

# AZ-600: Configuring and Operating Microsoft Azure Stack Hub

- **Download Latest Student Handbook and AllFiles Content**  
`/home/ll/Azure_clone/Azure_new/AZ-600-Configuring-and-Operating-Microsoft-Azure-Stack-Hub/../../releases/latest`
- **Are you a MCT?** - Have a look at our [GitHub User Guide for MCTs](#)
- **Need to manually build the lab instructions?** - Instructions are available in the [MicrosoftLearning/Docker-Build](#) repository

## What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

## **How should I use these files relative to the released MOC files?**

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

## **What about changes to the student handbook?**

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

## **How do I contribute?**

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

# **Notes**

## **Classroom Materials**

**It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.**

title: Online Hosted Instructions permalink: index.html layout: home

---

# Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.



# Labs

```
{% assign labs = site.pages | where_exp:"page", "page.url contains  
'/Instructions/Labs'" %} | Module | Lab | | --- | --- | {% for activity in labs  
%}| {{ activity.lab.module }} | {{ activity.lab.title }} | {% if  
activity.lab.type %} - {{ activity.lab.type }} | {% endif %} | {% endfor %}
```

## Demos

```
{% assign demos = site.pages | where_exp:"page", "page.url contains  
'/Instructions/Demos'" %} | Module | Demo | | --- | --- | {% for activity in  
demos %}| {{ activity.demo.module }} | .{{ activity.demo.title }} | {%  
endfor %}
```

---

demo: title: 'Demo: Deploying an ARM Template' module: 'Module 1:  
Exploring Azure Resource Manager'

---

# Demo: Deploying an ARM Template

## Instructions

1. Quisque dictum convallis metus, vitae vestibulum turpis dapibus non.

1. Suspendisse commodo tempor convallis.

2. Nunc eget quam facilisis, imperdiet felis ut, blandit nibh.

3. Phasellus pulvinar ornare sem, ut imperdiet justo volutpat et.

2. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

3. Vestibulum hendrerit orci urna, non aliquet eros eleifend vitae.

4. Curabitur nibh dui, vestibulum cursus neque commodo, aliquet accumsan risus.

`Sed at malesuada orci, eu volutpat ex`

5. In ac odio vulputate, faucibus lorem at, sagittis felis.

6. Fusce tincidunt sapien nec dolor congue facilisis lacinia quis urna.

**Note:** Ut feugiat est id ultrices gravida.

7. Phasellus urna lacus, luctus at suscipit vitae, maximus ac nisl.

◦ Morbi in tortor finibus, tempus dolor a, cursus lorem.

◦ Maecenas id risus pharetra, viverra elit quis, lacinia odio.

◦ Etiam rutrum pretium enim.

# 1. Curabitur in pretium urna, nec ullamcorper diam.

lab: title: 'Lab: Manage offers and plans in Azure Stack Hub' module: 'Module 2: Provide Services'

---

# **Lab - Manage offers and plans in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## Estimated Time

60 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to implement offers and plans for your users. You need to ensure that all users have the same set of base services but want to provide access to additional services only to some users.

# Objectives

After completing this lab, you will be able to:

- Implement Azure Stack Hub offers and plans by using the Azure Stack Hub administrator portal



# Lab Environment

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will create additional user accounts in the course of this lab.

## Exercise 0: Prepare for the lab

In this exercise, you will create Active Directory user accounts that you will be using in this lab:

1. Create user accounts (as a cloud operator)

### Task 1: Create the first user account (as a cloud operator)

1. If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**

2. Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
3. In the **Active Directory Administrative Center** console, click **azurestack (local)**.
4. In the details pane, double-click the **Users** container.
5. In the **Tasks** pane, in the **Users** section, click **New -> User**.
6. In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**
7. Repeat the steps 5-6 to create another user account with the following settings:
  - Full name: **T1U2**
  - User UPN logon: **t1u2@azurestack.local**
  - User SamAccountName: **azurestack\t1u2**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**

**Review:** In this exercise, you have created the Active Directory user accounts you will use in this lab.

## **Exercise 1: Create offers (as a cloud operator)**

In this exercise, you will act as a cloud operator and create a plan consisting of the compute, storage, and network services as well as an offer containing this plan. Next, you will make the offer public, allowing users to create subscriptions based on this offer. The exercise consists of the following tasks:

