# **Tech Challenge - Assessment**

## **Requirement:**

Deploy the Golang application into Azure Cloud environment via automated process using Azure DevOps pipeline and provisioning azure resources using Terraform.

## **Tools Used:**

1. GitHub
2. Azure DevOps
3. Azure Cloud
4. Terraform
5. Postgres Database

## **Pre-Requisites for deployment solution:**

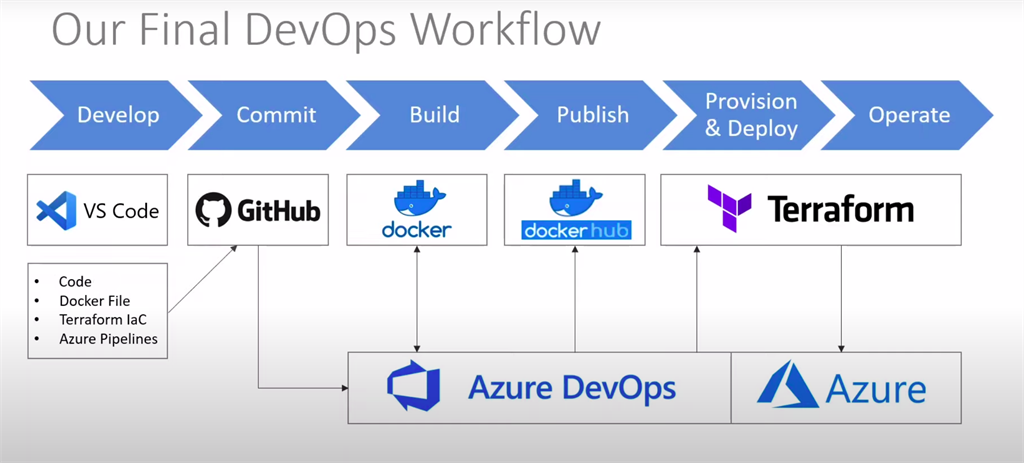
1. Fork the provided repository in GitHub to below location from s[ervian/TechChallengeApp](https://github.com/servian/TechChallengeApp)

[callsreenu/TechChallengeApp: Application for the Servian Tech Test (github.com)](https://github.com/callsreenu/TechChallengeApp)

1. Create a new project in Azure DevOps
2. Import code files from GitHub to Azure Repo
3. Create Subscription and Resource group in Azure Portal.
4. Create New Service connection in Azure DevOps Projects Settings.
5. Terraform scripts for provisioning the Azure resources.

## **High level architectural overview of deployment:**

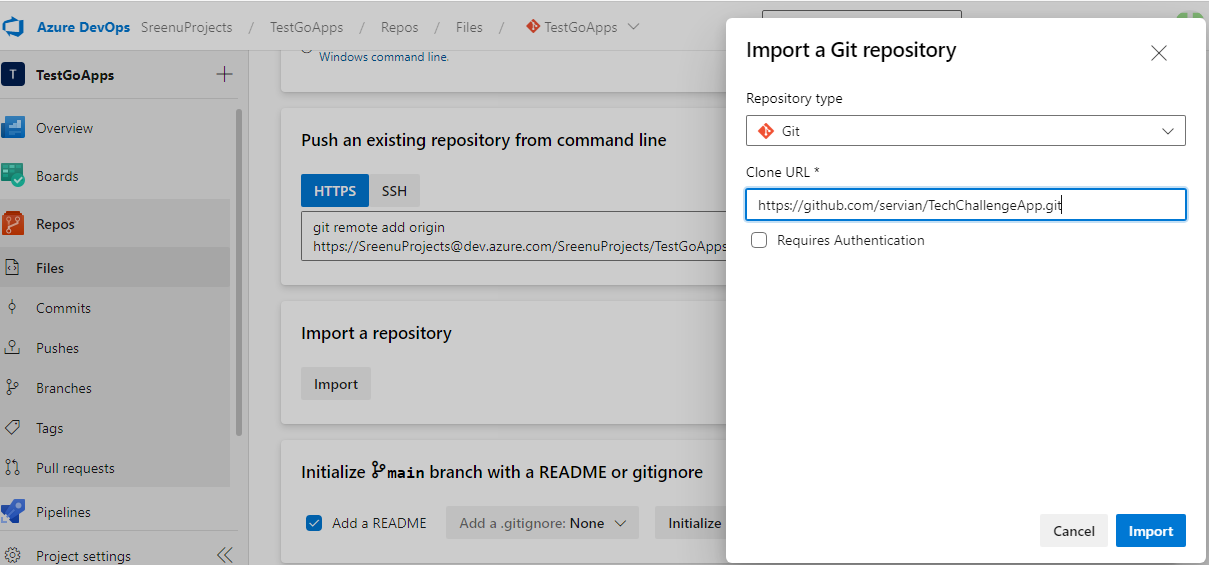
The below diagram depicts the deployment process for the Golang application deploying into Azure Cloud environment.



## **Process instructions for provisioning solution:**

Followed below instructions to implement the solution to the provided assessment.

Step1: Import the code from provided GitHub repository to Azure Repo.



Step 2: Created Azure YAML Pipeline to automate the entire process of deploying Golang application into Azure Cloud Environment. Following are the detailed steps as part of pipeline implementation.

1. Build Azure resources using Terraform modules.

Developed Terraform modules for creating below Azure resources and integrated in the Azure Pipeline.

* App service plan
* App service with a slot running Linux Container (Slots are used for deployment without Outage)
* App Insight
* Container registry
* Key vault
* Postgres SQL Server

1. Build Docker

* Install the docker and run the Dockerfile and then push the docker image to the Container Registry.

1. Build Go Application

* Compilation of the application using Go tool and package the application code and push to Artifactory.

1. Application Deployment

* Download artifact from the azure pipeline feed.
* Deploy the application package into azure web app container.
* Run the application to perform following activities
  + To create Postgres database and start it with test data.
  + To start serving requests.

Step 3: Download the code from Azure Repo and upload into GitHub repository.