**Documentation**

**of**

**IPERFORMFramework**

Version Number:1.0

**Table of Contents**

Introduction:………………………………………………………………………………………………………………………………2

[Scope: 2](#_Toc1766577935)

CI/CD Architecture:………………………………………………………………………………………………………………………………….2

Service Connection:……………………………………………………………………………………………………………………3

[Build Pipeline for IPERFORMFramework:…………………………………………………………………………………………………5](#_Toc984241579)

[Steps: 5](#_Toc96151906)

[Variables: 7](#_Toc1943940378)

[Agent: 7](#_Toc687689757)

[Triggers:- 7](#_Toc824237419) Tasks:……………………………………………………………………………………………………………………………………………….8[Validation: 10](#_Toc1639672742)

[Release pipeline for IPERFORMCollection 11](#_Toc442147090)

[Steps: 11](#_Toc954175103)

[Continuous deployment trigger: 13](#_Toc514064096)

[Pre-deployment conditions : 14](#_Toc1145827044)

[Tasks: 15](#_Toc1701658164)

[Agent Job: 15](#_Toc455866753)

[Azure powershell script:Inline Script 16](#_Toc783666366)

[ARM template deployment 17](#_Toc687119576)

[Azure App Service deploy 18](#_Toc1938074119)

[Variables Used: 19](#_Toc1620269157)

Validation:…………………………………………………………………………………………………………………………………………19

[Revision History 20](#_Toc1633143404)

**Introduction: -**

This is the documentation for the WebApp process. It includes the IASSIST solution file and release pipeline processes. Build pipelines include tasks, artifact generation, etc. Once the Build Pipeline 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: -**

