

Module 6: Infrastructure as a Service and Microsoft Azure Stack

Lab: Provisioning Storage and Virtual Machines in Microsoft Azure Stack

Exercise 1: Creating a Subscription and Resource Groups

Task 1: Create the AD DS User Account for AsUser

1. On the **Azure Stack Lab** VM, 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.
2. In the **User Account Control** dialog, click **Yes**
3. Click **Start**, 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. Right-click **Users**, then click **New**, then click **User**.
7. 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: **ASUser**
 - User logon name: **ASUser**
 - Password: **Pa55w.rd**
 - User must change password at next logon: **Deselect**
 - User cannot change password: **Select**
 - Password never expires: **Select**
8. Complete the wizard and then close **Active Directory Users and Computers**.

Task 2: Create a Subscription

1. On the **Azure Stack Lab** VM, 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. Click **Plans** then click **Add**.
5. In the **New Plan** blade that opens complete the form using the following settings, all other settings should remain default:
 1. Display Name: **Virtual-Machines**
 2. Resource Group: **Create new**
 3. Resource Group Name: **RG-OffersAndPlans**
 4. Services: **Microsoft.Compute**, **Microsoft.Network** and **Microsoft.Storage**
 5. Quotas: Accept default **Quotas**.
6. Click **Create** to create the **Plan**.
7. In the **Azure Stack Operator Portal**, from the navigation pane click **Offers**.
8. Click **Add**.
9. 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 plan: **Virtual Machines**

10. Click **Create** to create the **Offer**.
11. 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.
12. Click the **virtual-machines** offer, click **Change state**, then click **Public**
13. Close Internet Explorer.
12. Open Internet Explorer and browse to <https://portal.local.azurestack.external>
13. Sign into **Azure Stack** with the following credentials:
 - Username: **AzureStack\ASUser**
 - Password: **Pa55w.rd**
14. In the **Microsoft Azure Stack** page that opens, click **More Services**.
15. Click **Subscriptions**.
16. Click **Add**.
17. In the **Get a subscription** blade, in the **Display name** box, type **ASUser Subscription**.
18. Click **Select an offer**.
19. In the **Choose an offer** blade, click **Virtual Machines**.
20. In the **Get a subscription** blade, click **Create**, then click **Refresh**.
21. Click **More Services**, then click **Subscriptions** again.
22. In the **Subscriptions** blade, you will see the new **ASUser Subscription** listed.
23. Click the **Microsoft Azure Stack** title to return to the home page of the **Azure Stack** Portal.
24. Do not close **Internet Explorer**.

Task 3: Create Resource Groups

1. In the **Azure Stack User Portal** that is logged in using the **ASUser** account, click **More Services**.

2. Under the **General** section, click **Resource groups**.
3. In the **Resource groups** blade, click **Add**.
4. In the **Resource group** blade, in the **Resource group name**, type **RG-local-VMStorage**.
5. Click **Create**.
6. In the **Resource groups** blade, click **Refresh**. You will see the new **RG-local-VMStorage** resource group listed.
7. In the **Resource groups** blade, click **Add**.
8. In the **Resource group** blade, in the **Resource group name**, type **RG-local-VMs**.
9. Click **Create**.
10. In the **Resource groups** blade, click **Refresh**. You will see the new **RG-local-VMs** resource group listed.
11. In the **Resource groups** blade, click **Add**.
12. In the **Resource group** blade, in the **Resource group name**, type **RG-local-VNMain**.
13. Click **Create**.
14. In the **Resource groups** blade, click **Refresh**. You will see the new **RG-local-VNMain** resource group listed.
15. Click the **Microsoft Azure Stack** title to return to the home page of the **Azure Stack** Portal.
16. Do not close Internet Explorer.

Results: After completing this exercise, you will be able to: Create an AD DS User Account. Create a subscription to an offer. Create resource groups in the Azure Stack User Portal.

Exercise 2: Creating Storage Accounts

Task 1: Create a Storage Account for Virtual Machine Diagnostics

1. In the **Azure Stack User Portal** that is logged in as **ASUser**, click **More Services** and then click **Storage Accounts**.
2. In the **Storage accounts** blade, click **Add**.
3. In the **Create storage account** blade, in the **Name** box, type **asusersavmdiags**.
4. Under **Resource group** select **Use existing**.
5. Select **RG-local-VMStorage** from the drop-down box under **Use existing**.
6. Click **Create**.
7. In the **Storage accounts** blade, click **Refresh**. The **asusersavmdiags** Storage Account will be displayed.
8. Click the **Microsoft Azure Stack** title to return to the home page of the **Azure Stack** Portal.
9. Do not close **Internet Explorer**.

