

## Module 2: Foundational Components of Microsoft Azure Stack

### Lab: Reviewing the Azure Stack Infrastructure

## Exercise 1: Reviewing the Azure Stack Infrastructure Components

### Task 1: Azure Stack Active Directory

1. On the desktop, double-click the **Connectoid** icon. This will initiate a VPN connection to your Azure Stack once it has been running for a sufficient amount of time. Wait for the connection to be made before continuing on.
2. In the **User Account Control** dialog, click **Yes**
3. Click **Start**, and then click **Windows Administrative Tools**.
4. In the **Administrative Tools** window that opens, double-click **Active Directory Users and Computers**.
5. In the **Active Directory Users and Computers** window that opens, expand **azurestack.local**, and then click **Users**.
6. Review the **Security Groups** that begin with **AzS**.
7. Click **Managed Service Accounts** and review the **AzS** accounts that have been created by **Azure Stack**.
8. Click **Service Principals** and review the User accounts created by **Azure Stack**.
9. Click **Computers** and review the computer accounts in the **azurestack.local** domain.
10. Click **Domain Controllers** and review the **Domain Controller** accounts in the **azurestack.local** domain.
11. Close the **Active Directory Users and Computers** window.

### Task 2: Azure Stack Storage Spaces

1. On the desktop, double-click **Enter-PSSession**
2. In the **User Account Control** dialog, click **Yes**

3. In the **Administrator: Enter-PSSession** window that opens, type the following command, and then press Enter on the keyboard:

```
Get-Cluster | Format-List *
```

4. Review the **Cluster** information that is returned.
5. Type the following command and then press Enter on the keyboard.

```
Get-ClusterStorageSpacesDirect | Format-List *
```

6. Review the **Storage Spaces Direct** cluster information that is returned.
7. Type the following command and then press Enter on the keyboard.

```
Get-PhysicalDisk
```

8. Review the **Disk** information that is returned.

### **Task 3: Azure Stack Network Controller**

**Note:** In this task, you connect to an Azure Stack infrastructure virtual machine using Hyper-V Manager. In the case where multiple users are accessing the same Azure Stack host, only 1 connection is available. In this scenario, the instructor will perform this task and students should review the instructors screen instead of performing the task themselves.

1. In the open **PowerShell** window type the following command, and then press Enter:

```
Enter-PSSession -vmname azs-nc01
```

2. When prompted, enter the following credentials

```
AzureStack\AzureStackAdmin  
Pa55w.rd
```

3. Type the following command, and then press enter:

```
Get-NetworkController
```

8. Review the **Network Controller** information that is returned.

9. Close the **Windows PowerShell** window.

#### **Task 4: Azure Stack Software Load Balancer**

1. Open Internet Explorer.
2. Browse to <https://adminportal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:
  - Username: **AzureStack\CloudAdmin**
  - Password: **Pa55w.rd**
4. In the Microsoft **Azure Stack - Administration**, page that opens, click **More Services**.
5. Type **balancer** in the text box, and then click **Load Balancers**.
6. In the **Load balancers** page that opens, click **FrontEndServersLoadBalancer**
7. Review the **Load Balancer** information, provided.
8. Repeat Steps 6 and 7 for the **ManagementServersLoadBalancer** and **PublishersLoadBalancer**
9. Close the **Azure Stack** Portal.

**Results:** After this exercise, you should have used the administrative tools on the Hyper-V host to confirm availability of the Azure Stack infrastructure components.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No

real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 3: Deploying Microsoft Azure Stack

Lab: Confirming prerequisites and installing Microsoft Azure Stack

## Exercise 1: Confirming the Prerequisites

### Task 1: Confirm the CPU meets the requirements

1. On the desktop, double-click **Enter-PSSession**
2. In the **User Account Control** dialog, click **Yes**
3. In the **Administrator: Enter-PSSession** Console enter the following PowerShell command and press Enter:

```
Get-CimInstance -ClassName Win32_Processor -Property Name,  
SecondLevelAddressTranslationExtensions, VirtualizationFirmwareEnabled,  
VMMonitorModeExtensions |
```

```
Format-List Name, SecondLevelAddressTranslationExtensions,  
VirtualizationFirmwareEnabled, VMMonitorModeExtensions
```

4. In the information that is returned, review the following:
  - o SecondLevelAddressTranslationExtensions
  - o VirtualizationFirmwareEnabled
  - o VMMonitorModeExtensions

6. Enter the following PowerShell command and press Enter:

```
Get-CimInstance -ClassName Win32_Processor -Property Name, SocketDesignation,  
NumberOfCores, NumberOfEnabledCore, NumberOfLogicalProcessors | Format-List  
Name, SocketDesignation, NumberOfCores, NumberOfEnabledCore,  
NumberOfLogicalProcessors
```

8. Ensure the **NumberOfCores** for each CPU returned is **6** or higher.
9. Leave the **Administrator: Windows PowerShell** Console open.

## Task 2: Confirm the number of drives

1. Due to the virtual-hosted solution for this lab, this task has been removed. Please continue to the next task.

## Task 3: Confirm the quantity of memory

1. In the **Administrator: Enter-PSSession** window, type the following command and press Enter:

```
$Memory = (Get-Ciminstance Win32_OperatingSystem | Select-Object -ExpandProperty MaxProcessMemorySize) /1GB
```

2. Type the following command to see the amount of memory installed and press enter:

```
$Memory
```

3. To round the number, enter the following command and press enter:

```
[math]::Round($Memory,0)
```

4. This will display the amount of memory in GB.

**Results:** After this exercise, you should have confirmed the hardware on which the Azure Stack Development Kit is to be deployed meets the prerequisites.

## Exercise 2: Confirm the installation was successful

### Task 1: Connect to Azure Stack

1. If not already connected to your Azure Stack instance, double-click the **Connectoid** icon on the desktop (If you have already done this, please continue to the next task).