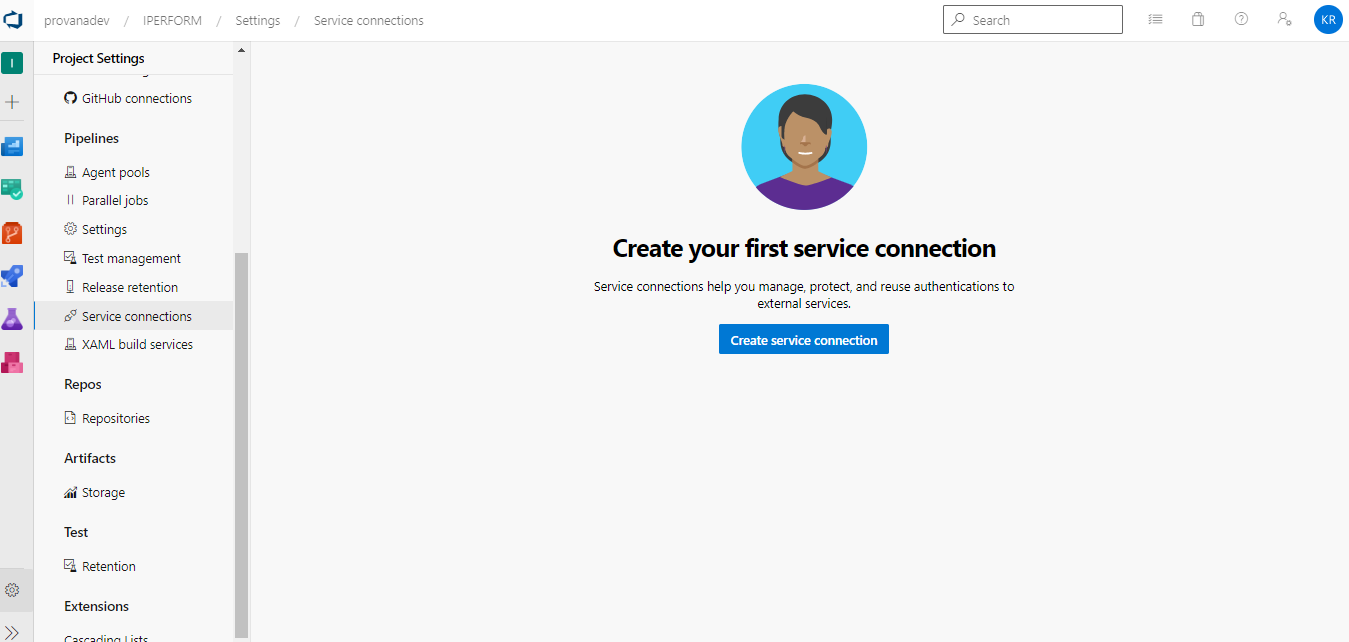
**Diagram

Description automatically generated**

**Service Connection: -**

Service Connections are used in Azure DevOps Pipelines to connect to external services, like Azure, GitHub, Docker, Kubernetes, and many other services.