1. Create plans consisting of the compute, storage, and network services (as a cloud operator).
2. Create a public offer based on the plan (as a cloud operator)
3. Create a private offer based on the plan (as a cloud operator)

### **Task 1: Create plans consisting of the compute, storage, and network services (as a cloud operator)**

In this task, you will:

- Create two plans consisting of the compute, storage, and network services (as a cloud operator).
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **base-plan1**
  - Resource name: **base-plan1**
  - Resource group: the name of a new resource group **base-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Compute**, **Microsoft.Storage**, and **Microsoft.Network** checkboxes.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, specify the following settings
  - Microsoft.Compute: **Default Quota**
  - Microsoft.Storage: **Default Quota**
  - Microsoft.Network: **Default Quota**

- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **base-plan2**
  - Resource name: **base-plan2**
  - Resource group: the name of a new resource group **base-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Compute**, **Microsoft.Storage**, and **Microsoft.Network** checkboxes.
- Click **Next: Quotas>**.
- On the **Quotas** tab of the **New plan** blade, next to the **Microsoft.Compute** drop-down list, click **Create New**.
- On the **Create Compute quota** blade, specify the following settings and click **OK**:
  - Name: **base-plan1-compute-quota**
  - Number of virtual machines: **40**
  - Number of virtual machine cores: **100**
  - Number of availability sets: **20**
  - Number of virtual machine scale sets: **40**
  - Capacity (GB) of standard managed disks: **4096**
  - Capacity (GB) of premium managed disks: **4096**
- Back on the **Quotas** tab of the **New plan** blade, next to the **Microsoft.Storage** drop-down list, click **Create New**.
- On the **Create Storage quota** blade, specify the following settings and click **OK**:

- Name: **base-plan1-storage-quota**
  - Maximum capacity (GB): **4096**
  - Total number of storage accounts: **40**
- Back on the **Quotas** tab of the **New plan** blade, next to the **Microsoft.Network** drop-down list, click **Create New**.
- On the **Create Network quota** blade, specify the following settings and click **OK**:
  - Name: **base-plan1-network-quota**
  - Max virtual networks: **100**
  - Max virtual network gateways: **2**
  - Network connections: **4**
  - Max public IPs: **100**
  - Max NICs: **200**
  - Max load balancers: **100**
  - Max network security groups: **100**
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

## **Task 2: Create a public offer based on the plan (as a cloud operator)**

In this task, you will:

- Create a public offer based on the plan (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **base-offer1**
  - Resource name: **base-offer1**
  - Resource group: **base-offers-RG**

- Make this offer public: **Yes**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **base-plan1** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### **Task 3: Create a private offer based on the plan (as a cloud operator)**

In this task, you will:

- Create an private offer based on the plan (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **base-offer2**
  - Resource name: **base-offer2**
  - Resource group: **base-offers-RG**
  - Make this offer public: **No**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **base-plan2** entry.
- Click **Next: Add-on plans >**.

- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

**Review:** In this exercise, you have created a plan and two offer based on that plan, with one of them being public and the other private.

## **Exercise 2: Create subscriptions based on the offers (as users)**

In this exercise, you will create subscriptions based on two offers you created in the previous exercise. The exercise consists of the following tasks:

1. Sign up for the offer (as the first user)
2. Assign a subscription to the second user (as an operator)

### **Task 1: Sign up for the offer (as the first user)**

In this task, you will:

- Sign up for the offer (as the first user)
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, on the Dashboard, click the **Get a subscription** tile.
- On the **Get a subscription** blade, in the **Name** text box, type **t1u1-base-subscription1**.
- In the list of offers, select **base-offer1** and click **Create**.
- When presented with the message **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.

- In the Azure Stack Hub tenant portal, in the hub menu, click **All services**.
- In the list of services, click **Subscriptions**.
- On the **Subscriptions** blade, click **t1u1-base-subscription1**.
- On the **t1u1-base-subscription1** blade, click **Resources**.
- On the **t1u1-base-subscription1 - Resources** blade, click **+ Add**.
- On the **New** blade, click **Security + Identity** and then click **Key Vault**.
- On the **Basics** tab of the **Create key vault** blade, specify the following settings (leave others with their default values):
  - Resource group: the name of a new resource group **kv-RG**
  - Key vault name: any unique name consisting of between 3 and 24 alphanumeric characters and dashes, starting with a letter
- Click **Review + create** and then click **Create**.
- Verify that the deployment failed.