### Task 2: Ensure both the User and Operator Portals are functioning

1. Open **Internet Explorer**.
2. Browse to <https://adminportal.local.azurestack.external>

3. Sign into **Azure Stack** with the following credentials:
  - Username: **AzureStack\CloudAdmin**
  - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack - Administration**, page that opens, click **More Services**.
5. Under **Administrative Resources** click **Resource provider manifests**.
6. In the **Resource provider manifests** blade, ensure the following **Resource Provider Manifests** are listed:
  - Microsoft.Compute—local--admin
  - Microsoft.SQLAdapter
  - Microsoft.Web.Admin
  - Microsoft.Network—local--admin
  - Microsoft.KeyVault—local—admin
  - Microsoft.Storage—local--admin
7. Open a new tab in **Internet Explorer** and browse to <https://portal.local.azurestack.external>
8. In the navigation pane, click **More services**.
9. Confirm the list of **Services** returned include the following:
  - Dashboard
  - Resource groups
  - All resources
  - Tags
  - Recent
  - Subscriptions

- Marketplace
- Resource explorer
- Portal settings
- Audit logs

#### 10. Close **Internet Explorer**.

**Results:** After this exercise is complete, you will be able to determine if the Azure Stack Development Kit has been successfully installed and is operating as required.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 4: Offering Microsoft Azure Stack Resources

Lab: Managing Offers and Plans in Microsoft Azure Stack

## **Exercise 1: Configuring Plans**

### **Task 1: Creating a Plan**

1. Open **Internet Explorer**.
2. Browse to <https://adminportal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:

- Username: **AzureStack\CloudAdmin**
  - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack - Administration**, page that opens, click **Plans**
  5. Click **Add**.
  6. In the **New Plan** blade that opens complete the form using the following settings, all other settings should remain default:
    - Display Name: **Virtual-Machines**
    - Resource Group: **Create new**
    - Resource Group Name: **RG-OffersAndPlans**
    - Services: **Microsoft.Compute**, **Microsoft.Network** and **Microsoft.Storage**
    - Quotas: Accept default **Quotas**.
  7. Click **Create** to create the **Plan**.
  8. In the **Plans** blade that opens confirm the **Virtual-Machines** Plan is listed. If it is not listed, click the **Refresh** button at the top of the blade.

## **Task 2: Creating an Add-On Plan**

1. In the **Azure Stack Portal**, from the navigation pane click **Plans**.
2. Click **Add**.
3. In the **New Plan** blade that opens complete the form using the following settings, all other settings should remain default:
  - Display Name: **Virtual-Machines-Add-On**
  - Resource Group: **Use existing**
  - Resource Group Name: **RG-OffersAndPlans**
  - Services: **Microsoft.Compute**
4. Click **Quotas**.

5. Click **Microsoft.Compute (local)** and then click **Create new quota**.
6. In the **Name** box, type **Additional-VMs**.
7. In the **Max number of virtual machines** box, remove **50** and then type **5**.
8. Click **OK** and then click **Additional-VMs** to select this **Quota**. Then click **OK**.
9. Click **Create** to create the **Plan**.
10. In the **Plans** blade that opens confirm the **Virtual-Machines-Add-On** Plan is listed. If it is not listed, click the **Refresh** button at the top of the blade.

**Results:** After this exercise, you should have created a plan that includes the following services: Network. Storage. Compute. You should have also configured the service quotas such that a maximum of five virtual machines can be created per tenant.

## Exercise 2: Configuring Offers

### Task 1: Creating an Offer

1. In the **Azure Stack Operator Portal**, from the navigation pane click **Offers**.
2. Click **Add**.
3. In the **New Offer** blade that opens configure the following settings. All other settings should remain as default:
  - Display Name: **Virtual Machines**
  - Provider Subscription: **Default Provider Subscription**
  - Resource Group: **Use existing**
  - Resource Group Name: **RG-OffersAndPlans**
  - Base plans: **virtual-machines**
4. Click **Create** to create the **Offer**.
5. In the **Offers** blade that opens, confirm the **virtual-machines** Offer is listed. If it is not listed, click the **Refresh** button at the top of the blade.

## Task 2: Adding an Add-On Plan to an Offer

1. In the **Azure Stack Operator Portal**, from the navigation pane, click **Offers**.
2. In the **Offers** blade that opens, click **virtual-machines**.
3. In the **virtual-machines** page that opens, scroll down and then click **Add On Plans**.
4. Click **Add**.
5. In the **Plan** blade that opens click **virtual-machines-add-on** and then click **Select**.
6. Close the **virtual-machines Add On Plans** page.
7. Close the **Offers** blade.

**Results:** After this exercise, you should have created a new offer that includes the virtual machines plan. You should have also associated the virtual machines add-on plan with the offer.

## Exercise 3: Subscribing to an Offer

### Task 1: Changing an Offers Status

1. In the **Azure Stack** Operator Portal, from the navigation pane, click **Offers**.
2. In the **Offers** blade that opens click **virtual-machines**.
3. From the top of the blade, click **Change State**.
4. In the **Change State** drop-down list click **Public**.
5. Close the **virtual-machines** blade.

### Task 2: Subscribing to an Offer

1. In **Internet Explorer**, click the settings button on the toolbar.
2. Navigate to **Safety** and then click **InPrivate Browsing**.

3. In the **InPrivate** address bar type <https://portal.local.azurestack.external> and then press Enter.
4. In the **Sign in** dialog, in Username box type **AzureStack\CloudAdmin** in the **Password** box type **Pa55w.rd**.
5. Click **Sign in**.
6. In the **Microsoft Azure Stack** User Portal that opens navigate to **More Services**, and click on **Subscriptions**.
7. In the **Subscriptions** blade, click **Add**.
8. In the **Get a Subscription** blade that opens click **Select an Offer**.
9. In the **Choose an Offer** page that opens click **Virtual Machines**.
10. Click **Create** and then click **Refresh**.
11. Click **More Services** and then click **Subscriptions**.
12. Confirm the **Virtual Machines** Subscription is listed. If it is not listed, press **F5** to refresh the entire browser window.
13. Do not close **the InPrivate** browsing window.

### **Task 3: Adding an Add-On Plan to an Offer**

1. In the **Subscriptions** blade click **Virtual Machines**.
2. In the **Virtual Machines** blade that opens, click **Add Plan**.
3. In the **Add Plan** blade that opens click **Virtual-Machines-Add-On**.
4. Confirm in the **Virtual-Machines** blade that the **Add-on Plans** tile now shows **1**.
5. Close all open windows.

