**Q1 - SCENARIO**

A car rental company called FastCarz has a .net Web Application and Web API which are recently migrated from on-premise system to Azure cloud using Azure Web App Service

and Web API Service.

The on-premises system had 3 environments Dev, QA and Prod.

The code repository was maintained in TFS and moved to Azure GIT now. The TFS has daily builds which triggers every night which build the solution and copy the build package to drop folder.

deployments were done to the respective environment manually. The customer is planning to setup Azure DevOps Pipeline service for below requirements:

*1) The build should trigger as soon as anyone in the dev team checks in code to master branch.*

*2) There will be test projects which will create and maintained in the solution along the Web and API. The trigger should build all the 3 projects - Web, API and test.*

*The build should not be successful if any test fails.*

*3) The deployment of code and artifacts should be automated to Dev environment.*

*4) Upon successful deployment to the Dev environment, deployment should be easily promoted to QA and Prod through automated process.*

*5) The deployments to QA and Prod should be enabled with Approvals from approvers only.*

Explain how each of the above the requirements will be met using Azure DevOps configuration.

Explain the steps with configuration details.

1. ***Requirement 1***
   1. ***We will get option to select YAML pipelines.***
   2. ***Create build pipeline and select corresponding Azure repo in which all three projects (i.e., solution )are checked in and select classic pipeline.***
   3. ***Select starter pipeline or we could even use recommended templates based on solution which is given by azure devops. In our case, I will select starter pipeline and we could delete content.***
   4. ***Make sure have trigger in place pointing to Master branch.***
2. ***Requirement 2***
   1. *Set pool image: Ubuntu/window*
   2. *We can have multi stage build pipeline.*
   3. *Under stages will have stage will contains Jobs under which will contain list of jobs.*
   4. *I would design pipeline into five stages*
      1. *Restore & Build .net core API*
         1. *Using Restore and Build task which would need to point to that specific project.*
      2. *Restore & Build .net core web app*
         1. *Using Restore and Build task which would need to point to that specific project.*
      3. *Test unit test cases*
         1. *We would use test task which would run .net core unit test cases in the project using test task of .net core*
         2. *Publish the result using Publish test result task.*
         3. *Fail task on failed test property should be set to true, to make sure task is failed.*
         4. *Continueonerror property at job level needs be set to false to make sure we don’t run the next job.*
      4. *Publish API project*
      5. *Publish web project*
3. ***Requirement 3***
   1. *We can create release pipeline and select corresponding artifacts from published artifactory.*
   2. *Enable Continuous deployment enable trigger.*
   3. *Create Dev stage and create two agent job under which each one would point to corresponding API and web to deployment azure web apps.*
   4. *We can create QA and prod stage which would use same deployables and by selecting after stage option pointing to Dev.*

*Note: we need to have service connection in place to deploy the same.*

1. ***Requirement 4***
   1. *We need to have a group created in Azure devops for approving build with user added to it.*
   2. *We can enable pre deployment approval and select the specific group whom can deploy to higher environments.*

**Q2 - SCENARIO**

Macro Life, a healthcare company has recently setup the entire Network and Infrastructure on Azure.

The infrastructure has different components such as Virtual N/W, Subnets, NIC, IPs, NSG etc.

The IT team currently has developed PowerShell scripts to deploy each component where all the properties of each resource is set using PowerShell commands.

The business has realized that the PowerShell scripts are growing over period of time and difficult to handover when new admin onboards in the IT.

The IT team has now decided to move to Terraform based deployment of all resources to Azure.

All the passwords are stored in a Azure Service known as key Vault. The deployments needs to be automated using Azure DevOps using IaC(Infrastructure as Code).

*1) What are different artifacts you need to create - name of the artifacts and its purpose*

*2) List the tools you will to create and store the Terraform templates.*

*3) Explain the process and steps to create automated deployment pipeline.*

*4) Create a sample Terraform template you will use to deploy Below services:*

*Vnet*

*2 Subnet*

*NSG to open port 80 and 443*

*1 Window VM in each subnet*

*1 Storage account*

*5) Explain how will you access the password stored in Key Vault and use it as Admin Password in the VM Terraform template.*

1. Requirement1
   1. Different artifacts required for this scenario
      1. Provider.tf file which would contains provider details i.e., azure
      2. Variables.tf file which would contain definition of variables
      3. Main.tf file which would contains all resources definition to provision above scenario.
      4. Auto.tfvars which is used for
      5. Backend.tf to maintain the state in remote blob storage
2. Requirement 2
   1. I would use VSCode for creating terraform and push it to the azure git repo for managing the version.
3. Requirement 3
   1. We have Azure git repo which contains our Infra terraform script in different folder and with correponsing TFvars file.
   2. We will create release pipeline from repo and setup different stage for different environment.
   3. We have created variable group and even associated with key vault to fetch password securely form key vault.
   4. We have associated corresponding variable group for each stage.
   5. We have tasks to init,plan and apply terraform scripts in Azure Devops.
4. Requirement 4
   1. PFA 
5. Requirement 5
   1. We can have azure devops to fetch password from keyvault.
   2. We could even use data source, which I feel would be other approach to fetch it.