Task 2: Create a Storage Account for Virtual Machines

1. On the **Azure Stack Lab** VM, click **Start**.
2. Navigate to **Windows PowerShell**.
3. Right-click **Windows PowerShell**.
4. Navigate to **More | Run as administrator**.
5. In the **User Account Control** dialog, click **Yes**.
6. In the **Administrator: Windows PowerShell** 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. Press Enter at the end of each PowerShell command:

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

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

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

8. Enter the following PowerShell command and press Enter to create a PowerShell credential object to login to Azure Stack with:

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

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

```
Login-AzureRmAccount -EnvironmentName "AzureStack" -TenantId $TenantID -  
Credential $Cred
```

10. Enter the following PowerShell command and press Enter:

```
$params = @{savmstorageType='Standard_LRS'; saname='uservm'}
```

11. Enter the following PowerShell command and press Enter

```
New-AzureRmResourceGroupDeployment -ResourceGroupName RG-local-VMStorage -  
Name asusersavmstorage -TemplateFile  
'C:\Media\ModuleFiles\Module06\storageAccount.json' -TemplateParameterObject  
$params -Verbose
```

12. When the command finishes, close the **Administrator: Windows PowerShell** console.
13. Go back to the **Azure Stack User Portal** (that should still be open). In the **Storage accounts** blade, click **Refresh**. The **asusersavmsdiags** and **uservm<random_id>** Storage Accounts will be shown.
14. Click the **Microsoft Azure Stack** title to return to the home page of the **Azure Stack** Portal.
15. Do not close **Internet Explorer**.

Results: After completing this exercise, you will be able to: Create a storage account by using the Azure Portal. Create a storage account by using Azure PowerShell and an ARM template.

Exercise 3: Creating a Virtual Network

Task 1: Create a Virtual Network

1. In the **Azure Stack User Portal** that is logged in as **ASUser**, click **More Services**.
2. Under **Networking**, click **Virtual Networks**.
3. In the **Virtual networks** blade, click **Add**.
4. In the **Create virtual network** blade, in the **Name** box type **VN-local-Main**.
5. In the **Address space** box change the value to **10.0.0.0/16**.
6. In the **Subnet address range** box change the value to **10.0.254.0/24**.
7. Under **Resource group** select **Use existing**.
8. Select **RG-local-VNMain** from the drop-down box under **Use existing**.
9. Click **Create**.
10. Wait approximately 30 seconds.
11. In the **Virtual networks** blade, click **Refresh**. The virtual network **VN-local-Main** will appear.
12. Do not close **Internet Explorer**.

Task 2: Create Virtual Network Subnets

1. Click **Start**.
2. Navigate to **Windows PowerShell**.
3. Right-click **Windows PowerShell**.
4. Navigate to **More | Run as administrator**.
5. In the **User Account Control** dialog, click **Yes**.
6. In the **Administrator: Windows PowerShell** 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. Press Enter at the end of each PowerShell command:

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

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

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

8. Enter the following PowerShell command and press Enter to create a PowerShell credential object to login to Azure Stack with:

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

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

```
Login-AzureRmAccount -EnvironmentName "AzureStack" -TenantId $TenantID -  
Credential $Cred
```

10. Enter the following PowerShell command and press Enter after each command:

```
$params = @{existingVNETName='VN-local-Main'; newSubnetName = 'Autosubnet0';  
newSubnetAddressPrefix = '10.0.0.0/24'}
```

```
New-AzureRmResourceGroupDeployment -Name AddSubnets -ResourceGroupName  
RG-local-VNMain -TemplateFile 'C:\Media\ModuleFiles\Module06\addSubnets.json' -  
TemplateParameterObject $params -Verbose
```

```
$params = @{existingVNETName='VN-local-Main'; newSubnetName = 'Autosubnet1';  
newSubnetAddressPrefix = '10.0.1.0/24'}
```

```
New-AzureRmResourceGroupDeployment -Name AddSubnets -ResourceGroupName  
RG-local-VNMain -TemplateFile 'C:\Media\ModuleFiles\Module06\addSubnets.json' -  
TemplateParameterObject $params -Verbose
```

```
$params = @{existingVNETName='VN-local-Main'; newSubnetName = 'Autosubnet2';  
newSubnetAddressPrefix = '10.0.2.0/24'}
```



```
New-AzureRmResourceGroupDeployment -Name AddSubnets -ResourceGroupName  
RG-local-VNMain -TemplateFile 'C:\Media\ModuleFiles\Module06\addSubnets.json' -  
TemplateParameterObject $params -Verbose
```

11. Go back in to the Internet Explorer window (that should still be open). In the **Virtual networks** blade, click **VN-local-Main**.
12. In the **VN-local-Main** blade, click **Subnets**.
13. You will see four subnets available:

| Name | Address Range |
|-------------|---------------|
| default | 10.0.254.0/24 |
| AutoSubnet0 | 10.0.0.0/24 |
| AutoSubnet1 | 10.0.1.0/24 |
| AutoSubnet2 | 10.0.2.0/24 |

14. Click the **Microsoft Azure Stack** title to return to the home page of the **Azure Stack** Portal.
15. Close **Internet Explorer**.