1. In Azure DevOps, open the Service connections page from the project settings page.

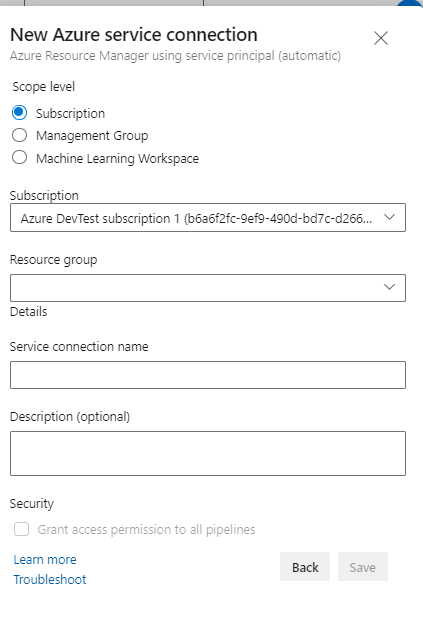


1. Click on create service connection and select Azure Resource Manager.

Graphical user interface, text, application, email

Description automatically generated

1. Choose Service Principal (Automatic) option and enter the Service Principal details..

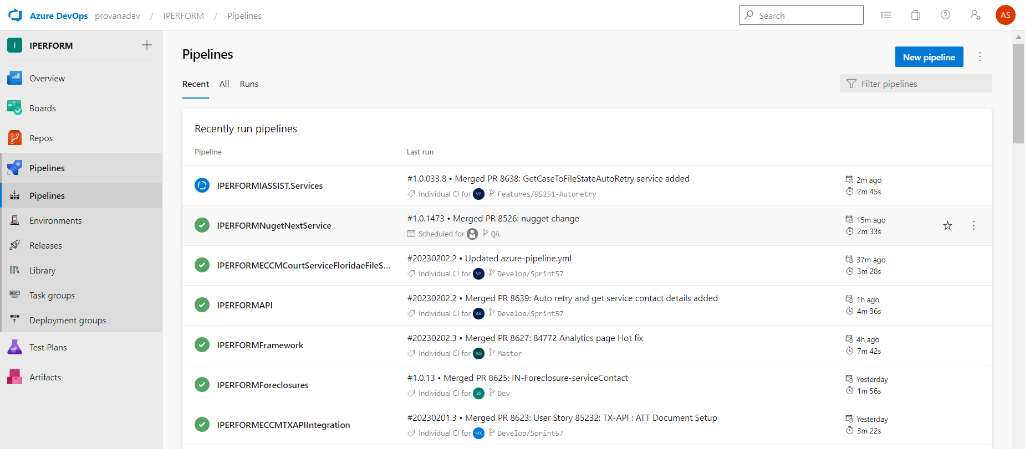


Build Pipeline for IPERFORMFramework: -

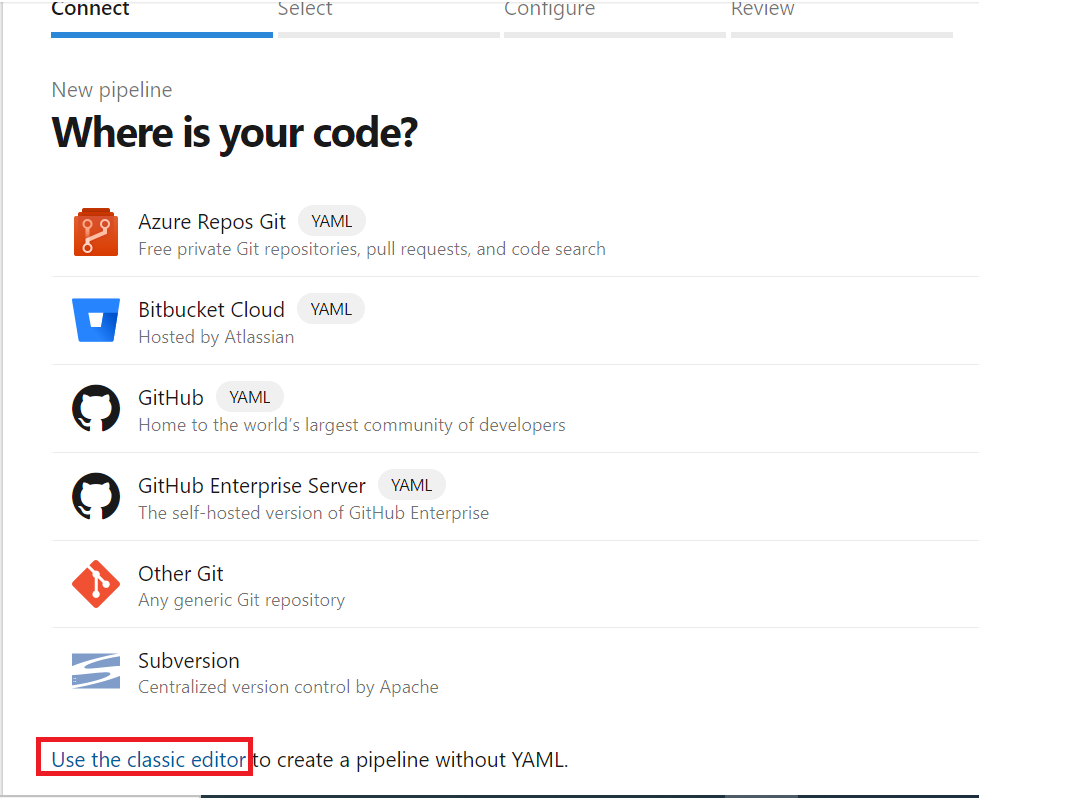
This pipeline – IPERFORMFramework will demonstrate the build process for IPERFORMFramework repository.