**Results:** After this exercise, you should have changed the Offers Status from **Private** to **Public**. You should have then signed into the portal as a tenant and subscribed to the virtual machines offer.

### **Exercise 4: Delegating Offers**

## Task 1: Create the delegated provider and user roles for Olympia

1. Click **Start**, then click **Windows Administrative Tools**.
2. In the **Administrative Tools** window that opens, double-click **Active Directory Users and Computers**.
3. In the **Active Directory Users and Computers** window that opens, expand **AzureStack.local** and then click **Users**.
4. Right-click **Users**, then click **New**, then click **User**.
5. Configure the **New Object – User wizard** with the following settings, all other setting should remain as default (For the purposes of this lab, the Active Directory Domain Services verification warning can be safely ignored, click **OK**):
  - First name: **OlympiaDP**
  - User logon name: **OlympiaDP**
  - Password: **Pa55w.rd**
  - User must change password at next logon: **Deselect**
  - User cannot change password: **Select**
  - Password never expires: **Select**
6. Complete the wizard.
7. Right-click **Users**, then click **New**, then click **User**.
8. Configure the **New Object – User wizard** with the following settings, all other setting should remain as default (For the purposes of this lab, the Active Directory Domain Services verification warning can be safely ignored, click **OK**):
  - First name: **OlympiaUser**
  - User logon name: **OlympiaUser**
  - Password: **Pa55w.rd**
  - User must change password at next logon: **Deselect**
  - User cannot change password: **Select**

- Password never expires: **Select**
9. Complete the wizard and then close **Active Directory Users and Computers**.

### **Task 2: Create a plan that will be used to delegate an offer to Olympia**

1. Open **Internet Explorer**.
2. Browse to <https://adminportal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:
  - Username: **AzureStack\CloudAdmin**
  - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack - Administration**, page that opens, click **Plans**
5. Click **Add**.
6. In the **New Plan** page that opens configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaProviderPlan**
  - Resource Group: Create a new Resource Group named **DelegationRG**
  - Services: **Microsoft.Subscriptions**
  - Quotas: **delegatedProviderQuota**
7. Click **Create** to create the plan.
8. Click the **Refresh** button at the top of the page until the **OlympiaProviderPlan** is displayed.
9. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.

### **Task 3: Create an offer that will be used to provide delegation services to Olympia**

1. In the **Microsoft Azure Stack - Administration** page, click Offers.

2. Click **Add**.
3. In the **New Offer** page that opens configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaProviderOffer**
  - Resource Group: **DelegationRG**
  - Base Plans: **OlympiaProviderPlan**
4. Click **Create** to create the **Offer**.
5. Click the **Refresh** button at the top of the page until the **OlympiaProviderOffer** is displayed.
6. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.

#### **Task 4: Add the Olympia delegated provider as a subscriber to the offer**

1. In the **Microsoft Azure Stack - Administration**, click **More Services**.
2. Under **Administrative Resources** click **User Subscriptions**.
3. In the **User Subscriptions** page that opens, click **Add**.
4. In the **New User Subscription** page that opens configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaSubscription**
  - User: [OlympiaDP@azurestack.local](mailto:OlympiaDP@azurestack.local)
  - Offer: **OlympiaProviderOffer**
5. Click **Create** to create the User Subscription.
6. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.

#### **Task 5: Create a plan that includes the services that Olympia will offer**

1. In the **Microsoft Azure Stack - Administration**, click **Plans**

2. Click **Add**.
3. In the **New Plan** page that opens configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaPlan**
  - Resource Group: **DelegationRG**
  - Services: **Microsoft.Compute**, **Microsoft.Network** and **Microsoft.Storage**
  - Quotas: Select **Default Quota** for all services
4. Click **Create** to create the plan.
5. Click the **Refresh** button at the top of the page until the **OlympiaPlan** is displayed.
6. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.

#### **Task 6: Create an offer that will be used to delegate services to Olympia**

1. In the **Microsoft Azure Stack - Administration**, click **Offers**.
2. Click **Add**.
3. In the **New Offer** page that opens configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaOffer**
  - Resource Group: **DelegationRG**
  - Base Plans: **OlympiaPlan**
4. Click **Create** to create the **Offer**.
5. Click the **Refresh** button at the top of the page until the **OlympiaOffer** is displayed.
6. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.

### Task 7: Delegate the offer to Olympia

1. In the **Microsoft Azure Stack - Administration**, click **Offers**.
2. In the **Offers** page that opens, click **OlympiaOffer**.
3. In the **OlympiaOffer** page that opens, click **Delegated Providers**.
4. In the **Delegated Providers** page that opens, click **Add**.
5. In the **Delegate Offer** page that opens, click the drop-down list under **Pick Delegated Provider Subscription** and then click **OlympiaSubscription**.
6. Click **Delegate**.
7. Click the **Microsoft Azure Stack - Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.
8. Sign-out of the **Azure Stack** portal.
9. Close **Internet Explorer**

### Task 8: Create an offer using the delegated provider

1. Open **Internet Explorer**.
2. Browse to <https://portal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:
  - Username: **AzureStack\OlympiaDP**
  - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack**, page that opens, click **Offers**.
5. In the **Offers** page that opens, click **Add**.
6. In the **New Offer** page that opens, configure the following settings, all other settings should remain as default:
  - Display Name: **OlympiaDelegatedOffer**
  - Resource Group: Create a new Resource Group named **OlympiaRG**

- Delegated Offer: **OlympiaOffer<randomNumber>**
7. Click **Create** to create the Offer.
  8. Click the **Refresh** button at the top of the page until the **OlympiaDelegatedOffer** appears.
  9. Click the **Microsoft Azure Stack** link to return to the home page of the **Azure Stack** portal.

### **Task 9: Change the offers status to public**

1. In the **Microsoft Azure Stack** portal that was left open from the previous task, click **Offers**.
2. In the **Offers** page that opens, click the **OlympiaDelegatedOffer** offer.
3. In the **OlympiaDelegatedOffer** page that opens, click **Change State** and then click **Public**.
4. Click the **Microsoft Azure Stack** link to return to the home page of the **Azure Stack** portal.