Results: After completing this exercise, you will be able to: Create a virtual network by using the Azure Portal. Create additional subnets and an NSG by using Azure PowerShell and an ARM template.

Exercise 4: Deploying System Center 2016 Operations Manager using ARM templates

Due to the virtual hosted solution for this lab this exercise has been removed

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 7: Platform as a Service and Microsoft Azure Stack

Lab: Configuring Platform as a Service in Azure Stack

Exercise 1: Adding hosting servers

Task 1: Add a hosting server to the SQL resource provider

1. On **Azure Stack Lab** VM, 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. Click **SQL Hosting Servers**.
6. In the **SQL Hosting Servers** blade that opens, click **Add**.
7. In the **Add a SQL Hosting Server** blade that opens configure the following. All other settings should remain as default:
 - SQL Server Name: **sql4appsvc.local.cloudapp.azurestack.external**

- Username: **sa**
 - Password: **Pa55w.rd**
 - Size of Hosting Server in GB: **50**
 - Resource group: Use existing **appservicefileshare**
 - SKU: Create a new SKU with the following settings and then click **OK**:
 - Name: **DefaultSQL**
 - Family: **SQLServer2014**
 - Tier: **Standalone**
 - Edition: **Evaluation**
8. Click **Create**.
 9. Wait for the operation to complete and then confirm **sql4appsvc.local.cloudapp.azurestack.external** has been added to the **SQL Hosting Servers** blade.
 10. Click the **Microsoft Azure Stack – Administration** title to return to the home page.
 11. Leave **Internet Explorer** open.

Exercise 2: Configuring a Plan and Offer

Task 1: Create a Plan for SQL Server databases

1. In the **Microsoft Azure Stack – Administration** page, click **Plans**
2. In the **Plans** blade that opens, click **Add**.
3. In the **New plan** blade that opens, configure the **Plan** with the following settings, all other settings should remain as default:
 - Display name: **SQL-databases**
 - Resource group: Use existing **appservicefileshare**

- Services: **Microsoft.SQLAdapater**
 - Quotas: Create a new **Quota** with the following settings, all other settings should remain as default:
 - Quota name: **DefaultQuota**
4. Click **Create** to create the plan.
 5. In the **Plans** blade, click **Refresh** and confirm the **SQL-databases** Plan is displayed.
 6. Click the **Microsoft Azure Stack – Administration** title to return to the home page.
 7. Leave **Internet Explorer** open.

Task 2: Create an Offer for SQL Server databases

1. In the **Microsoft Azure Stack - Administration** page, click **Offers**
2. In the **Offers** blade that opens, click **Add**.
3. In the **New Offer** blade that opens, configure the following settings, all other settings should remain as default:
 - Display name: **SQL-databases**
 - Resource group: Use existing **appservicefileshare**
 - Base plan: **SQL-databases**
4. Click **Create** to create the **Offer**.
5. In the **Offer** blade, click **Refresh** and confirm the **SQL-databases** Offer is displayed.
6. Click the **SQL-databases** Offer.
7. In the **SQL-databases** blade that opens, click **Change State**, then click **Public**.
8. Close **Internet Explorer**.

Results: After this exercise, you should have created a Plan and an Offer for SQL databases. You should have also made the Offer public, in order for tenants to subscribe to it.

Exercise 3: Create SQL Server databases

Task 1: Subscribe to the SQL-databases Offer

1. On the **Azure Stack Lab** VM, open **Internet Explorer**.
2. Browse to <https://portal.local.azurestack.external>
3. Sign into **Azure Stack** with the following credentials:
 - Username: **AzureStack\ASUser**
 - Password: **Pa55w.rd**
4. In the **Microsoft Azure Stack**, page that opens, click **More Services** and then click **Subscriptions**.
5. Click **Add**.
6. In the **Get a subscription** blade that opens, type **SQL-databases** in the **Display name** box and then click **Select an offer**.
7. In the **Choose an offer** blade that opens, click **SQL-databases**.
8. Click **Create** and then click **Refresh**.
9. Click **More Services** and then click **Subscriptions**.
10. Confirm the **SQL-databases** subscription is listed in the **Subscriptions** blade that opens.
11. Click the **Microsoft Azure Stack** title to return to the home page.
12. Leave **Internet Explorer** open.