**Steps: -**

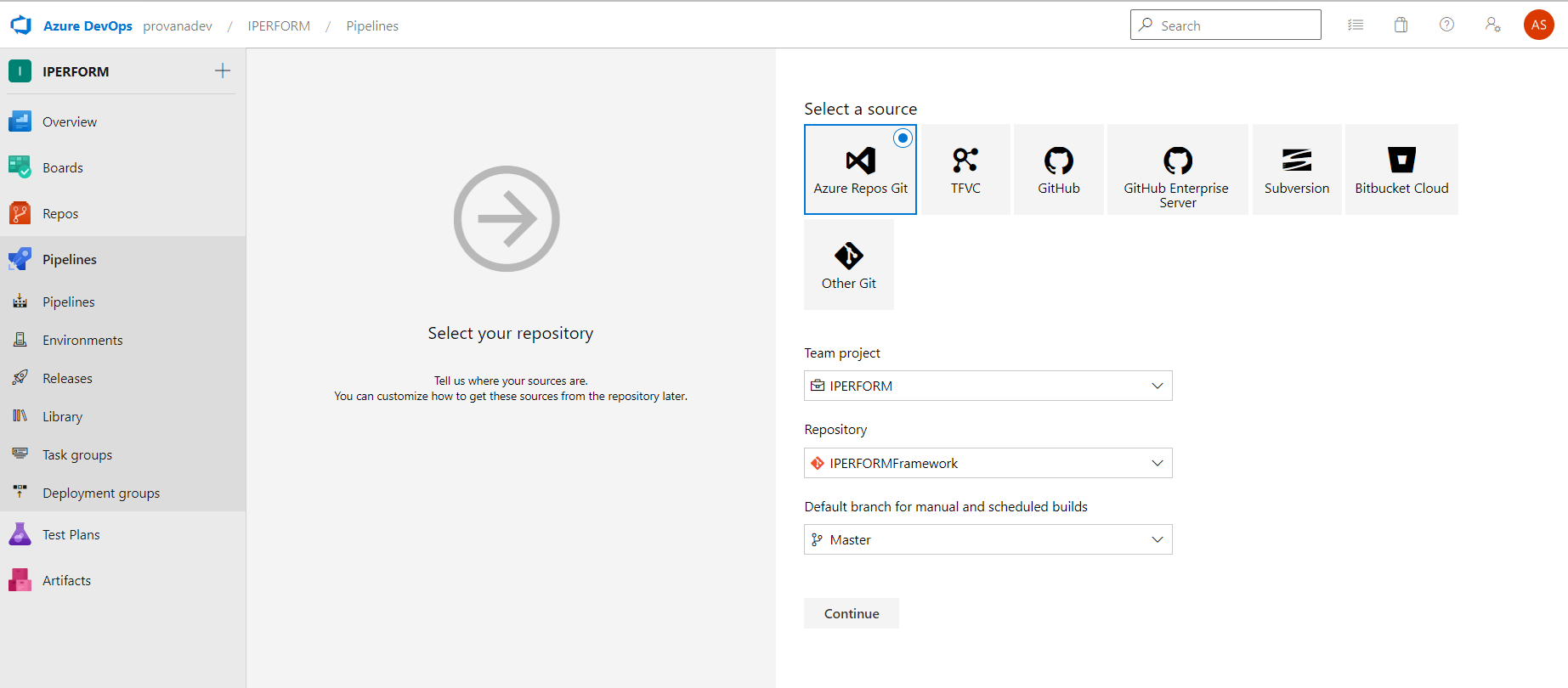
1) Navigate the page to Pipelines and click on the “New Pipeline”.



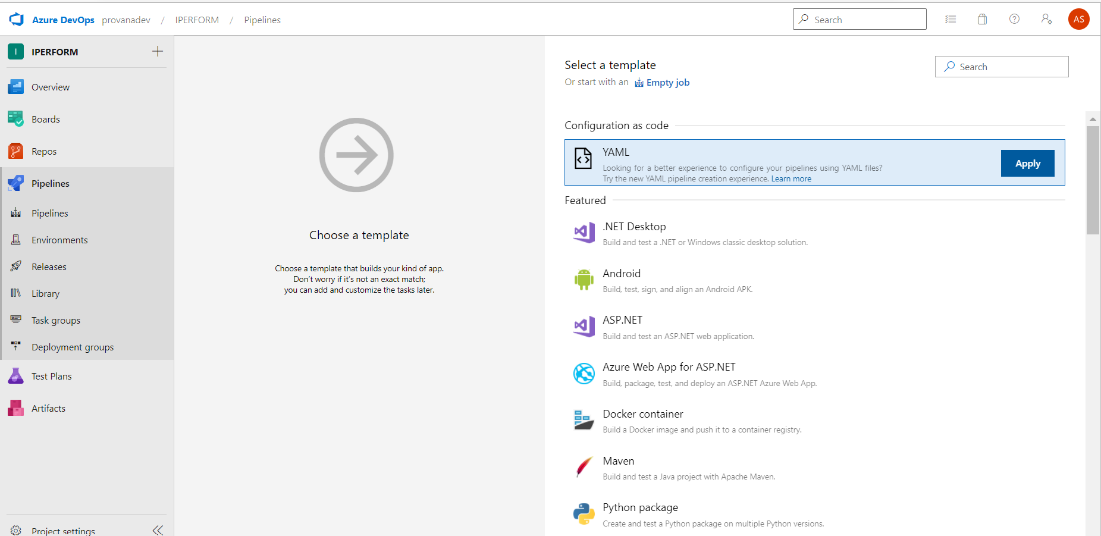
2) Click on the “Use the classic editor”.