**Note:** Review the error message **The resource namespace 'Microsoft.KeyVault' is invalid**. You cannot create the key vault since it was not included in the offer.

- Sign out from the Azure Stack Hub user portal and close the InPrivate session of the web browser.

## **Task 2: Assign a subscription to the second user (as an operator)**

In this task, you will:

- Assign a subscription to the second user (as an operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the [Azure Stack Hub administrator portal](#) where you are signed in as CloudAdmin@azurestack.local, select **All resources**.
- On the **All resources** blade, search for and select **base-offer2**.
- On the **base-offer2** blade, select **Subscriptions**.
- On the **base-offer2 | Subscriptions** blade, click **+ Add**.



- On the **Create a user subscription** blade, specify the following settings and click **Create**.
  - Name: **t1u2-base-subscription1**
  - User: **t1u2@azurestack.local**
  - Directory tenant: **ADFS.azurestack.local**
  - Offer name: **base-offer2**
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u2@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, in the hub menu, select **All services** and then, on the **All services** blade, **Subscriptions**.
- On the **Subscriptions** blade, select **t1u2-base-subscription1**.
- On the **t1u2-base-subscription1** blade, click **Resources**.
- On the **t1u2-base-subscription1 - Resources** blade, click **+ Add**.
- On the **New** blade, click **Security + Identity** and then click **Key Vault**.
- On the **Basics** tab of the **Create key vault** blade, specify the following settings (leave others with their default values):
  - Resource group: the name of a new resource group **kv-RG**
  - Key vault name: any unique name consisting of between 3 and 24 alphanumeric characters and dashes, starting with a letter
- Click **Review + create** and then click **Create**.
- Verify that the deployment failed.

**Note:** Review the error message **The resource namespace 'Microsoft.KeyVault' is invalid**. You cannot create the key vault since it was not included in the offer.

- Sign out from the Azure Stack Hub user portal and close the InPrivate session of the web browser.

**Review:** In this exercise, you have subscribed to the offer as two users, creating this way two new user subscriptions.

### Exercise 3: Create an add-on plan and assign it to a user subscription (as a cloud operator)

In this exercise, you will act as a cloud operator and create an add-on plan consisting of the Key Vault service. Next, you will assign it to one of the user subscriptions.

1. Create an add-on plan including the Key Vault service (as a cloud operator).
2. Assign the add-on plan to an existing offer (as a cloud operator)

#### Task 1: Create an add-on plan including the Key Vault service (as a cloud operator)

In this task, you will:

- Create an add-on plan consisting of the Key Vault service (as a cloud operator).
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal where you are signed in as CloudAdmin@azurestack.local, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Plan**.
- On the **New plan** blade, specify the following settings:
  - Display name: **add-on-plan2**
  - Resource name: **add-on-plan2**
  - Resource group: the name of a new resource group **add-on-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.KeyVault** checkbox.
- Click **Next: Quotas>**.
- On the **Quotas** tab of the **New plan** blade, in the **Microsoft.KeyVault** drop-down list, select **Unlimited**.

- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

## **Task 2: Assign the add-on plan to an existing offer (as a cloud operator)**

In this task, you will:

- Assign the add-on plan to an existing offer (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal where you are signed in as **CloudAdmin@azurestack.local**, click **All resources**.
- In the list of resources, search for and select **base-offer2**.
- On the **base-offer2** blade, select **Add-on plans**.
- On the **base-offer2 | Add-on plans** blade, click **+ Add**.
- On the **Plan** blade, click the **add-on-plan2** and click **Select**.

**Review:** In this exercise, you have created an add-on plan and assigned it to an existing offer associated with the subscription of the second user.