Task 2: Provision a new database

1. In the **Azure Stack Portal** that is logged in using the **ASUser** account, click **New**.

2. Click **Data + Storage** and then click **SQL Database**.
3. In the **Create Database** blade that opens, configure the following settings, all other settings should remain as default (Note: It may take up to 30 minutes for the SKU to become available):
 - Database name: **Contoso-Tenant**
 - Max size in MB: **64**
 - Subscription: **SQL-databases**
 - Resource Group: Create new: **SQL-databases**
 - SKU: **DefaultSQL**
 - Login: Create a new **Login** with the following settings:
 - Database login: **ASUser**
 - Password: **Pa55w.rd**
 - Confirm password: **Pa55w.rd**
4. Click **Create** to provision the new database.
5. Wait until the database has been provisioned and then click **More Services**, then click **SQL Databases**.
6. In the **SQL Databases** blade that opens, confirm the **Contoso-Tenant** database is listed.
7. close Internet Explorer.

Results: After this exercise, you should have used the tenant portal to subscribe to the SQL-databases offer and created a new database. This confirms that the SQL Server resource provider is functioning as expected. You should have also confirmed that the SQL database was provisioned on the hosting server.

Exercise 4: Configure a Plan and Offer for App Service

Task 1: Create an App Service Plan

1. On the **Azure Stack Lab** VM, 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. In the **Plans** blade that opens, click **Add**.
6. In the **New Plan** blade that opens, configure the following settings, all other settings should remain as default:
 - Display name: **App-Service-Websites**
 - Resource group: Create a new Resource Group named **AppService**
 - Services: **Microsoft.Web**
 - Quotas: **Default**
7. Click **Create**.
8. Confirm the **App-Service-Websites** Plan is displayed in the **Plans** blade. You may need to click the **Refresh** button.
9. Click the **Microsoft Azure Stack – Administration** title to return to the home page.
10. Leave **Internet Explorer** open.

Task 2: Create an App Service Offer

1. In the **Microsoft Azure Stack – Administration** portal that is opened using the **CloudAdmin** account, click **Offers**.
2. In the **Offers** blade that opens, click **Add**.
3. In the **New Offer** blade that opens configure the following settings, all other settings should remain as default:
 - Display name: **App-Service-Websites**

- Resource group: Use existing **AppService** Resource Group.
 - Base plan: **App-Service-Websites**
4. Click **Create** to create the **Offer** and then click **Refresh** intermittently until the **app-service-websites** Offer is displayed in the **Offers** blade.
 5. Click **app-service-websites**
 6. In the **app-service-websites** blade that opens, click **Change state** and then click **Public**.
 7. Click the **Microsoft Azure Stack – Administration** title to return to the home page.
 8. Close **Internet Explorer**.

Task 3: Subscribe to the App Service Offer

1. Open **Internet Explorer**.
2. Browse to <https://portal.local.azurestack.external>
3. Sign into the **Azure Stack** user portal with the following credentials:
 - Username: **AzureStack\ASUser**
 - Password: **Pa55w.rd**
4. In the **Microsoft Azure** page that opens, click **More Services** and then click **Subscriptions**.
5. In the **Subscriptions** blade that opens, click **Add**.
6. In the **Get a subscription** blade that opens, type **Websites** in the **Display name** box and then click **Offer**.
7. In the **Choose an offer** blade that opens, click **App-Service-Websites**.
8. Click **Create** and then click **Refresh**.
9. Click **More Services** and then click **Subscriptions**.

10. Confirm the **WebSites** Subscription is listed and then click the **Microsoft Azure Stack** title to return to the home page. Notice the **App Services** navigation tab has appeared on the home page.
11. Leave **Internet Explorer** open.

Task 4: Create a new App Service website

1. In the **Azure Stack** user portal that is signed in using the **ASUser** account, click **New**.
2. In the **New** blade that opens, click **Web + Mobile**.
3. Click **Web App**.
4. In the **Web App** blade that opens configure the following settings, all other settings should remain as default:
 - App name: **Fabrikam-Intranet**
 - Subscription: **WebSites**
 - Resource group: Create a new **Resource Group** named **Fabrikam-Intranet**
 - App Service plan/location: Create a new **App Service Plan** with the following settings:
 - App Service plan: **Default-plan**
 - Location: **local**
 - Pricing tier: **F1 Free**
5. Click **OK** on the **New App Service Plan** blade.
6. Click **Create** on the **Web App** blade.
7. The page automatically returns to the home page.
8. Refresh the page intermittently until the **Fabrikam-Intranet** icon appears in the **All resources** section of the **Azure Stack** dashboard. You may need to click **See more** to display all resources.

9. Open a new **Internet Explorer** tab and then browse to <http://fabrikam-intranet.appservice.local.azurestack.external>
10. Confirm the page displays the message stating **This web app has been successfully created.**
11. Close the **Internet Explorer** page that displays the **message This web app has been successfully created.**
12. Sign-out and close **Internet Explorer**.