### **Task 10: Obtain the delegated portal URL**

1. In the **Microsoft Azure Stack** portal that was left open from the previous task, click **More Services** and then click **Subscriptions**.
2. In the **Subscriptions** page that opens, click **OlympiaSubscription**.
3. In the **OlympiaSubscription** page that opens, click **Properties**.
4. In the **OlympiaSubscription – Properties** page that opens, copy the contents of the **Portal URL** box and then paste it into a new **Notepad** document.
5. Sign-out of the **Azure Stack** Portal and close **Internet Explorer**.

### **Task 11: Sign-up to the delegated Offer and confirm delegated services**

1. Open **Internet Explorer** and browse to the **URL** that was pasted into the **Notepad** document that was left open in the previous task.
2. Login to the **Azure Stack** user portal with the following credentials:

- User name: **AzureStack\OlympiaUser**
  - Password: **Pa55w.rd**
3. In the **Microsoft Azure Stack** portal that opens click **Get a Subscription** from the home page.
  4. In the **Get a Subscription** page that opens type **OlympiaOffer** in the **Display Name** box.
  5. Click **Offer**, then in the **Choose an Offer** page that opens, click **OlympiaDelegatedOffer** and then click **Create**.
  6. Click **Refresh** to refresh the page.
  7. From the **Azure Stack** portal home page, click **New**, then click **Compute**, then click **Windows Server 2016 Datacenter Core Eval**.
  8. In the **Basics** page that opens, configure the following settings:
    - Name: **OlympiaUserVM**
    - Username: **OlympiaUser**
    - Password: **Pa55w.rd12345**
    - Confirm Password: **Pa55w.rd12345**
    - Resource Group: Create a new **Resource Group** named **OlympiaRG**
  9. Click **OK**.
  10. In the **Size** page that opens, click **A1 Standard** and then click **Select**.
  11. In the **Settings** page that opens, click **OK**.
  12. In the **Summary** page that opens, click **OK**.
  13. From the **Azure Stack** portal home page, click **Virtual Machines**.
  14. Wait until the **OlympiaUserVM** VM appears. This confirms the VM is being created.
  15. Optionally wait until the status of the VM changes from **Creating** to **Running**. This confirms the VM has been created successfully.

16. Sign-out of the **Azure Stack** portal.

17. Close Internet Explorer.

**Results:** After this exercise, you should have created a delegated provider in Azure Stack and associated the appropriate subscription such that offers and plans can be subscribed to by Olympia users. You should have also confirmed that an Olympia user can successfully use Azure Stacks user portal to provision a virtual machine.

## Exercise 5: Creating a new Marketplace Item

### Task 1: Use an existing Marketplace Item

1. Navigate to the **C:\Media\ModuleFiles\Module4\Azure Stack Marketplace Item Generator and Sample\Contoso.StockCheck\strings** folder and right-click the **resources.json** file, and then click **Open with**, and then click **Notepad**.
2. Edit the third, fourth and fifth lines as shown below:
  - "summary": "Check stock availability and price",
  - "longSummary": "This application can be used to check availability of Contoso's stock and also get an up-to-date price on each stock item",
  - "description": "This application can be used to check availability of Contoso's stock and also get an up-to-date price on each stock item",
6. Save and close the **resources.resjson** file.
7. Navigate to the **C:\Media\ModuleFiles\Module4\Azure Stack Marketplace Item Generator and Sample\Contoso.StockCheck** folder and right-click the **manifest.json** file and then click **Open with**, and then click **Notepad**.
8. Edit the lines as shown below:
  - "name": "StockCheck",
  - "categories":["Contoso"]

9. Save and close the **manifest.json** file.

## **Task 2: Package the new Marketplace Item**

1. Right-click **Start** and then click **Command prompt (Admin)**.
2. Click **Yes** on the **User Account Control** window that opens.
3. In the **Administrator: Command Prompt** window that opens type the following command:

```
cd C:\media\ModuleFiles\Module04\Azure Stack Marketplace Item Generator and Sample\AzureGalleryPackageGenerator
```

4. Type the following command and then press enter on the keyboard:

```
AzureGalleryPackager.exe package -m "C:\media\modulefiles\module04\Azure Stack Marketplace Item Generator and Sample\Contoso.StockCheck\manifest.json" -o "C:\media\modulefiles\module04\Azure Stack Marketplace Item Generator and Sample\Contoso.StockCheck"
```

5. Close the **Administrator: Command Prompt** window.
6. Navigate to **C:\Media\ModuleFiles\Module4\Azure Stack Marketplace Item Generator and Sample\Contoso.StockCheck** and confirm the **Contoso.StockCheck.1.0.0.azpkg** file is present.

## **Task 3: Publish the Marketplace Item in Azure Stack**

1. Click **Start** then right-click **Windows PowerShell** and then click **Run as Administrator**.
2. If a **User Account Control** window opens, click **Yes**.
3. In the **Administrator: Windows PowerShell** window that opens type the following commands pressing enter after each command:

```
Import-Module C:\AzureStack-Tools-master\Connect\AzureStack.Connect.psm1
```

```
Add-AzureRmEnvironment -Name CloudAdmin -ArmEndpoint  
"https://adminmanagement.local.azurestack.external"
```

```
$Password = ConvertTo-SecureString "Pa55w.rd" -AsPlainText -Force
```

```
$Cred = New-Object System.Management.Automation.PSCredential  
("AzureStack\CloudAdmin", $Password)
```

```
$tenantID = Get-AzsDirectoryTenantID -ADFS -EnvironmentName CloudAdmin
```

4. Enter the following **PowerShell** command and press Enter to login to **Azure Stack**:

```
Login-AzureRmAccount -EnvironmentName "CloudAdmin" -TenantId $tenantID -  
Credential $Cred
```

