Online Lab - Deploying and Managing Virtual Machines (VMs)

Topic: Implementing Custom Azure VM Images

Scenario

Adatum Corporation wants to create custom Azure VM images

Objectives

After completing this lab, you will be able to:

- Install and configure HashiCorp Packer
- Create a custom VM image
- Deploy an Azure VM based on a custom image

Lab Setup

Estimated Time: 45 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Installing and configuring HashiCorp Packer

The main tasks for this exercise are as follows:

1. Download HashiCorp Packer
2. Configure HashiCorp Packer prerequisites

Task 1: Download HashiCorp Packer

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at http://portal.azure.com and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, in the Microsoft Edge window, start a **Bash** session within the **Cloud Shell**.

3. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:
    - Subsciption: the name of the target Azure subscription
    - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location
    - Resource group: **az3000300-LabRG**
    - Storage account: a name of a new storage account
    - File share: a name of a new file share
4. From the Cloud Shell pane, run the following to download the Packer compressed installation media:

```
wget https://releases.hashicorp.com/packer/1.3.1/packer_1.3.1_linux_amd64.zip
```

5. From the Cloud Shell pane, run the following to unzip the Packer installation media:

```
unzip packer_1.3.1_linux_amd64.zip
```

Task 2: Configure HashiCorp Packer prerequisites

1. From the Cloud Shell pane, run the following to create a resource group and store the JSON output in a variable (replace the `<Azure region>` placeholder with the name of the Azure region that is available in your subscription and which is closest to the lab location):

```
RG=$(az group create --name az3000301-LabRG --location <Azure region>)
```

2. From the Cloud Shell pane, run the following to create a service principal that will be used by Packer and store the JSON output in a variable:

```
AAD_SP=$(az ad sp create-for-rbac)
```

Result: After you completed this exercise, you have downloaded HashiCorp Packer and configured its prerequisites.

Exercise 2: Creating a custom image

The main tasks for this exercise are as follows:

1. Configure a Packer template
2. Build a Packer-based image

Task 1: Configure a Packer template

1. From the Cloud Shell pane, run the following to retrive the value of the service pricipal appId and store it in a variable

```
CLIENT_ID=$(echo $AAD_SP | jq .appId | tr -d '"')
```

2. From the Cloud Shell pane, run the following to retrive the value of the service pricipal password and store it in a variable

```
CLIENT_SECRET=$(echo $AAD_SP | jq .password | tr -d '"')
```

3. From the Cloud Shell pane, run the following to retrive the value of the service pricipal tenant ID and store it in a variable

```
TENANT_ID=$(echo $AAD_SP | jq .tenant | tr -d '"')
```

4. From the Cloud Shell pane, run the following to retrive the value of the subscription ID and store it in a variable:

```
SUBSCRIPTION_ID=$(az account show --query id | tr -d '"')
```

5. From the Cloud Shell pane, run the following to retrive the value of the resource group location and store it in a variable:

```
LOCATION=$(echo $RG | jq .location | tr -d '"')
```

1. From the Cloud Shell pane, upload the Packer template **F:\AZ300\Labs\03\template03.json** into the home directory.
2. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **client*id*** *parameter with the value of the* ***$CLIENT*ID** variable in the Packer template:

   ```
   sed -i.bak1 's/"$CLIENT_ID"/"'"$CLIENT_ID"'"/' ~/ubuntu.json
   ```

3. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **client*secret*** *parameter with the value of the* ***$CLIENT*SECRET** variable in the Packer template:

   ```
   sed -i.bak2 's/"$CLIENT_SECRET"/"'"$CLIENT_SECRET"'"/' ~/ubuntu.json
   ```

4. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **tenant*id*** *parameter with the value of the* ***$TENANT*ID** variable in the Packer template:

```
sed -i.bak3 's/"$TENANT_ID"/"'"$TENANT_ID"'"/' ~/ubuntu.json
```

5. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **subscription*id*** *parameter with the value of the* ***$SUBSCRIPTIONID*** variable in the Packer template:

```
sed -i.bak4 's/"$SUBSCRIPTION_ID"/"'"$SUBSCRIPTION_ID"'"/'
~/ubuntu.json
```

6. From the **Cloud Shell** pane, run the following to replace the placeholder for the value of the **location** parameter with the value of the **$LOCATION** variable in the Packer template:

```
sed -i.bak5 's/"$LOCATION"/"'"$LOCATION"'"/' ~/ubuntu.json
```

Task 2: Build a Packer-based image

1. From the Cloud Shell pane, run the following to retrive the value of the client ID and store it in a variable:

```
./packer build ubuntu.json
```

2. Monitor the built progress until it completes.

> **Note**: The build process might take about 10 minutes.

Result: After you completed this exercise, you have created a Packer template and used it to build a custom image.

Exercise 3: Deploying a custom image

The main tasks for this exercise are as follows:

1. Deploy an Azure VM based on a custom image
2. Validate Azure VM deployment

Task 1: Deploy an Azure VM based on a custom image

1. From the Cloud Shell pane, run the following to deploy an Azure VM based on the custom image.

`az vm create --resource-group az3000301-LabRG --name az3000301-vm --image az3000301-image --admin-username student --generate-ssh-keys` 1. Wait for the deployment to complete

NOTE: The deployment process might take about 3 minutes.

2. Once the deployment completes, from the Cloud Shell pane, run the following to allow inbound traffic to the newly deployed VM on TCP port 80:

```
az vm open-port --resource-group az3000301-LabRG --name az3000301-vm --port 80
```

Task 2: Validate Azure VM deployment

1. From the Cloud Shell pane, run the following to identify the IP address associated with the newly deployed Azure VM.

`az network public-ip show --resource-group az3000301-LabRG --name az3000301-vmPublicIP --query ipAddress` 1. Start Microsoft Edge and navigate to the IP address you identified in the previous step.

2. Verify that Microsoft Edge displays the **Welcome to nginx!** page.

Result: After you completed this exercise, you have deployed an Azure VM based on a custom image and validated the deployment.