Task 5: Configure application access

1. Open **Internet Explorer**.
2. Browse to <https://portal.local.azurestack.external>
3. Sign into the **Azure Stack** user portal with the following credentials:
 - Username: **AzureStack\ASUser**
 - Password: **Pa55w.rd**
4. In the **Microsoft Azure** page that opens, click **App Services**
5. Click **Fabrikam-Intranet**.
6. In the **Fabrikam-Intranet** blade that opens, click **Deployment credentials**.
7. Configure the credentials as follows:
 - FTP/deployment username: **FabrikamAdmin**
 - Password: **Pa55w.rd**
 - Confirm password: **Pa55w.rd**
8. Click **Save**.
9. Click the **Microsoft Azure Stack** title to return to the home page and then close **Internet Explorer**.

Task 6: Change the home page of the Fabrikam Intranet website

1. Open **File Explorer**.
2. Browse to **ftp://ftp.appservice.local.azurestack.external/**
3. In the **Log On As** dialog box that opens enter the following credentials:
 - User name: **Fabrikam-IntranetX\FabrikamAdmin**
 - Password: **Pa55w.rd**
4. Click **Log On**.
5. Expand **site\wwwroot** and note the **hostingstart.html** file that is present. This is the current home page of the **Fabrikam Intranet** website.
6. On **AZS-RDP01** navigate to **C:\Media\ModuleFiles\Module07** and copy the **hostingstart.html** file.
7. Paste the **hostingstart.html** file into the **wwwroot** folder replacing the existing file.
8. Close **File Explorer**.
9. Open **Internet Explorer** and browse to <http://fabrikam-intranet.appservice.local.azurestack.external>
10. Notice the home page has changed to the **Fabrikam Intranet**. You may need to refresh the **Internet Explorer** page to see the changes.
11. Close Internet Explorer.

Results: After this exercise, you should have created a Plan and Offer that includes the services offered in App Service. You should have then subscribed to the offer as a tenant and provisioned a web site. Finally, you should have configured access to the FTP site associated with the website and then changed the default home page.

©2016 Microsoft Corporation. All rights reserved.

Module 8: Monitoring in Microsoft Azure Stack

Lab: Troubleshooting and Monitoring Microsoft Azure Stack and Guest Operating Systems

Exercise 1: Monitoring Azure Stack

Due to the virtual hosted solution for this lab this exercise has been removed

Exercise 2: Monitoring Guest Operating Systems

Due to the virtual hosted solution for this lab this exercise has been removed

Exercise 3: Troubleshooting Azure Stack by using the Audit Logs and Alerts

Task 1: Reviewing Azure Stack Alerts in Operations Manager

Continue to the next task

Task 2: Reviewing Azure Stack Alerts

1. On **Azure Stack Lab** VM, double-click the **Enter-PSSession** icon on the desktop.
2. Type the following command, and then press Enter:

```
stop-vm azs-ca01
```
3. Browse to <https://adminportal.local.azurestack.external>
4. Sign into **Azure Stack** with the following credentials:
 - Username: **AzureStack\CloudAdmin**
 - Password: **Pa55w.rd**
5. In the **Microsoft Azure Stack – Administration** Portal, click **More Services** and then under **Administration**, click **Region Management**.
6. In the Alerts section, click **Warning**.
7. In the **Alerts** blade that opens click the **Infrastructure role instance unavailable** alert.
8. In the **Alert Details** blade that opens review the following properties:
 - Name
 - Severity

- State
 - Component
 - Description
 - Remediation
9. Close the **Infrastructure role instance unavailable** blade.
 10. In the open PowerShell Window type the following command, and the press Enter:

```
start-vm azs-ca01
```
 11. When the command has completed, close the **PowerShell** window.
 12. Wait for 5 minutes.
 13. Press **F5** in the **Azure Stack** Portal on **AzS-RDP01** to refresh the page.
 14. Notice the **Warning** alert has now disappeared as the problem no longer exists.
 15. Click the **Microsoft Azure Stack – Administration** title to return to the home page.
 16. Leave **Internet Explorer** open.

Task 3: Reviewing Azure Stack Audit Logs

1. In the **Azure Stack– Administration** portal, click **Resource Groups**.
2. In the **Resource Groups** blade that opens, click **Add**.
3. In the **Resource group** blade, type **Test** in the Resource group name box and then click **Create**.
4. Close the **Resource groups** blade.
5. Click **More Services** and then scroll down and click **Audit Logs**.
6. Click **Filter** and then in the **Filter** blade that opens, click the drop-down list next to **Time Span** and then click **past 1 hour**, then click **Done**.
7. Click the log entry that displays an **Operation of Update resource group**.

