**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**

We would be creating multiple artifacts for terraform as well as for Azure pipeline as below.

1. We need an Azure Service Principal that will be used to authenticate and provide permission to Azure devops pipeline to Azure resource Manager in Azure cloud. Once this Azure Service principal is created it will be configured in the Azure Devops Service connections.
2. We must create a storage account in the Azure cloud to store the terraform backend states
3. Azure key vault must be created to store the passwords related to the admin user for the windows VM
4. Terraform template files mentioned below have to be created
   1. providers.tf – This will be used to authenticate and configure infrastructure in [Microsoft Azure](https://azure.microsoft.com/en-us/) using the Azure Resource Manager API
   2. locals.tf – This will be used to assign value that can be used repeatedly in the terraform templates
   3. data.tf – This will be used to get the information that is defined outside this terraform template
   4. variables.tf – This will be used to define the input variables hat will serve as parameters to the templates.
   5. output.tf – This will define the return values from this terraform template that can be used by other modules or templates.
   6. main.tf – This will contain the backend details of where the terraform state will be stored as well as the terraform resources that needs to be provisioned on Azure cloud.
5. Yaml based azure devops pipeline that will triggered to apply and destroy resources on Azure cloud

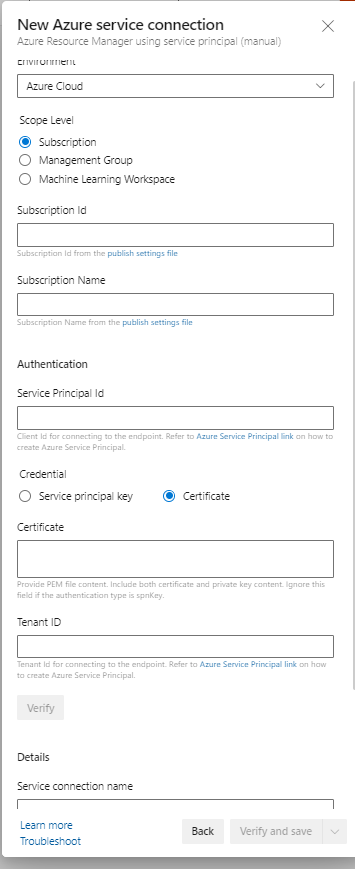
**2) List the tools you will to create and store the Terraform templates.**

Terraform templates will be stored in the Azure Devops Git repos from where pipeline will be triggered to provision resource on Azure cloud.

Azure storage account will be used as a *Terraform remote backend*. Storage account can be provisioned using the Azure CLI and service principal.

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

1. Create a Azure service connection using the service principal by filling all the details below.



1. Create a Azure storage account for Terraform remote state backend.

RESOURCE\_GROUP\_NAME=rg-terraform

STORAGE\_ACCOUNT\_NAME=tfstate$RANDOM

CONTAINER\_NAME=dev

# Create resource group

az group create --name $RESOURCE\_GROUP\_NAME--location westeurope

# Create storage account

az storage account create --resource-group $RESOURCE\_GROUP\_NAME--name $STORAGE\_ACCOUNT\_NAME--sku Standard\_LRS --encryption-services blob

# Get storage account key

ACCOUNT\_KEY=$(az storage account keys list --resource-group $RESOURCE\_GROUP\_NAME--account-name $STORAGE\_ACCOUNT\_NAME--query [0].value -o tsv)

# Create blob container

az storage container create --name $CONTAINER\_NAME--account-name $STORAGE\_ACCOUNT\_NAME--account-key $ACCOUNT\_KEY

1. Create a Azure key vault to store VM admin password and provide necessary access permissions to the service principal so that it can retrieve the password from key vault.
2. Create the azure pipeline to run the terraform init/validate/apply destroy commands. The Azure pipeline is part of the github repo.

**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**

All the terraform resources and templates are part of the github repo

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

Password can be accessed through terraform datasources mentioned in the data.tf file

data "azurerm\_key\_vault\_secret" "adminpassword" {

name = "adminpwd"

vault\_uri = "https://macrolife.vault.azure.net/"

}

It can be used in azurerm\_windows\_virtual\_machine resource by referencing the data.

admin\_username = "adminuser"

admin\_password = data.azurerm\_key\_vault\_secret.adminpassword