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

**Ans :-** *We need application artifact with arm template. In deployment, first resources will be get created and then application will deploy over the resource.*

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

**Ans:- T**he Main tool require to create an Arm template is an editor like VS code And to execute the Arm template we can utilize Azure PowerShell CLI.

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

**Ans:-**

In deployment pipeline, first we need to get the artifact from build or need to link with build. We can choose an ARM template deployment task. Initially we also need to connection with azure resource service.

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

**Ans:-**

{

    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

    "contentVersion": "1.0.0.0",

    "parameters": {

        "location": {

            "type": "String"

        },

        "networkInterfaceName": {

            "type": "String"

        },

        "networkSecurityGroupName": {

            "type": "String"

        },

        "networkSecurityGroupRules": {

            "type": "Array"

        },

        "subnetName": {

            "type": "String"

        },

        "virtualNetworkName": {

            "type": "String"

        },

        "addressPrefixes": {

            "type": "Array"

        },

        "subnets": {

            "type": "Array"

        },

        "publicIpAddressName": {

            "type": "String"

        },

        "publicIpAddressType": {

            "type": "String"

        },

        "publicIpAddressSku": {

            "type": "String"

        },

        "virtualMachineName": {

            "type": "String"

        },

        "virtualMachineComputerName": {

            "type": "String"

        },

        "virtualMachineRG": {

            "type": "String"

        },

        "osDiskType": {

            "type": "String"

        },

        "virtualMachineSize": {

            "type": "String"

        },

        "adminUsername": {

            "type": "String"

        },

        "adminPassword": {

            "type": "SecureString"

        },

        "patchMode": {

            "type": "String"

        },

        "autoShutdownStatus": {

            "type": "String"

        },

        "autoShutdownTime": {

            "type": "String"

        },

        "autoShutdownTimeZone": {

            "type": "String"

        },

        "autoShutdownNotificationStatus": {

            "type": "String"

        },

        "autoShutdownNotificationLocale": {

            "type": "String"

        }

    },

    "variables": {

        "nsgId": "[resourceId(resourceGroup().name, 'Microsoft.Network/networkSecurityGroups', parameters('networkSecurityGroupName'))]",

        "vnetId": "[resourceId(resourceGroup().name,'Microsoft.Network/virtualNetworks', parameters('virtualNetworkName'))]",

        "subnetRef": "[concat(variables('vnetId'), '/subnets/', parameters('subnetName'))]"

    },

    "resources": [

        {

            "type": "Microsoft.Network/networkInterfaces",

            "apiVersion": "2018-10-01",

            "name": "[parameters('networkInterfaceName')]",

            "location": "[parameters('location')]",

            "dependsOn": [

                "[concat('Microsoft.Network/networkSecurityGroups/', parameters('networkSecurityGroupName'))]",

                "[concat('Microsoft.Network/virtualNetworks/', parameters('virtualNetworkName'))]",

                "[concat('Microsoft.Network/publicIpAddresses/', parameters('publicIpAddressName'))]"

            ],

            "properties": {

                "ipConfigurations": [

                    {

                        "name": "ipconfig1",

                        "properties": {

                            "subnet": {

                                "id": "[variables('subnetRef')]"

                            },

                            "privateIPAllocationMethod": "Dynamic",

                            "publicIpAddress": {

                                "id": "[resourceId(resourceGroup().name, 'Microsoft.Network/publicIpAddresses', parameters('publicIpAddressName'))]"

                            }

                        }

                    }

                ],

                "networkSecurityGroup": {

                    "id": "[variables('nsgId')]"

                }

            }

        },

        {

            "type": "Microsoft.Network/networkSecurityGroups",

            "apiVersion": "2019-02-01",

            "name": "[parameters('networkSecurityGroupName')]",

            "location": "[parameters('location')]",

            "properties": {

                "securityRules": "[parameters('networkSecurityGroupRules')]"

            }

        },

        {

            "type": "Microsoft.Network/virtualNetworks",

            "apiVersion": "2019-09-01",

            "name": "[parameters('virtualNetworkName')]",

            "location": "[parameters('location')]",

            "properties": {

                "addressSpace": {

                    "addressPrefixes": "[parameters('addressPrefixes')]"

                },

                "subnets": "[parameters('subnets')]"

            }

        },

        {

            "type": "Microsoft.Network/publicIpAddresses",

            "apiVersion": "2019-02-01",

            "name": "[parameters('publicIpAddressName')]",

            "location": "[parameters('location')]",

            "sku": {

                "name": "[parameters('publicIpAddressSku')]"

            },

            "properties": {

                "publicIpAllocationMethod": "[parameters('publicIpAddressType')]"

            }

        },

        {

            "type": "Microsoft.Compute/virtualMachines",

            "apiVersion": "2020-06-01",

            "name": "[parameters('virtualMachineName')]",

            "location": "[parameters('location')]",

            "dependsOn": [

                "[concat('Microsoft.Network/networkInterfaces/', parameters('networkInterfaceName'))]"

            ],

            "properties": {

                "hardwareProfile": {

                    "vmSize": "[parameters('virtualMachineSize')]"

                },

                "storageProfile": {

                    "osDisk": {

                        "createOption": "fromImage",

                        "managedDisk": {

                            "storageAccountType": "[parameters('osDiskType')]"

                        }

                    },

                    "imageReference": {

                        "publisher": "MicrosoftWindowsServer",

                        "offer": "WindowsServer",

                        "sku": "2012-Datacenter-smalldisk",

                        "version": "latest"

                    }

                },

                "networkProfile": {

                    "networkInterfaces": [

                        {

                            "id": "[resourceId('Microsoft.Network/networkInterfaces', parameters('networkInterfaceName'))]"

                        }

                    ]

                },

                "osProfile": {

                    "computerName": "[parameters('virtualMachineComputerName')]",

                    "adminUsername": "[parameters('adminUsername')]",

                    "adminPassword": "[parameters('adminPassword')]",

                    "windowsConfiguration": {

                        "enableAutomaticUpdates": true,

                        "provisionVmAgent": true,

                        "patchSettings": {

                            "patchMode": "[parameters('patchMode')]"

                        }

                    }

                },

                "diagnosticsProfile": {

                    "bootDiagnostics": {

                        "enabled": true

                    }

                }

            }

        },

        {

            "type": "Microsoft.DevTestLab/schedules",

            "apiVersion": "2017-04-26-preview",

            "name": "[concat('shutdown-computevm-', parameters('virtualMachineName'))]",

            "location": "[parameters('location')]",

            "dependsOn": [

                "[concat('Microsoft.Compute/virtualMachines/', parameters('virtualMachineName'))]"

            ],

            "properties": {

                "status": "[parameters('autoShutdownStatus')]",

                "taskType": "ComputeVmShutdownTask",

                "dailyRecurrence": {

                    "time": "[parameters('autoShutdownTime')]"

                },

                "timeZoneId": "[parameters('autoShutdownTimeZone')]",

                "targetResourceId": "[resourceId('Microsoft.Compute/virtualMachines', parameters('virtualMachineName'))]",

                "notificationSettings": {

                    "status": "[parameters('autoShutdownNotificationStatus')]",

                    "notificationLocale": "[parameters('autoShutdownNotificationLocale')]",

                    "timeInMinutes": "30"

                }

            }

        }

    ],

    "outputs": {

        "adminUsername": {

            "type": "String",

            "value": "[parameters('adminUsername')]"

        }

    }

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

Ans:- Fist we have to create a azure key vault service on azure portal and give the get and list permission to user. Now connect from pipeline using the service connection. Once that’s done then those variable will be available in our pipeline.