**Azure Resource Manager (ARM) – Hands-on Lab Guide**

**Lab 1: Getting Started with ARM Templates**

**Objective:** Understand the structure of an ARM template and deploy your first resource (Storage Account).

**Steps:**

1. **Open Azure Portal**: Go to https://portal.azure.com
2. **Search** for "Template Deployment" in the search bar.
3. Select **"Deploy a custom template"**.
4. Click on **"Build your own template in the editor"**.
5. Paste the below template:

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"storageAccountName": {

"type": "string",

"defaultValue": "mystoragearmdemo"

}

},

"resources": [

{

"type": "Microsoft.Storage/storageAccounts",

"apiVersion": "2022-09-01",

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

"location": "[resourceGroup().location]",

"sku": {

"name": "Standard\_LRS"

},

"kind": "StorageV2",

"properties": {}

}

]

}

1. Click **Save**, provide a resource group and region, then **Deploy**.

**Expected Outcome:**

A new **Storage Account** is created using ARM deployment.

**Lab 2: Parameterizing the Template**

**Objective:** Learn how to create and use a parameter file.

**Steps:**

1. Use the same template from Lab 1.
2. Save this as azuredeploy.json.
3. Create a new file azuredeploy.parameters.json with the following content:

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"storageAccountName": {

"value": "mystorageparamdemo"

}

}

}

1. Open **Cloud Shell** or **local PowerShell/CLI** and run:

az deployment group create \

--resource-group <your-resource-group> \

--template-file azuredeploy.json \

--parameters @azuredeploy.parameters.json

**Expected Outcome:**

Storage account named mystorageparamdemo is created using external parameters.

**Lab 3: Deploy a Virtual Network with Subnets**

**Objective:** Deploy a VNet with two subnets using an ARM template.

**Template Snippet:**

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"vnetName": { "type": "string", "defaultValue": "demo-vnet" }

},

"resources": [

{

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

"apiVersion": "2022-07-01",

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

"location": "[resourceGroup().location]",

"properties": {

"addressSpace": { "addressPrefixes": [ "10.0.0.0/16" ] },

"subnets": [

{ "name": "web", "properties": { "addressPrefix": "10.0.1.0/24" } },

{ "name": "db", "properties": { "addressPrefix": "10.0.2.0/24" } }

]

}

}

]

}

**Steps:**

1. Save this as vnet-template.json
2. Deploy using:

az deployment group create \

--resource-group <your-resource-group> \

--template-file vnet-template.json

**Expected Outcome:**

A Virtual Network named demo-vnet with two subnets: web and db.

**Lab 4: Deploy a Linux Virtual Machine**

**Objective:** Provision a basic Ubuntu VM using an ARM template.

**Template Snippet :**

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"adminUsername": { "type": "string" },

"adminPassword": { "type": "securestring" }

},

"resources": [

{

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

"apiVersion": "2023-03-01",

"name": "demoVM",

"location": "[resourceGroup().location]",

"properties": {

"hardwareProfile": { "vmSize": "Standard\_B1s" },

"storageProfile": {

"imageReference": {

"publisher": "Canonical", "offer": "UbuntuServer",

"sku": "18.04-LTS", "version": "latest"

},

"osDisk": {

"createOption": "FromImage"

}

},

"osProfile": {

"computerName": "demoVM",

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

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

},

"networkProfile": {

"networkInterfaces": [

{

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

}

]

}

}

}

]

}

**Steps:**

1. Prepare a network interface (demoNic) using CLI or another ARM template.
2. Deploy using az deployment group create.

**Expected Outcome:**

Ubuntu VM deployed with provided credentials.

**Lab 5: ARM Template with Outputs**

**Objective:** Capture and use outputs like resource ID, IP, etc.

**Sample Output Block:**

"outputs": {

"storageAccountId": {

"type": "string",

"value": "[resourceId('Microsoft.Storage/storageAccounts', parameters('storageAccountName'))]"

}

}

**Steps:**

1. Add to any previous template.
2. Deploy and use --query in CLI to extract outputs:

az deployment group create \

--resource-group <your-rg> \

--template-file azuredeploy.json \

--parameters @azuredeploy.parameters.json \

--query "properties.outputs"

**Expected Outcome:**

Deployment output values are printed after deployment completes.

**Lab 6: Use ARM Template from GitHub**

**Objective:** Deploy a prebuilt ARM template from a GitHub URL.

**Steps:**

1. Run the below CLI command:

az deployment group create \

--resource-group <your-rg> \

--template-uri "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-linux/azuredeploy.json"

1. Observe deployment in the portal.

**Expected Outcome:**

Ubuntu VM is deployed using the template directly from GitHub.