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



4)Select the template as “YAML”



Variables:

* In the Variables tab, you can create variables that can be used in every step of the pipeline. For builds, you could set the username and password of an external service that you want to call.

Graphical user interface, text, application

Description automatically generated

Agent:

We are using Azure pipelines (Microsoft hosted agent) as the agent pool for this pipeline. Microsoft-2019 OS version will be used in the agent.



Triggers:

Shape

Description automatically generated with medium confidence

Tasks:

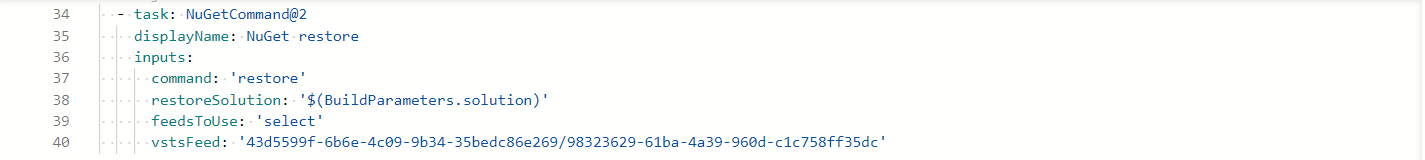
* **NuGet tool installer**

This acquires a specific version of NuGet from the internet or the tools cache and adds it to the PATH. Use this task to change the version of NuGet used in the NuGet tasks.

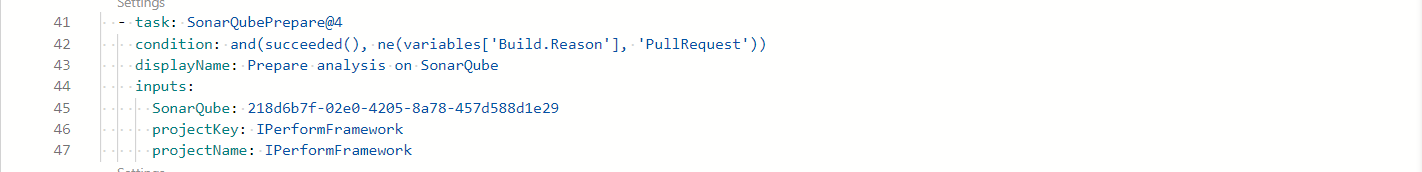


* **NuGet restore**

This step restores all the NuGet packages of our solution. Here, we use a variable $(solution) which is specified in the variables section.