5. If a **Select Y to enable data collection [Y/N]** prompt opens, type **N** and then press enter.

6. Type the following commands pressing enter after each command:

```
$subscriptionid = (Get-AzureRmSubscription -SubscriptionName 'Default Provider  
Subscription').SubscriptionId
```

```
Select-AzureRMSubscription -SubscriptionID $Subscriptionid
```

```
New-AzureRmResourceGroup -Name azurestack -Location local
```

```
New-AzureRmStorageAccount -ResourceGroupName azurestack -Name  
azurestackstorage -Type Standard_LRS -location local
```

```
Set-AzureRMCurrentStorageAccount -ResourceGroupName azurestack -  
StorageAccountName azurestackstorage
```

```
$StorageAccount = Get-AzureRmStorageAccount -ResourceGroupName azurestack -  
Name azurestackstorage
```

```
$GalleryContainer = New-AzureStorageContainer -Name gallery -Permission Blob -  
Context $StorageAccount.Context
```

```
$AzureStackPackage = $GalleryContainer | Set-AzureStorageBlobContent -File  
"C:\media\modulefiles\module04\Azure Stack Marketplace Item Generator and  
Sample\Contoso.StockCheck\Contoso.StockCheck.1.0.0.azpkg"
```

```
Add-AzSGalleryItem -GalleryItemUri
```

```
$AzureStackPackage.ICloudBlob.StorageUri.PrimaryUri.AbsoluteUri
```

7. After entering the final command, you should receive a **Status Code** of **Created**.

8. Close the **Administrator: Windows PowerShell** window.

#### **Task 4: Confirm the new Marketplace Item is available in the Azure Stack Portal**

1. Open **Internet Explorer**.
2. Browse to <https://adminportal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:
  - Username: **AzureStack\CloudAdmin**
  - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack - Administration**, page that opens, click **New**
5. Notice the new **Contoso Marketplace** item is now displayed.
6. Click **Contoso** and review the information provided in the **Contoso** blade that opens.
7. Close the Azure Stack Portal.

**Results:** After this exercise, you should have created a new Marketplace item for Contoso's StockCheck application. You should have also published the Marketplace item in Azure Stack. Finally, you should have confirmed that tenants can select the new Marketplace item when they sign into the Azure Stack Portal.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

Module 5: Microsoft Azure Stack and DevOps

Lab: Debugging and Deploying ARM Templates

## Exercise 1: Using Windows PowerShell to obtain API versions

### Task 1: Obtaining the Microsoft.Compute and Microsoft.Storage API versions

1. Click **Start** and navigate to **Windows PowerShell**.
2. Right-click **Windows PowerShell**.
3. Navigate to **More | Run as administrator**.
4. In the **User Account Control** dialog, click **Yes**.
5. In the **Administrator: Windows PowerShell** Console, enter the following PowerShell command and press Enter to import the required PowerShell Module:

```
Import-Module C:\AzureStack-Tools-master\Connect\AzureStack.Connect.psm1
```

7. Enter the following PowerShell commands to obtain the GUID for the Active Directory registration and add the Azure Administration environment to PowerShell. Press Enter at the end of each PowerShell command:

```
$Password = ConvertTo-SecureString "Pa55w.rd" -AsPlainText -Force
```

```
$Cred = New-Object System.Management.Automation.PSCredential  
("AzureStack\CloudAdmin", $Password)
```

```
Add-AzureRmEnvironment -Name CloudAdmin -ArmEndpoint  
"https://adminmanagement.local.azurestack.external"
```

```
$tenantID = Get-AzsDirectoryTenantID -ADFS -EnvironmentName CloudAdmin
```

8. Enter the following PowerShell command and press Enter to login to the Tenant environment:

```
Login-AzureRmAccount -EnvironmentName CloudAdmin -TenantId $tenantID -  
Credential $Cred
```

9. Enter the following PowerShell command to return the Subscriptions that are associated with the Tenant environment and press Enter:

Get-AzureRmSubscription

10. In the returned information ensure the value of **SubscriptionName** is **Default Provider Subscription**.

11. Enter the following PowerShell and press Enter to return the Microsoft.Compute API's that are available:

```
(Get-AzureRmResourceProvider -ProviderNamespace Microsoft.Compute).ResourceTypes | Where-Object -Property ResourceTypeName -EQ -Value virtualMachines | Select-Object -ExpandProperty ApiVersions
```

12. Enter the following PowerShell and press Enter to return the Microsoft.Storage API's that are available:

```
(Get-AzureRmResourceProvider -ProviderNamespace Microsoft.Storage).ResourceTypes | Where-Object -Property ResourceTypeName -EQ -Value storageaccounts | Select-Object -ExpandProperty ApiVersions
```

13. Close the **Windows PowerShell** Console.

**Results:** After completing this exercise, you will be able to: Install AzureRM and Azure Stack PowerShell from the NuGet repository. Configure Azure PowerShell to connect to Azure Stack.

## Exercise 2: Debug an ARM Template with Visual Studio

### Task 1: Debug the ARM Template

1. Click **Start**.
2. Type **Visual Studio**.
3. Right-click **Visual Studio 2017** and then click **Run as Administrator**.
4. If a **User Account Control** window appears, click **Yes**.
5. In the **Sign in to Visual Studio** dialog, sign in using a Microsoft account. If you do not have one, create one for the purposes of this lab. This is required to activate **Visual Studio Community**.
6. On the menu bar, click **File**

7. Navigate to **Open**, then click **Project/Solution**
8. In **Open Project** dialog, navigate to **C:\Media\ModuleFiles\Module05\VirtualNetwork**
9. Select the **VirtualNetwork.deployproj** file and click **Open**
10. Click the **File** menu and then click **Save All**.
11. If prompted, accept the default location and then click **Save**.
12. In the **Solution Explorer** pane, double-click **azuredeploy.json**
13. If an **Update Available** window appears, click **Not Now**.
14. In the **azuredeploy.json** central pane, navigate to **line 81**
15. Change the content of line 81 from:

```
"location": "[variables('AzureRegion')]",
```

To

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

18. Click **Save** on the toolbar
19. The **Template** is now valid.
20. Do not close **Visual Studio**.

## **Task 2: Review and update the Parameters File and validate the ARM template**

1. In **Visual Studio**, from the **Solution Explorer**, double click **azuredeploy.parameters.json**.
2. In the central pane change the contents of the file from by adding the **EnvironmentName** parameter as shown below:

```
{
```

```
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json",
```

```
"contentVersion": "1.0.0.0",
```

```
"parameters": {  
  "DeploymentDate": {  
    "value": "2017-02-07"  
  }  
}  
  
to  
  
{  
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "DeploymentDate": {  
      "value": "2017-02-07"  
    },  
    "EnvironmentName": {  
      "value": "QA"  
    }  
  }  
}
```

