

Deploying and Managing Virtual Machines (VMs)

Lab: 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:

- Create a custom VM image using HashiCorp Packer
- Deploy an Azure VM based on a custom image

Lab Setup

Estimated Time: 45 minutes

Interface: Use Azure Cloud Shell in BASH mode

Exercise 1: 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 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:
 - Subscription: 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 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>)
```

Note: To list Azure regions, run `az account list-locations --output table`

- 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)
```

- From the Cloud Shell pane, run the following to retrieve the value of the service principal appId and store it in a variable

```
CLIENT_ID=$(echo $AAD_SP | jq -r .appId)
```

- From the Cloud Shell pane, run the following to retrieve the value of the service principal password and store it in a variable

```
CLIENT_SECRET=$(echo $AAD_SP | jq -r .password)
```

- From the Cloud Shell pane, run the following to retrieve the value of the service principal tenant ID and store it in a variable

```
TENANT_ID=$(echo $AAD_SP | jq -r .tenant)
```

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

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

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

```
LOCATION=$(echo $RG | jq -r .location)
```

- From the Cloud Shell pane, upload the Packer template \allfiles\AZ-300T01\Module_03\template03.json into the home directory. To upload a file, click the document icon that has an up and down arrow in the Cloud Shell pane.

- 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"'/' ~/template03.json
```

- 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"'/' ~/template03.json
```

- 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"'/' ~/template03.json
```

- 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 **\$SUBSCRIPTION_ID** variable in the Packer template:

```
sed -i.bak4 's/"$SUBSCRIPTION_ID"/'"$SUBSCRIPTION_ID"'/' ~/template03.json
```

- 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"'/' ~/template03.json
```

Task 2: Build a Packer-based image

- From the Cloud Shell pane, run the following to build the packer-based image:

```
packer build template03.json
```

- 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 2: Deploying a custom image

The main tasks for this exercise are as follows:

- Deploy an Azure VM based on a custom image
- Validate Azure VM deployment

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

- 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
```

- Wait for the deployment to complete

Note: The deployment process might take about 3 minutes.

- 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

- 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
```

- Start Microsoft Edge and navigate to the IP address you identified in the previous step.
- 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.

Exercise 3: Remove lab resources

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

```
az group list --query "[?starts_with(name,'az3000301')].name --output tsv
```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name,'az3000301')].name --output tsv  
| xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Result: In this exercise, you removed the resources used in this lab.