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.

Answer- In the Azure Pipeline, goto   
 - Triggers tab – In Continuous Integration Repository

- Check the box Enable Continuous Integration

- In Branch Filters, specify the branch as master

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.

Answer- In the Azure Pipeline, goto   
 - Tasks tab – Add a Task to Agent job – Select Visual Studio Build and specify the Web and API solutions .sln File

- Add another task to build Test project Solution

- Add the Task – VsTest – TestAssemblies and specify the Test selection –

Select Test Assemblies and specify the Test files as - \*\*\\*test\*.dll so that it will pick all the Test Dlls from Default working Directory and run the Tests

* Check the checkbox – Fail the task if a minimum number of Tests are not run, this will fail the build if any of the tests are not successful

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

Answer- In the Build pipeline, Add the task Publish Build artifacts and publish to File share accessible in the Dev Environment.   
 - Add a release pipeline, and select the Artifacts as Build and select the build from the drop down  
 - Add a stage as an Empty job and name it as Dev  
 - In the Dev stage, add the task – Create Environment Task. Specify the Dev Environment name as Dev and use the login credentials defined in the Environment as variables

4) Upon successful deployment to the Dev environment, deployment should be easily promoted to QA

and Prod through automated process.  
Answer – In the same release pipeline add 2 more stages and name them as QA and Prod select the predeployment condition as to run after Stage   
- select the stage - Dev from the drop down for both QA and Prod stages  
- In QA and Prod also, add the respective environments to the deployed to

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.

Answer - Goto, the organization’s Azure link and select Environments.

* For QA and Prod Environments, select Approvals and checks and add the users or group
* It can be added in Pre-deployment approvals of QA and Prod stages in release pipeline
* In the stage, goto Pre-deployment conditions and enable Pre-deployment approvals and select the users or user group. Check the box, users requesting a release or deployment should not approve it.

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  
Answer- Terraform configuration file - .tf file which contains the deployment of each component

2) List the tools you will to create and store the Terraform templates.  
Answer - Git, Azure pipeline

3) Explain the process and steps to create automated deployment pipeline.  
 i) Create a release 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.