**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 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.

**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 ARM 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*

There are two main artifacts one is Template file and second is template parameter file.

Template – Templates are JSON files, which contains Schema, Parameters, Variables, Resources and Outputs as data to deploy an infrastructure on Azure through Automation.

Template Parameters - When automating deployments it can be easier to pass a set of values for your environment. Parameter files make it easier to package parameter values for a specific environment in a separate Parameters file

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

Visual Studio Code – Used to create ARM templates

Github – Used to store ARM templates

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

a). Prepare ARM templates and commit them to Azure Repos / Github

b). Create a build pipeline with Azure Repo / Github as the source for templates (artifacts)

c). Use MS-Hosted agent or Self-Hosted agent for execution of Pipeline

d). Create deployment task and create Agent Job’s under the task

e). Create service connection to Azure Subscription to deploy resources from Azure Devops

f). Provide resource group information if already exists, or create a new one using Variables

f). In the Agent Job, choose ARM template deployment (predefined) job and provide artifact information such as which template to use and which parameter file to use for deployment.

g). After pipeline is configured with required inputs, save it and run to deploy resources automatically.

*4) Create a sample ARM template you will use to deploy a Windows VM of any size*

Sent you through email.

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

a). Create a keyvault and create a secret which will be used as Admin Password for VM

b). Click on Access Policies in the KeyVault, select “Azure Resource Manager for template deployment” option and save.

c). Create a template parameter file and add a parameter named “adminPassword” with below code to access the secret that we have created earlier.

"adminPassword": {

"reference": {

"keyVault": {

"id": "Key Vault Id Here"

},

"secretName": "Name of the secret in Azure Key Vault"

}

}

d). keyVault Id can be retrieved from the properties of KeyVault and secretName should be the name of the secret that we have create earlier.

**Q3 - SCENARIO**

A Toy Retail company ToyTrex has it retail application deployed as 3-tier application - Web App(UI), Web API(middle layer) and Database as Azure SQL.

The user load started increasing multiple fold every month and complex programs getting implemented, the application started performing poorly.

As a result, company decided to re-architect the middle layer as microservices using Azure Kubernetes Services.

The new architecture has below design decisions.

*1) The middle layer should be implemented as Microservices using Azure AKS*

*2) The middle layer API should be deployed as containerized application images*

*3) The container images will use Azure Container Repository (ACR) as the private image repository*

*4) The CI/CD pipelines for microservices should be implemented using Azure DevOps services.*

*5) The Azure DevOps should be able to access ACR and download the container images for microservices deployment*

*6) The image should be deployed as templates such as <image\_name>:<build\_id>*

*Explain the DevOps configuration and steps in detail for above requirements*