## **Exercise 4: Check availability of the add-on plan (as the second user)**

In this exercise, you will act as users who signed up for the offer you created in the first exercise and created new subscriptions. The exercise consists of the following tasks:

1. Modify the subscription by adding the add-on plan (as the second user)
2. Verify the functionality of the add-on plan (as the second user)

### **Task 1: Modify the subscription by adding the add-on plan (as the second user)**

In this task, you will:

- Check availability of the add-on plan (as the second user)
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the hub menu of the Azure Stack Hub user portal, click **All services**.
- In the list of services, click **Subscriptions**.
- On the **Subscriptions** blade, click **t1u2-base-subscription1**.
- On the **t1u2-base-subscription1** blade, click **+ Add plan**.
- On the **Add plan** blade, select **add-on-plan2**. If prompted, refresh the web browser page displaying the Azure Stack user portal.
- Back on the **t1u2-base-subscription1** blade, click **Add-on plans**.
- On the **t1u2-base-subscription2 | Add-on plans** blade, note the entry representing the **add-on-plan2**.

## Task 2: Verify the functionality of the add-on plan (as the second user)

In this task, you will:

- Check availability of the add-on plan (as the second user)
- Within the Remote Desktop session to **AzS-HOST1**, in the InPrivate session of a web browser displaying the Azure Stack Hub user portal where you are signed in as **t1u2@azurestack.local**,
- On the **t1u2-base-subscription1** blade, click **Resources**.
- On the **t1u2-base-subscription1 - Resources** blade, click **+ Add**.
- On the **New** blade, click **Security + Identity** and then click **Key Vault**.
- On the **Basics** tab of the **Create key vault** blade, specify the following settings (leave others with their default values):
  - Resource group: **kv-RG**
  - Key vault name: any unique name consisting of between 3 and 24 alphanumeric characters and dashes, starting with a letter

- Click **Review + create** and then click **Create**.
- Verify that the deployment was successful.
- Sign out from the Azure Stack Hub user portal and close the InPrivate session of a web browser.

**>Review: In this exercise, you have verified the functionality of the add-on plan.**

lab: title: 'Lab: Add custom Marketplace Items by using Azure Gallery Packager' module: 'Module 2: Provide Services'

---

# **Lab - Add custom Marketplace Items by using the Azure Gallery Packager**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

45 minutes



## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to create custom Azure Stack Marketplace items by using the Azure Gallery Packager tool.

# Objectives

After completing this lab, you will be able to:

- Create custom Azure Stack Hub Marketplace items by using the Azure Gallery Packager

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will download and install software necessary to create custom Azure Stack Marketplace items in the course of this lab.

## Exercise 1: Customize and publish Azure Stack Hub Marketplace items

In this exercise, you will customize and publish Azure Marketplace items by using the Azure Gallery Packager tool.

1. Download the Azure Gallery Packager tool and sample packages
2. Modify an existing Azure Gallery Packager package
3. Upload the package to an Azure Stack Hub storage account
4. Publish the package to Azure Stack Hub Marketplace

5. Verify availability of the published Azure Stack Hub Marketplace item

## **Task 1: Download the Azure Gallery Packager tool and sample packaging files**

In this task, you will:

- Download the Azure Gallery Packager tool and sample packaging files
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, open a web browser window, navigate to the [Azure Gallery Packager tool download page](#) and download the **Microsoft Azure Stack Gallery Packaging Tool and Sample 3.0.zip** archive file to the **Downloads** folder.
- Once the download completes, extract the **Packager** folder within the zip file into the **C:\Downloads** folder (create the folder if needed).

## **Task 2: Modify an existing Azure Gallery Packager package**

In this task, you will:

- Modify an existing Azure Gallery Packager package, such that it will use a Windows Server 2019 image (rather than Windows Server 2016).
- Within the Remote Desktop session to **AzS-HOST1**, in File Explorer, navigate to the **C:\Downloads\Packager\Samples for Packager** folder, copy the **Sample.SingleVMWindowsSample.1.0.0.azpkg** package to the **C:\Downloads** folder, and rename its extension to **.zip**.