8. In the **Update resource group** blade that opens, click the **Update resource group** log entry that displays a **Status** of **Succeeded**.
9. In the **Detail** blade that opens, review the information provided including:
 - Caller
 - Resource Group
 - Category
 - Level
10. Close the **Detail** blade.
11. Close the **Update resource group** blade.
12. Close the **Audit Logs** blade.
13. Click **Resource Groups** and then in the **Resource group blade** that opens click the **Test** Resource Group.
14. In the **Test** blade that opens, click **Delete resource group**.
15. In the **Are you sure you want to delete Test?** blade that opens, type **Test** in the **Type the Resource Group Name** box and then click **Delete**.
16. Wait until the **Test** Resource Group has been deleted.
17. Close the **Resource groups** blade.
18. Click **More Services** and then scroll down and click **Audit Logs**.
19. Click the log entry that displays an **Operation** of **Delete resource group**. You may need to refresh the page until it appears.
20. In the **Delete resource group** blade that opens, click the **Delete resource group** log entry that displays a **Status** of **Succeeded**.
21. In the **Detail** blade that opens, review the information provided including:
 - Caller
 - Resource Group

- Category
- Level

22. Close the **Detail** blade.

23. Close the Delete resource group blade.

24. Close the **Audit Logs** blade.

25. Close the **Azure Stack** Portal.

Results: After this exercise, you should have used the **Active Alerts** section in **Region Management** to review the alert information that is provided when an Azure Stack infrastructure component is offline. You should have also used the Audit Logs in the Azure Stack Portal to review log entries that are created when a resource group is create or deleted.

©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 9: Licensing Microsoft Azure Stack and Billing Tenants

Lab: Obtaining Usage Information

Exercise 1: Obtaining Azure Stack usage

Task 1: Extract Tenant Usage from Azure Stack using PowerShell

1. On the **Azure Stack Lab** VM, 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 AzureStack -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 AzureStack
```

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

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

5. Enter the following PowerShell command and press Enter after each command:

```
Set-Location C:\Media\ModuleFiles\Module09
```

```
.\Get-TenantUsage.ps1
```

6. Provide a start date of yesterday and an end date of today.
7. Review the **Hourly** usage information that is returned. Note that to determine the resource used you must reference the **MeterID** returned with the table provided in the **Tenant Based Usage** topic in this module.
8. Close the **Windows PowerShell** window.

Task 2: Create a Service Principal to authenticate with Azure Stack to retrieve Provider usage using the REST API

1. On the **Azure Stack Lab** VM, click **Start**.
2. Right-click **Windows PowerShell**.
3. Click **Run as administrator**.
4. In the **User Account Control** dialog, click **Yes**.
5. In the **Administrator: Windows PowerShell** enter the following PowerShell command and press Enter to import the required PowerShell Module:

```
Import-Module AzureRM.Profile
```

6. Enter the following command to add the IP address of the privileged endpoint (AzS-ERCS01) to the trusted hosts on the **Azure Stack Lab** VM:

```
winrm set winrm/config/client '@{TrustedHosts="192.168.200.225"}'
```

7. Enter the following command to store the **CloudAdmin** credentials:

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

8. In the **Windows PowerShell credential** request window that opens, type **AzureStack\CloudAdmin** in the User name box.

9. Type **Pa55w.rd** in the **Password** box and then click **OK**.

10. Type the following commands pressing enter after each.

```
$session = New-PSSession -ComputerName 192.168.200.225 -ConfigurationName PrivilegedEndpoint -Credential $creds
```

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

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

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

11. Make a note of the **ClientID** that is returned.

12. Open **Internet Explorer** and sign into the **Operator Portal** using your **AzureStack\CloudAdmin** credentials.
13. In the **Azure Stack** operator portal that opens, click **More Services** and then click **Subscriptions**.
14. In the Subscriptions page that opens click **Default Provider Subscription**.
15. In the **Default Provider Subscription** page that opens, make a note of the **Subscription ID** that is provided on the **Overview** page and then click **Access Control (IAM)**.
16. In the **Default Provider Subscription – Access Control (IAM)** page that opens, click **Add**.
17. In the **Add permissions** page that opens, click the drop-down list next to **Role** and then click **Owner**.
18. In the **Select** box, type **AzureStack-CloudAdmin**.
19. Select the **AzureStack-CloudAdmin** account and then click **Save**.
20. Wait until the **Added user** notification appears, then click **More Services** and then click **User subscriptions**.
21. In the User subscriptions page that opens, click **ASUserSubscription**
22. In the **user subscription** page that opens, make a note of the **User Subscription ID** that is shown on the **Overview** page.
23. Close the **Azure Stack** operator portal.
24. Leave the **Administrator: Windows PowerShell** window open.

Task 3: Obtain an Access Token from Azure Stack to retrieve Provider usage using the REST API

1. In the **Administrator: Windows PowerShell** window that was left open from the previous task, enter the following commands pressing enter after each command:

