**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. We need to enable auto trigger using trigger and specify the branch name in the pipeline.

trigger:

- master

1. *We need to use the two task one is used to run unit tests using VSTests job this will run the test and Use publish test results job to fail the pipeline on test failure.*

We can use test assemblies and specify the test files located in the project like below.  
  
\*\*\\*test\*.dll

!\*\*\\*TestAdapter.dll

!\*\*\obj\\*\*

We need to specify the type of test like VSTest and test results file like \*\*/TEST-\*.xml.

1. We can enable Continuous deployment trigger option in artifacts in the release pipeline to enable auto trigger if new artifact is available in azure artifacts Specify pre-deployment condition and enable pull request deployment option to enable auto deployment in DEV stage.
2. We can create multiple stage in deployment pipeline one is for DEV, QA and Prod, we can Specify pre-deployment condition and enable auto trigger and select trigger as After stage DEV of QA stage once the DEV deployment stage is completed and enable auto trigger in Prod stage as After stage QA to proceed once the QA deployment stage is completed.
3. We can enable use pre-deployment approval in QA and DEV Stage to procced with the deployment only it is approved by specified approvers / owners.  
     
   We can set following settings in this stage multiple approvers and time out in this stage

**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)

1. **Storage Account** to store the Terraform state file.
2. **Azure Service Principal** used as an identity to authenticate you within your Azure Subscription to allow you to deploy the relevant Terraform code.
3. **Terraform Extension** required for deploying terraform into azure need to install it if it is not already installed in the organization settings.
4. **Azure repo** for storing the terraform templates.
5. **Azure pipeline** to deploy the template to Azure.

2) I have used VS code to create the terraform template and used git extension in VS Code to push it to Azure repo.

3) Need to create a pipeline using Starter Pipeline and Now we need to add three tasks install, validate and apply

We need to add the service principal in project settings > service connections.

First task needs to install terraform (terraform tool install)

Second task is the terraform validate task

Third task is the terraform apply task

Validate and Apply task we need to provide below required details.

backendServiceArm: Service Principal Name

backendAzureRmResourceGroupName: Resource Group of Storage Account

backendAzureRmStorageAccountName: Storage Account Name

backendAzureRmContainerName: Storage Container Name

backendAzureRmKey: State File Name

4)



5) We need to create Variable Groups in pipelines > library by linking key vaults as variable.  
  
The same can be used in pipeline using

Variables:

- group: <Created group name>  
  
while running terraform command we can specify the same in commandOptions like below.  
  
commandOptions: 'Password=$(admin\_password)'