- Extract the content of the **Sample.SingleVMWindowsSample.1.0.0.zip** archive to the **C:\Downloads\SamplePackage** folder (you will need to create it first).
- In File Explorer, navigate to the **C:\Downloads\SamplePackage\DeploymentTemplates** folder and open the **createuidefinition.json** file in Notepad.
- Modify the **sku** parameter in the **imageReference** section of the file such that it references a Windows Server 2019 Datacenter Core image which was downloaded earlier to the ASDK instance:

```
json "imageReference": { "publisher":  
"MicrosoftWindowsServer", "offer": "WindowsServer",  
"sku": "2019-Datacenter-Core-smalldisk" },
```

- Save the changes and close Notepad.
- In File Explorer, navigate to the **C:\Downloads\SamplePackage\strings** folder and open the **resources.resjson** file in Notepad.
- Modify the content of the **resources.resjson** file by setting the following values in the key-value pairs :
  - displayName: **Custom Windows Server 2019 Core VM**
  - publisherDisplayName: **Contoso**
  - summary: **Custom Windows Server 2019 Core VM (small disk)**
  - longSummary: **Custom Contoso Windows Server 2019 Core VM (small disk)**
  - description:

**Sample customized Windows Server 2019 Azure Stack Hub VM**

**Based on Azure Stack Hub Quickstart template**

- documentationLink: **Documentation**

**Note:** This should yield the following content:

```
json { "displayName": "Custom Windows Server 2019 Core  
VM", "publisherDisplayName": "Contoso", "summary":
```

```
"Custom Windows Server 2019 Core VM (small disk)",  
"longSummary": "Custom Contoso Windows Server 2019 Core  
VM (small disk)", "description": "<p>Sample customized  
Windows Server 2019 Azure Stack Hub VM</p><p>Based on  
Azure Stack Hub Quickstart template</p>",  
"documentationLink": "Documentation" }
```

- Save the changes and close Notepad.
- In File Explorer, navigate to the **C:\Downloads\SamplePackage** folder and open the **manifest.json** file in Notepad.
- Modify the content of the **manifest.json** file by setting the following values in the key-value pairs :
  - name: **CustomVMWindowsSample**
  - publisher: **Contoso**
  - version: **1.0.1**
  - displayName: **Custom Windows Server 2019 Core VM**
  - publisherDisplayName: **Contoso**
  - publisherLegalName: **Contoso Inc.**
  - uri of the documentation link: **<https://docs.contoso.com>**

**Note:** This should yield the following content (starting with the line 2 of the file):

```
json "$schema": "https://gallery.azure.com/schemas/2015-  
10-01/manifest.json#", "name": "CustomVMWindowsSample",  
"publisher": "Contoso", "version": "1.0.1",  
"displayName": "Custom Windows Server 2019 Core VM",  
"publisherDisplayName": "Contoso", "publisherLegalName":  
"Contoso Inc.", "summary": "ms-resource:summary",  
"longSummary": "ms-resource:longSummary", "description":  
"ms-resource:description", "longDescription": "ms-  
resource:description", "uiDefinition": { "path":  
"UIDefinition.json" }, "links": [ { "displayName": "ms-  
resource:documentationLink", "uri":  
"https://docs.contoso.com/" } ],
```

- Save the changes and close Notepad.

### **Task 3: Generate the customized Azure Gallery Packager package**

In this task, you will:

- Regenerate the newly customized Azure Gallery Packager package.
- Within the Remote Desktop session to **AzS-HOST1**, start **Command Prompt**.
- From the **Command Prompt**, run the following to change the current directory:

```
cmd cd C:\Downloads\Packager
```

- From the **Command Prompt**, run the following to generate a new package based on the content you modified in the previous task:

```
cmd AzureStackHubGallery.exe package -m  
C:\Downloads\SamplePackage\manifest.json -o C:\Downloads\
```

- Verify that the **Contoso.CustomVMWindowsSample.1.0.1.azpkg** package was automatically saved to the **C:\Downloads** folder.

#### Task 4: Upload the package to an Azure Stack Hub storage account

In this task, you will:

- Upload the Azure Stack Hub Marketplace item package to an Azure Stack Hub storage account.
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Data + Storage**.
- On the **Data + Storage** blade, click **Storage account**.
- On the **Basics** tab of the **Create storage account** blade, specify the following settings:
  - Subscription: **\*\*Default Provider Subscription**
  - Resource group: the name of a new resource group **marketplace-pkgs-RG**
  - Name: a unique name consisting of between 3 and 24 lower case letters or digits

- Location: **local**
- Performance: **Standard**
- Account kind: **Storage (general purpose v1)**
- Replication: **Locally-redundant storage (LRS)**
- On the **Basics** tab of the **Create storage account** blade, click **Next: Advanced >**.
- On the **Advanced** tab of the **Create storage account** blade, leave the default settings in place and click **Review + create**.
- On the **Review + create** tab of the **Create storage account** blade, click **Create**.