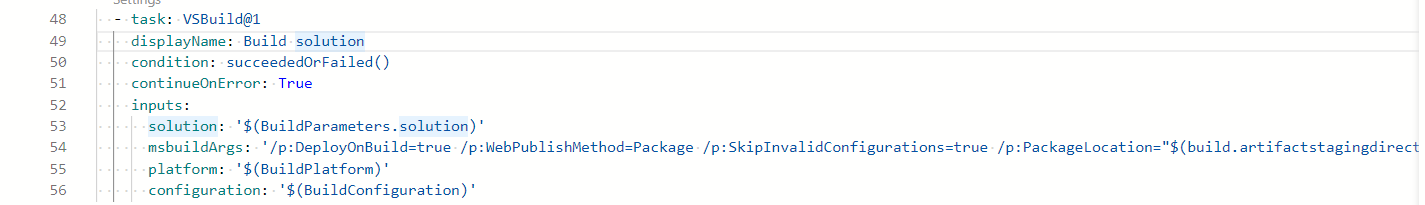


* **SonarQubePrepare@4**

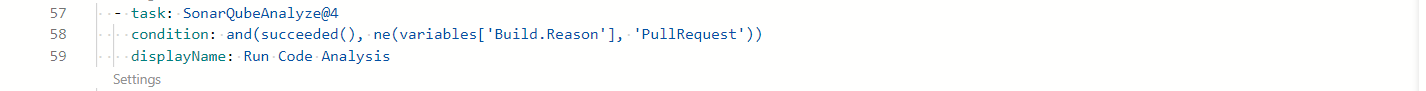


* **Visual Studio Build**

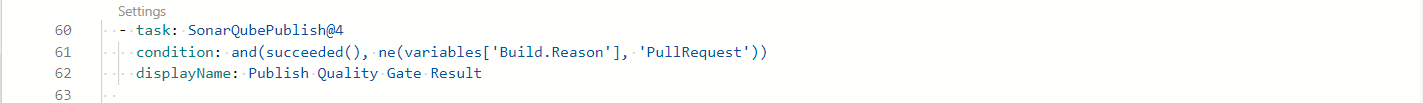
The Visual Studio Build step builds your application using the solution file. Here, we are using variable $(solution) which is specified in the variables section as the solution file and selecting platform configuration as $(buildPlatform), $(buildConfiguration) as the configuration file which is specified in the variables section as well.



* **SonarQubeAnalyze@4**

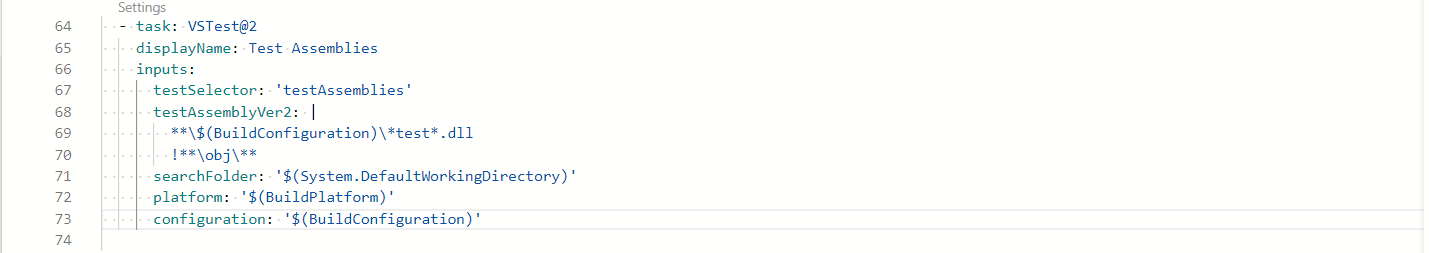


* **SonarQubePublish@4**



* **VSTEST@2**

Run unit and functional tests using the Visual Studio Test (VsTest) runner.Tests can be distributed on multiple agents using this task (version 2).

****

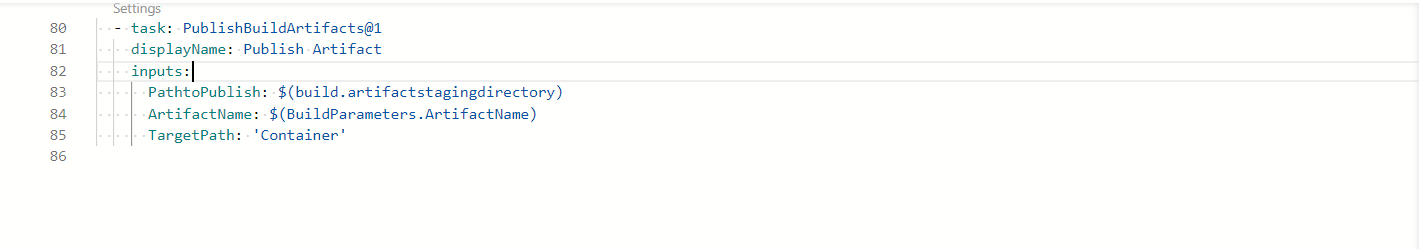
* **CopyFiles@2**

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

****

* **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.