3. From the toolbar, click the **Save** button.
4. Close **Visual Studio**.
5. Click **Start**.
6. On the taskbar, right-click **Windows PowerShell** and click **Run as Administrator**

7. In the **Windows PowerShell** Console enter the following PowerShell command and press Enter:

```
Import-Module C:\AzureStack-Tools-master\Connect\AzureStack.Connect.psm1
```

9. Enter the following PowerShell command and press Enter:

```
Set-Location 'C:\Media\ModuleFiles\Module05\VirtualNetwork'
```

10. Enter the following PowerShell commands to obtain the GUID for the Active Directory registration and add the Azure Administration environment to PowerShell. Press Enter at the end of each PowerShell command:

```
$Password = ConvertTo-SecureString "Pa55w.rd" -AsPlainText -Force
```

```
$Cred = New-Object System.Management.Automation.PSCredential  
("AzureStack\CloudAdmin", $Password)
```

```
Add-AzureRmEnvironment -Name AzureStack -ArmEndpoint  
"https://management.local.azurestack.external"
```

```
$tenantID = Get-AzsDirectoryTenantID -ADFS -EnvironmentName AzureStack
```

11. Enter the following PowerShell command and press Enter to login to the Tenant environment:

```
Login-AzureRmAccount -EnvironmentName AzureStack -TenantId $tenantID -Credential  
$Cred
```

12. Enter the following PowerShell command to return the Subscriptions that are associated with the Tenant environment and press Enter:

```
Select-AzureRmSubscription -SubscriptionName "Virtual Machines"
```

13. Enter the following PowerShell command and press Enter:

```
New-AzureRmResourceGroup -Name "RG-VNet" -Location local
```

```
Test-AzureRmResourceGroupDeployment -ResourceGroupName RG-VNet -  
TemplateFile .\azuredeploy.json -TemplateParameterFile .\azuredeploy.parameters.json -  
Mode Incremental -Verbose
```

14. Close the **Window PowerShell** console.

**Results:** After completing this exercise, you will be able to debug ARM Templates in Visual Studio, review and update parameter files for deployments and validate ARM Templates prior to deployment.

## Exercise 3: Deploy a virtual machine using Azure CLI

### Task 1: Export the Azure Stack CA root certificate

2. On the desktop of the **Azure Stack Lab** VM, double-click the **RDP-AzS-Host** icon, at the user account control click **Yes**
3. Enter the following credentials:
  - AzSAdmin
  - Pa55w.rdPa55w.rd
4. Click **Yes** on the certificate warning window
5. If prompted to install updates, ignore and do not install them.
6. On the taskbar, click on **Hyper-V Manager**
7. Double-click to connect to the **AzS-Host** VM
8. Log in using the following credentials:
  - Azurestackadmin
  - Pa55w.rd
9. In the **AzS-Host on EdX-AzS-Host** window, click Start, then right-click **Windows PowerShell**.
10. Click **More then click Run as administrator**.
11. In the **User Account Control** dialog, click **Yes**.
12. In the **Administrator: Windows PowerShell** window that opens, type the following commands:

```
Set-Location C:\media\modulefiles\Module05
```

```
.\ExportRootCACert.ps1
```

13. Confirm the **C:\Media\ModuleFiles\Module05** folder now contains the **root.pem** file.
14. Leave the **Administrator: Windows PowerShell** window open.

### **Task 2: Trust the Azure Stack CA root certificate**

1. Edit the **C:\Media\MdouleFiles\\Module05\ TrustCARootCert.ps1** file with **Notepad** and change the following line from this:

```
$pemFile = "<Fully qualified path to the PEM certificate Ex:  
C:\Student1\Module5\root1.pem>"
```

2. To this:

```
$pemFile = "C:\Media\ModuleFiles\Module05\root.pem"
```

3. Save and then close the file.
4. In the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command and then press enter:

```
.\TrustCARootCert.ps1
```

5. Confirm the **Python Cert** store was updated for allowing the azure stack CA root certificate message appears.
6. Leave the **Administrator: Windows PowerShell** window open.

### **Task 3: Upload the VM Alias file to Azure Stack**

1. Minimize the **AzS-Host** RDP session
2. On the **Azure Stack Lab** VM, open **Internet Explorer**.
3. Browse to <https://adminportal.local.azurestack.external>
4. Sign into **Azure Stack** with the following credentials:
  - o Username: **AzureStack\CloudAdmin**
  - o Password: **Pa55w.rd**

5. In the **Microsoft Azure Stack - Administration**, page that opens, click **More Services**, then click **Storage accounts**.
6. In the **Storage accounts** page that opens, click **samslmedia**.
7. In the **samslmedia** page that opens, click **Containers**.
8. In the **samslmedia – Containers** page that opens, click **+ Container**.
9. In the **Name** box type **json**.
10. Click the **Access type** drop-down list and then click **Blob**.
11. Click **OK**.
12. Click the **json** container.
13. In the **json** page that opens click **Upload**.
14. In the **Upload** page that opens, click the icon to the right of the **File** box.
15. In the **Choose file to upload** window that opens, navigate to **C:\Media\ModuleFiles\Module05** and then click the **VMAlias.json** file and then click **Open**.
16. Click **Upload**.
17. Close the **Upload blob** page.
18. Click the **VMAlias.json** blob.
19. In the **Blob** properties page that opens copy the **URL** and then paste it into a new **Notepad** document.
20. Click the **Microsoft Azure Stack – Administration** link at the top of the page to return to the home page of the **Azure Stack** operator portal.
21. Leave the **Notepad** document open.