**Note:** Wait until the storage account is provisioned. This should take about one minute.

- In the web browser window displaying the Azure Stack Hub administrator portal, in the hub menu, select **Resource groups**.
- On the **Resource group** blade, in the list of resource groups, click the **marketplace-pkgs-RG** entry.
- On the **marketplace-pkgs-RG** blade, click the entry representing the newly created storage account.
- On the storage account blade, click **Containers**.
- On the Containers blade, click **+ Container**.
- On the **New container** blade, in the **Name** textbox, type **gallerypackages**, in the **Public access level** drop down list, select **Blob (anonymous read access for blobs only)**, and click **Create**.
- Back on the Containers blade, click the **gallerypackages** entry representing the newly created container.
- On the **gallerypackages** blade, click **Upload**.
- On the **Upload blob** blade, click the folder icon next to the **Select a file** text box.
- In the **Open** dialog box, navigate to the **C:\Downloads** folder, select the **Contoso.CustomVMWindowsSample.1.0.1.azpkg** package file and click **Open**.
- Back on the **Upload blob** blade, click **Upload**.

## **Task 5: Publish the package to Azure Stack Hub Marketplace**

In this task, you will:



- Publish the package to Azure Stack Hub Marketplace.
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as administrator.
- Within the Remote Desktop session to **AzS-HOST1**, from the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to install the Azure Stack Hub PowerShell modules required for this lab:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Install-Module -Name
Az.BootStrapper -Force -AllowPrerelease -AllowClobber
Install-AzProfile -Profile 2019-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 2.0.2-
preview -AllowPrerelease
```

**Note:** Disregard any error messages regarding already available commands.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to register your Azure Stack Hub operator environment:

```
powershell Add-AzEnvironment -Name 'AzureStackAdmin' -
ArmEndpoint
'https://adminmanagement.local.azurestack.external' ` -
AzureKeyVaultDnsSuffix
adminvault.local.azurestack.external ` -
AzureKeyVaultServiceEndpointResourceId
https://adminvault.local.azurestack.external
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to sign in to your Azure Stack Hub operator PowerShell environment with the AzureStack\CloudAdmin credentials by leveraging your existing browser session to the Azure Stack Hub administrator portal:

```
powershell Connect-AzAccount -EnvironmentName
'AzureStackAdmin'
```

**Note:** This will automatically open another browser tab displaying the message informing you about successful authentication.

- Close the browser tab, switch back to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and verify that you have successfully authenticated as **CloudAdmin@azurestack.local**.
- From the Administrator: **Administrator: C:\Program Files\PowerShell\7\pwsh.exe**, run the following to publish the package to Azure Stack Hub Marketplace (where the `<storage_account_name>` placeholder represents the name of the storage account you assigned in the previous task):
 

```
powershell Add-AzsGalleryItem -GalleryItemUri `
https://<storage_account_name>.blob.local.azurestack.external/gallerypackages/Contoso.CustomVMWindowsSample.1.0.1.azpkg -Verbose
```
- Review the output of the **Add-AzsGalleryItem** command to verify that the command completes successfully.

## Task 6: Verify availability of the published Azure Stack Hub Marketplace item

In this task, you will:

- Verify availability of the published Azure Stack Hub Marketplace item.
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub user portal](#).
- In the web browser window displaying the Azure Stack Hub administrator portal, click **Marketplace**.
- On the **Marketplace** blade, click **Compute** and then click **See More**.
- On the **Marketplace** blade, ensure that the **Custom Windows Server 2019 Core VM** appears in the list of choices.

**Note:** In order to ensure that the new Azure Stack Hub Marketplace item is functioning as intended, you would also need to ensure that all of the prerequisites for its deployment, such as OS images referenced by its template, are in place. This is beyond the scope of this lab.