Validation: -

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

Graphical user interface, application

Description automatically generated

Release Pipeline for IPERFORMFramework: -

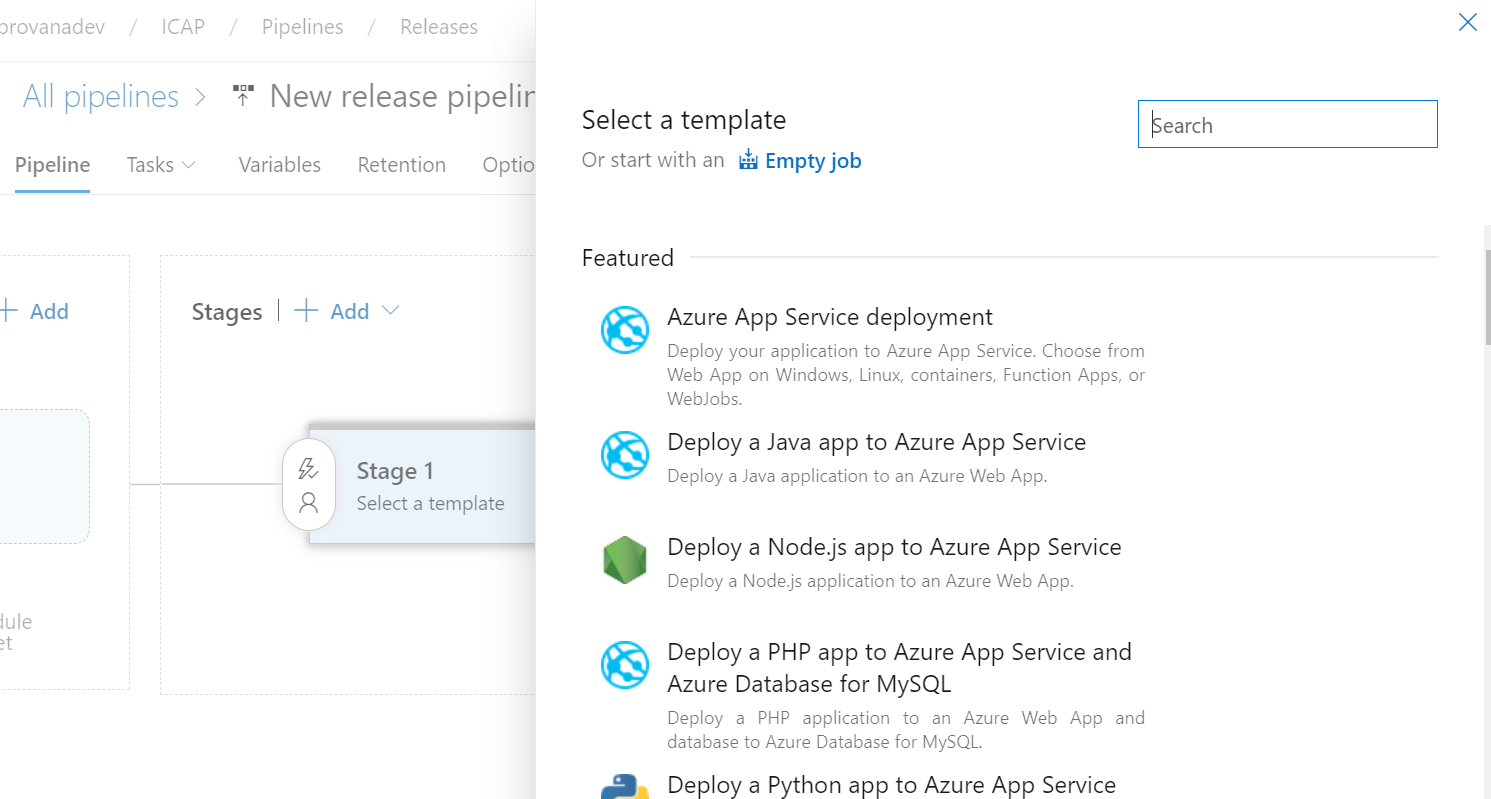
**Steps: -**

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

Graphical user interface, text, application, email

Description automatically generated

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



1. As soon as the artifact is added, we have to customize jobs according to the functionality of the project.

Graphical user interface, application

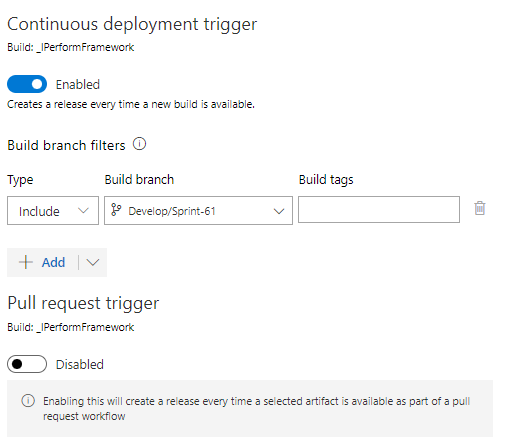
Description automatically generated

Graphical user interface, application

Description automatically generated

**Continuous deployment trigger: -**

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

****

**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.

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

Description automatically generated**

**Tasks: -**

The jobs needed in the IPERFROMFramework are shown below:

Graphical user interface, text, application, email

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.

Graphical user interface, text, application, email

Description automatically generated

* **Azure PowerShell**