#### **Task 4: Create a Service Principal for AD FS**

1. Open the **AzS-Host** RDP window

2. Type the following command in the open PowerShell window in the open **AzS-Host on EdX-AzS-Host** window and then press enter:

```
$creds = Get-Credential
```

3. In the **Windows PowerShell credential** request window that opens, type **AzureStack\CloudAdmin** in the **User name** box, type **Pa55w.rd** in the **Password** box and then click **OK**.
4. Type the following command and then press enter.

```
$session = New-PSSession -ComputerName AzS-ERCS01 -ConfigurationName PrivilegedEndpoint -credential $creds
```

5. Wait until the command completes and then enter the following commands, pressing enter after each command.

```
$cert = New-SelfSignedCertificate -CertStoreLocation "cert:\CurrentUser\My" -Subject "CN=StudentCert" -KeySpec KeyExchange
```

```
Invoke-Command -Session $session -ScriptBlock { New-GraphApplication -Name 'StudentApp' -ClientCertificates $using:cert }
```

```
$session|remove-psession
```

6. Make a note of the **ClientID** and **ApplicationName** values that are returned by copying the values into the **Notepad** document that was left open in the previous task.

### **Task 5: Create a PEM file of the Service Principal certificate**

1. In the **AzS-Host on EdX-AzS-Host** window Navigate to **C:\Media\ModuleFiles\Module05** and then double-click **Win32OpenSSL\_Light-1\_1\_0h.exe**.
2. In the **User Account Control** window that opens, click **Yes**.
3. In the **Setup – OpenSSL Lite (32-bit)** wizard that opens, click **Next**.
4. On the **License Agreement** page, select **I accept** the agreement and then click **Next**.
5. On the **Select Destination Location** page, click **Next**.

6. On the **Select Start Menu Folder** page, click **Next**.
7. On the **Select Additional Tasks** page, click **Next**.
8. On the **Ready to Install** page, click **Install**.
9. On the **Completing the OpenSSL Lite (32-bit) Setup Wizard** page, de-select all options and then click **Finish**.
10. Right-click **Start**, then click **Run**.
11. In the **Run** window that opens type **MMC** in the **Open** box and then click **OK**.
12. In the **User Account Control** window that opens, click **Yes**.
13. In the **Console 1 – [Console Root]** window that opens click the **File** menu and then click **Add/Remove Snap-In**.
14. In the **Add/Remove Snap-Ins** window that opens, click **Certificates** and then click **Add**.
15. In the **Certificates snap-in** window that opens, ensure **My user account** is selected and then click **Finish**, then click **OK**.
16. Expand **Certificates – Current User\Personal** and then click **Certificates**.
17. Right-click the **StudentCert** and then click **All Tasks** and then click **Export**.
18. In the **Certificates Export Wizard** that opens, click **Next**.
19. On the **Export Private Key** page, select **Yes, export the private key** and then click **Next**.
20. On the **Export File Format** page, click **Next**.
21. On the **Security** page, select **Password** and then type **Pa55w.rd** in the **Password** and **Confirm password** boxes, and then click **Next**.
22. On the **File to Export** page  
type **C:\media\modulefiles\Module05\StudentCert.pfx** in the **File name** box and then click **Next**.
23. On the **Completing the Certificate Export Wizard** page, click **Finish**.
24. In the **Certificate Export Wizard** popup window that opens, click **OK**.

25. Close the **MMC Console** without saving any changes.
26. Right-click **Start** and then click **Command Prompt (Admin)**.
27. In the **User Account Control** window that opens, click **Yes**.
28. In the **Administrator: Command Prompt** window that opens type the following command and then press enter:

```
cd C:\OpenSSL-Win32\bin
```

```
openssl pkcs12 -in C:\media\modulefiles\Module05\StudentCert.pfx -nodes -out  
C:\media\modulefiles\Module05\StudentCert.pem
```

30. When the **Enter Import Password** prompt appears, type **Pa55w.rd** and then press enter.
31. Type **Exit** and then press enter to close the **Administrator: Command Prompt** window.
32. Confirm the **StudentCert.pem** file has been created in the **C:\media\modulefiles\Module05** folder.
33. Minimize the **AzS-Host** RDP window

### **Task 6: Assign a Role to the Service Principal**

1. On the **Azure Stack Lab** VM, in the **Azure Stack** operator portal that was left open in **Task 3** click **More Services**, then click **Subscriptions**.
2. In the **Subscriptions** page that opens, click **Default Provider Subscription**.
3. In the **Default Provider Subscription** page that opens, click **Access Control (IAM)**.
4. In the **Default Provider Subscription – Access Control (IAM)** page that opens, click **Add**.
5. In the **Add permissions** page that opens, click the drop-down list next to **Role** and then click **Owner**.
6. In the **Select** box type the first part of the **ApplicationName** that was recorded in **Task 4**. For example, type **Azurestack-StudentApp**

7. Select **AzureStack-StudentApp** from the results and then click **Save**.
8. Click the **Microsoft Azure Stack – Administration** link at the top of the page to return to the home page of the **Azure Stack** portal.
9. Leave **Internet Explorer** open.

### Task 7: Get the Tenant ID for Azure Stack

1. Open the RDP window and in the **AzS-Host on EdX-AzS-Host** window, in the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command and then press enter:

```
Set-Location C:\media\modulefiles\Module05
```

2. Type the following command and then press enter:

```
Import-Module .\GetTenant.psm1
```

3. Type the following command and then press enter:

```
Get-AzsDirectoryTenantIdentifier -authority https://adfs.local.azurestack.external/adfs
```

4. Make a note of the **Tenant ID** that is returned by copying it to the **Notepad** document left open in **Task 3**.
5. Leave the **Administrator: Windows PowerShell** window open.

