is available as part of a pull request workflow.

A screenshot of a computer

Description automatically generated

# Pre-deployment conditions :

Pre-deployment gates ensures there are no active issues in the work item or problem management system before deploying a build to an environment.

A screenshot of a computer

Description automatically generated

Note –

* Triggers - Set the trigger that will start the deployment to this stage automatically. Select "Release" to deploy to the stage every time a new release is created. Use the "Stage" option to deploy after deployments to selected stages are successful. To allow only manual deployments, select "Manual".
* Artifact filters - Select artifact condition(s) to trigger a new deployment. A release will be deployed to this stage only if all artifact conditions match.

Pre-deployment approvals –

Select the users who can approve or reject deployments to this stage.

A screenshot of a computer screen

Description automatically generated

# Tasks:

1. We must provide Stage Name, Azure subscription and App type and Name.

A screenshot of a computer

Description automatically generated

# Agent Job:

A job is a logical grouping of tasks that defines the runtime target on which the tasks will execute. An agent job executes tasks on an agent in an agent pool.

A screenshot of a computer

Description automatically generated

# Azure powershell script:Inline Script

The PowerShell task allows you to add PowerShell code directly within the YAML pipeline or execute an existing script in the source repo .

1. Making ProgramData Accessible to Everyone.

A screenshot of a computer

Description automatically generated

* 1. Replace tokens in FeederService.py - This task for Azure Pipelines extension that replace tokens in **text** files with variable values.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Replace tokens in SystemConfig.json - This task for Azure Pipelines extension that replace tokens in **text** files with variable values.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Copy files –

Copying of files from source folder to target folder. $(system.defaultworkingdirectory) to

  $( $(script)).

A screenshot of a computer

Description automatically generated

2) Copying of files from source folder to target folder. $(system.defaultworkingdirectory) to

  $( $( Targetpathconfig)).

A screenshot of a computer

Description automatically generated

3) Copying of files from source folder to target folder. $(system.defaultworkingdirectory) to

  $( $( TargetUtility)).

A screenshot of a computer

Description automatically generated

4) Copying of files from source folder to target folder. $(system.defaultworkingdirectory) to

  $( $( Feeder)).

A screenshot of a computer

Description automatically generated

Delete files: Delete folders, or files matching a pattern.

A screenshot of a music album

Description automatically generated

Variables:

A screenshot of a computer

Description automatically generated

# Validation

Once the release is successful, We check on the VM.

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **19/09/2023** | 1.0 | **Draft and Initial** | **Atul Sikarwal** |
| **19/09/2023** | 1.1 | **Review** |  |