```
$TestResource = Invoke-  
RestMethod https://adminmanagement.local.azurestack.external/metadata/endpoints?api-version=1.0
```

```
$Resource = $TestResource.authentication.audiences
```

```
$AuthorityEndpoint = "https://adfs.local.azurestack.external/adfs"
```

```
$ServicePrincipalThumbprint = $Cert.Thumbprint
```

2. Type the following command replacing the **GUID** with the **GUID** returned in the **ClientID** noted in **Step 11** from the previous task:

```
$ClientId = "42ade138-026b-4fbc-903a-33d792dd9547"
```

3. Type the following commands, pressing enter after each command:

```
$authContext = New-Object
```

```
Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext($AuthorityEndpoint, $false)
```

```
$store = New-Object
```

```
System.Security.Cryptography.X509Certificates.X509Store([System.Security.Cryptography.X509Certificates.StoreName]::My,[System.Security.Cryptography.X509Certificates.StoreLocation]::CurrentUser)
```

```
$store.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadOnly)
```

```
$certCollection = $store.Certificates.Find([System.Security.Cryptography.X509Certificates.X509FindType]::FindByThumbprint, $ServicePrincipalThumbprint, $false)
```

```
$store.Close()
```

```
$cacert=New-Object
```

```
Microsoft.IdentityModel.Clients.ActiveDirectory.ClientAssertionCertificate($ClientId, $certCollection[0])
```

```
$Token = ($authContext.AcquireToken($Resource, $cacert)).AccessToken
```

4. Leave the **Administrator: Windows PowerShell** window open.

Task 4: Formulate the Request URI and retrieve Provider Usage using the REST API

1. In the **Administrator: Windows PowerShell** window that was left open from the previous task, enter the following command replacing the first **GUID** with

the **Default Provider Subscription Subscription ID** from **Step 15** in **Task 3** and the second **GUID** with the **ASUserSubscription User Subscription ID** obtained in **Step 22** of **Task 3**. Also modify the **reportedStartTime** with the date of yesterday and a **reportedEndTime** date

```
$SubscriptionURI =  
'https://adminmanagement.local.azurestack.external/subscriptions/2c2c1dad-3f44-  
4337-b516-  
0ec212943c66/providers/Microsoft.Commerce/subscriberUsageAggregates?reportedSta  
rtTime=2018-01-01&reportedEndTime=2018-02-  
20&aggregationGranularity=Daily&subscriberId=99d181b0-4ff9-4268-b30b-  
ba5968aed045&api-version=2015-06-01-preview'
```

2. Enter the following commands pressing enter after each command:

```
$Headers = @{'authorization'="Bearer $($Token)"}
```

```
$Request = Invoke-RestMethod -Method GET -ContentType 'application/json' -Headers  
$Headers -Uri $SubscriptionURI
```

```
$Request.value.properties | Format-List
```

3. Review the usage information returned. This information is taken from the **ASUserSubscription** which is part of the **Default Provider Subscription**. You can correlate the **meterID** values with the **Meters** shown in the **Tenant-Based Usage** topic to determine the actual resources used.
4. Type the following command and then press enter:

```
$Request.value.properties | Measure
```

5. Note the **Count** value returned. This is the number of resource usage records recorded.
6. Use the up-arrow to return to the command entered in Step 1, then remove the 2nd GUID from the command such as **&subscriberId=99d181b0-4ff9-4268-b30b-ba5968aed045**
7. Enter the command and then enter the following commands again:

```
$Request = Invoke-RestMethod -Method GET -ContentType 'application/json' -Headers  
$Headers -Uri $SubscriptionURI
```

\$Request.value.properties | Format-List

8. Review the usage information returned. This information now shows all subscriptions under the **Default Provider Subscription**.
9. Run the following command again:

\$Request.value.properties | Measure

10. Note the **Count** value returned is much higher as usage for all subscriptions is returned.
11. Leave the **Administrator: Windows PowerShell** window open.

Task 5: Update Azure Stack permissions to retrieve Tenant usage using the REST API

1. From the desktop, open the **User Portal** and sign in using your **AzureStack\ASUser** credentials.
2. In the **Azure Stack** user portal that opens, click **More Services** and then click **Subscriptions**.
3. In the **Subscriptions** page that opens click **App-Service-Websites**.
4. In the **App-Service-Websites** page that opens, make a note of the **Subscription ID** that is provided on the **Overview** page and then click **Access Control (IAM)**.
5. In the **App-Service-Websites – Access Control (IAM)** page that opens, click **Add**.
6. In the **Add permissions** page that opens, click the drop-down list next to **Role** and then click **Owner**.
7. In the **Select** box, type **AzureStack-CloudAdmin**.
8. Select the **AzureStack-CloudAdmin** account and then click **Save**.
9. Wait until the **Added user notification** appears, then close the **Azure Stack** user portal.

Task 6: Obtain an Access Token from Azure Stack to retrieve Tenant usage using the REST API

1. In the **Administrator: Windows PowerShell** window that was left open from the previous task, enter the following commands pressing enter after each command:

```
$TestResource = Invoke-  
RestMethod https://management.local.azurestack.external/metadata/endpoints?api-  
version=1.0
```

```
$Resource = $TestResource.authentication.audiences
```

```
$AuthorityEndpoint = "https://adfs.local.azurestack.external/adfs"
```

```
$ServicePrincipalThumbprint = $Cert.Thumbprint
```

2. Type the following commands, pressing enter after each command:

```
$authContext = New-Object  
Microsoft.IdentityModel.Clients.ActiveDirectory.AuthenticationContext($AuthorityEndpoi  
nt, $false)
```

```
$store = New-Object  
System.Security.Cryptography.X509Certificates.X509Store([System.Security.Cryptography  
.X509Certificates.StoreName]::My,[System.Security.Cryptography.X509Certificates.StoreL  
ocation]::CurrentUser)
```

```
$store.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadOnly)
```

```
$certCollection = $store.Certificates.Find(  
[System.Security.Cryptography.X509Certificates.X509FindType]::FindByThumbprint,  
$ServicePrincipalThumbprint, $false)
```

```
$store.Close()
```

```
$cacert=New-Object  
Microsoft.IdentityModel.Clients.ActiveDirectory.ClientAssertionCertificate($ClientId,  
$certCollection[0])
```

```
$Token = ($authContext.AcquireToken($Resource, $cacert)).AccessToken
```

3. Leave the **Administrator: Windows PowerShell** window open.

Task 7: Formulate the Request URI and retrieve Tenant Usage using the REST API

1. In the **Administrator: Windows PowerShell** window that was left open from the previous task, enter the following command replacing the **GUID** with the **App-Service-Websites Subscription ID** from **Step 4** in **Task 6**. Also modify the **reportedStartTime** so that the date is the start date of this course and the **reportedEndTime** date is yesterday's date.

```
$SubscriptionURI =  
'https://management.local.azurestack.external/subscriptions/07a828d0-e911-4005-  
a0ab-  
513bdbb8072c/providers/Microsoft.Commerce/UsageAggregates?reportedStartTime=2  
018-01-01&reportedEndTime=2018-02-20&aggregationGranularity=Daily&api-  
version=2015-06-01-preview'
```

2. Enter the following commands pressing enter after each command:

```
$Headers = @{'authorization'="Bearer $($Token)"}
```

```
$Request = Invoke-RestMethod -Method GET -ContentType 'application/json' -Headers  
$Headers -Uri $SubscriptionURI
```

```
$Request.value.properties | Format-List
```

3. Review the usage information returned. This information is taken from the **App-Service-Websites** subscription. You can correlate the **meterID** values with the **Meters** shown in the **Tenant-Based Usage** topic to determine the actual resources used.
4. Type the following command and then press enter:

```
$Request.value.properties | Measure
```

5. Note the **Count** value returned. This is the number of resource usage records recorded.
6. Close the Administrator: Windows PowerShell window.

Task 8: Review Azure Stack resource consumption using the Azure Stack operator portal

1. On **AZS-RDP01**, 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 the **Administration** section, click **Region Management**.
6. In the **local** blade that opens, review the following information:
 - Alerts
 - Update
 - Resource Providers
 - Infrastructure Roles
7. From the navigation pane, under **Resource Providers**, click **Capacity**.
8. In the **local – Capacity** blade that opens, review the following information:
 - Alerts
 - Scale Units
 - Physical storage
 - Physical memory
 - Public IP pools usage
9. From the navigation pane, click **Scale Units**.
10. In the **local – Scale units** blade that opens, click **S-Cluster**.
11. In the **S-Cluster** blade that opens, review the following information:
 - Region
 - Total logical cores
 - Type

- Total memory
12. Close the **S-Cluster** blade.
 13. From the navigation pane, under **Tenant Resources**, click **Virtual networks**.
 14. Review the information provided in the **local – Virtual networks** blade that opens.
 15. Close the **Microsoft Azure Stack - Administration** portal.
 16. Close all open windows.

Results: After completing this exercise, you will be able to: Obtain Tenant usage information in Azure Stack using PowerShell. Obtain Tenant and Provider usage information in Azure Stack using the REST API's. Obtain Azure Stack resource consumption by using the Administration 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.