### Task 8: Connect to Azure Stack using Azure CLI

1. In the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command and replace the **<VM Alias File URL>** place holder with the **URL** of the **VMAlias.json** file recorded earlier in the **Notepad** document.

```
az cloud register -n CloudAdmin --endpoint-resource-manager  
"https://adminmanagement.local.azurestack.external" --suffix-storage-endpoint  
"local.azurestack.external" --suffix-keyvault-dns ".adminvault.local.azurestack.external" --  
endpoint-active-directory-graph-resource-id "https://graph.windows.net/" --endpoint-  
vm-image-alias-doc <VM Alias File URL>
```

2. Press enter to run the command and wait for it to return to the **PowerShell** prompt.
3. Type the following command and then press enter:

```
az cloud set -n CloudAdmin
```

4. Type the following command and replace **<Tenant ID>** with the **Tenant ID** recorded in the Notepad document earlier:

```
az cloud update --profile 2017-03-09-profile --endpoint-resource-manager
"https://adminmanagement.local.azurestack.external" --endpoint-active-directory
"https://adfs.local.azurestack.external/adfs" --endpoint-active-directory-resource-id
"https://adminmanagement.adfs.azurestack.local/<Tenant ID>" --endpoint-active-
directory-graph-resource-id https://graph.windows.net/
```

5. Press enter to run the command.
6. Type the following command and replace **<Tenant ID>** and **<Client ID>** with the values stored in the **Notepad** document.

```
az login --tenant <Tenant ID> --service-principal -u <Client ID> -p
C:\media\modulefiles\Module05\StudentCert.pem
```

7. Press enter to run the command.
8. Output similar to the following should be returned:

```
[
{
"cloudName": "CloudAdmin1",
"id": "ee9d325c-f875-4bfd-930d-48aa686d2e00",
"isDefault": true,
"name": "Default Provider Subscription",
"state": "Enabled",
"tenantId": "9611d359-436f-4b9f-8c1c-37f4a0d3cd02",
```

```
"user": {  
  
  "name": "dbce9a89-2894-425b-bd71d03b6a59780e",  
  
  "type": "servicePrincipal"  
}  
  
}
```

9. This confirms you have signed into **Azure Stack** using **Azure CLI**.
10. Do not close the **Administrator: Windows PowerShell** window.

### **Task 9: Create a Resource Group using Azure CLI**

1. In the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command:

```
az group create -n Student_VMRG -l local
```

2. Wait until the command completes and the "**provisioningState**": "**Succeeded**" message appears.
3. Minimize the RDP window and in the **Azure Stack** operator portal, click **Resource Groups**.
4. In the **Resource groups** page that opens confirm the **Student\_VMRG** Resource Group is listed.
5. Click the **Microsoft Azure Stack – Administration** link to return to the home page of the **Azure Stack** portal.

### **Task 10: Update the VM deployment parameters file**

1. Open the RDP window and in the **AzS-Host on EdX-AzS-Host** window, navigate to **C:\media\modulefiles\Module05** and open the **DeployVM.parameters.json** file with **Notepad**.
2. Replace **VM\_NAME** with **StudentVM**.
3. Save and close the **DeployVM.parameters.json** file.

## Task 11: Debug an ARM template using Azure CLI

1. In the **AzS-Host on EdX-AzS-Host** window, in the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command to deploy a virtual machine to the **Student\_VMRG** resource group.

```
az group deployment create --resource-group "Student_VMRG" --template-file  
C:\media\modulefiles\Module05\DeployVM.json --parameters  
C:\media\modulefiles\DeployVM.parameters.json
```

2. Press enter to run the command.
3. Notice the command fails with the **Multiple error occurred: BadRequest. Please see details message.**
4. Type the following command to deploy a virtual machine to the **StudentX\_VMRG** resource group. Replace **X** where it appears in the command with your student number. Note the **--debug** parameter has been added this time.

```
az group deployment create --resource-group "Student_VMRG" --template-file  
C:\media\modulefiles\Module05\DeployVM.json --parameters  
C:\media\modulefiles\Module05\DeployVM.parameters.json --debug
```

5. Press enter to run the command.
6. Review the debug information returned and note the message at the end stating **The template parameters `azureRegion` in the parameters file are not valid; they are not present in the original template and can therefore not be provided at deployment time.**
7. Navigate to **C:\media\modulefiles\Module05** and open the **DeployVM.parameters.json** file with **Notepad**.
8. Remove the **azureRegion** parameter by changing the contents of the file from this:

```
{  
  
"$schema": "http://schema.management.azure.com/schemas/2015-01-01/  
01/deploymentParameters.json#",  
  
"contentVersion": "1.0.0.0",
```

```
"parameters": {  
  "azureRegion": {  
    "value": "local"  
  },  
  "VMNAME": {  
    "value": "Student1VM"  
  },  
  "ADMINUSERNAME": {  
    "value": "DemoAdmin"  
  },  
  "ADMINPASSWORD": {  
    "value": "Pa55w.rd12345"  
  }  
}
```

9. To this.

```
{  
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01/  
  deploymentParameters.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "VMNAME": {  
      "value": "StudentVM"  
    }  
  }  
}
```

```
},  
"ADMINUSERNAME": {  
  "value": "DemoAdmin"  
},  
"ADMINPASSWORD": {  
  "value": "Pa55w.rd12345"  
}  
}  
}
```

10. Save and then close the **DeployVM.parameters.json** file.

### **Task 12: Deploy the virtual machine using Azure CLI and confirm it has been created**

1. In the **Administrator: Windows PowerShell** window that was left open in the previous task, type the following command to deploy a virtual machine to the **Student\_VMRG** resource group.

```
az group deployment create --resource-group "Student_VMRG" --template-file  
C:\media\modulefiles\Module05\DeployVM.json --parameters  
C:\media\modulefiles\Module05\DeployVM.parameters.json
```

2. Press enter to run the command.
3. Wait until the **Running ...** notification appears.
4. Minimize the RDP window. In the **Azure Stack** portal on the **Azure Stack Lab** VM click **Virtual Machines**.
5. After approximately 2 minutes the **StudentVM** will appear with a **Status** of **Creating**.
6. Also notice the **Resource Group** that the virtual machine is being created in.

7. Wait approximately 15 minutes until the **Status** changes from **Creating** to **Running**.
8. In the **Administrator: Windows PowerShell** window confirm the "**provisioningState**": "**Succeeded**" message is displayed. This confirms the deployment was successful.
9. Type the following command:

```
az group delete --name Student_VMRG
```

10. When the **Are you sure you want to perform this operation? (y/n)** prompt appears, type **Y** then press enter.
11. After approximately 10 minutes the **Resource Group** and associated virtual machine are removed from **Azure Stack**.
12. Close the **Administrator: Windows PowerShell** and **Azure Stack** portal windows.

**Results:** After this exercise, you should have connected to Azure Stack using Azure CLI and then used an ARM template to deploy a new virtual machine in Azure Stack.

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.