Run a PowerShell script within an Azure environment. This task is used to check App service

Exist or not and creating the new App service.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* **ARM template deployment**

This job is used to deploy an Azure Resource Manager (ARM) template to all the

deployment scopes.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

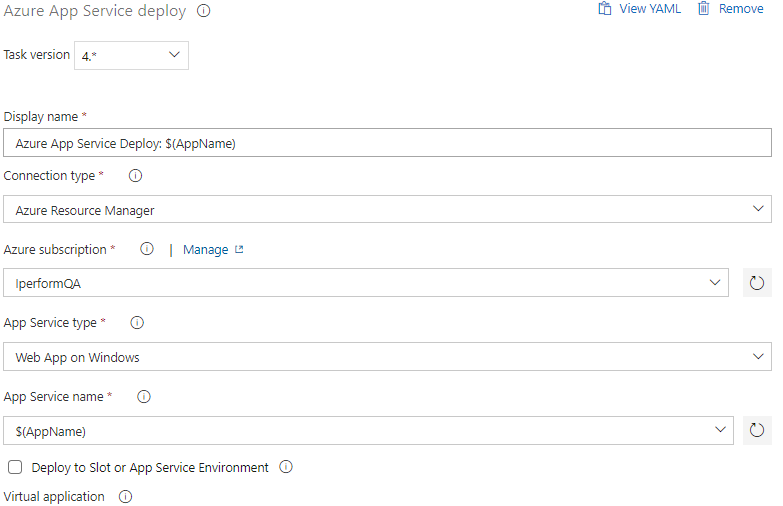
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* **Azure App Service deploy**

Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET, .NET Core, Node.js, PHP, Python, or Ruby.



Graphical user interface, text, application, email

Description automatically generated

**Variables used: -**

Graphical user interface, text, application, email

Description automatically generated

**Validation: -**

* Once the release pipeline (App Services) is successful, it checks if the app service plan exists or not; if it doesn't, it creates the deployment parameters.
* It will then begin deploying the web app via zip format.
* Once the deployment is complete, we can go to the web app link and check whether the site is up or not.

**Revision History: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **05/02/2023** | 1.0 | **Draft and Initial** | **Kamal Rajput** |