**>Review: In this exercise, you have created a custom Azure Stack Hub Marketplace item by using the Azure Gallery Packager and published it by using the Add-AzsGalleryItem cmdlet.**

lab: title: 'Lab: Validate Azure Resource Manager (ARM) Templates with Azure Stack Hub' module: 'Module 2: Provide Services'

---

# **Lab - Validate Azure Resource Manager (ARM) Templates with Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## Estimated Time

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to use your existing Azure Resource Manager (ARM) templates to automate deployment of Azure Stack Hub resources.

# Objectives

After completing this lab, you will be able to:

- Validate ARM templates for Azure Stack Hub deployments.

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment has the following configuration:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab.

## Exercise 1: Validate an ARM template with Azure Stack Hub

In this exercise, you will validate an ARM template with Azure Stack Hub.

1. Build a cloud capabilities file
2. Run a successful template validation
3. Run a failed template validation
4. Remediate template issues



## Task 1: Build a cloud capabilities file

In this task, you will:

- Build a cloud capabilities file
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as administrator.
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to configure PowerShell Gallery as a trusted repository

```
powershell Set-PSRepository -Name 'PSGallery' -
InstallationPolicy Trusted
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to install PowerShellGet:

```
powershell Install-Module PowerShellGet -MinimumVersion
2.2.3 -Force
```

**Note:** Disregard any warning messages regarding in-use modules.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to install the PowerShell Az module for Azure Stack Hub:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Install-Module -Name
Az.BootStrapper -Force -AllowPrerelease -AllowClobber
Install-AzProfile -Profile 2019-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 2.0.2-
preview -AllowPrerelease
```

**Note:** Disregard any error messages regarding already available commands.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to register your Azure Stack Hub operator PowerShell environment:

```
powershell Add-AzEnvironment -Name 'AzureStackAdmin' -
ArmEndpoint
'https://adminmanagement.local.azurestack.external' ` -
AzureKeyVaultDnsSuffix
adminvault.local.azurestack.external ` -
AzureKeyVaultServiceEndpointResourceId
https://adminvault.local.azurestack.external
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to sign in to the newly registered **AzureStackAdmin** environment:

```
powershell Connect-AzAccount -EnvironmentName
'AzureStackAdmin'
```

- If prompted, authenticate with the **CloudAdmin@azurestack.local** account.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to download the Azure Stack Tools:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Set-Location -Path
'C:\' Invoke-WebRequest
https://github.com/Azure/AzureStack-Tools/archive/az.zip
-OutFile az.zip Expand-Archive az.zip -DestinationPath .
-Force Set-Location -Path '\AzureStack-Tools-az'
```

**Note:** This step copies the archive containing the GitHub repository hosting the Azure Stack Hub tools to the local computer and expands the archive to the **C:\AzureStack-Tools-master** folder. The tools contain PowerShell modules that offer a range of features, including validation of Azure Stack Hub Resource Manager templates.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to import

the AzureRM.CloudCapabilities PowerShell module:

```
powershell Import-Module  
.\CloudCapabilities\Az.CloudCapabilities.psm1 -Force
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to generate a cloud capabilities JSON file:

```
powershell $path = 'C:\Templates' New-Item -Path $path -  
ItemType Directory -Force Get-AzCloudCapability -Location  
'local' -OutputPath $path\AzureCloudCapabilities.Json -  
Verbose
```

- Within the Remote Desktop session to **AzS-HOST1**, start File Explorer, navigate to the **C:\Templates** folder and verify that the file **AzureCloudCapabilities.Json** has been successfully created.

## Task 2: Run a successful template validation

In this task, you will:

- Run template validator against an Azure Stack Hub Quickstart template
- Within the Remote Desktop session to **AzS-HOST1**, start a web browser and navigate to the Azure Stack Hub QuickStart Templates repository [MySQL Server on Windows for AzureStack page](#).
- On the **MySQL Server on Windows for AzureStack** page, click **azuredeploy.json**.
- On the [AzureStack-QuickStart-Templates/mysql-standalone-server-windows/azuredeploy.json](#) page, review the content of the template.
- Switch to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and run the following to download the azuredeploy.json file and save it as a file named **sampletemplate1.json** in the **C:\Templates** folder.

```
powershell Invoke-WebRequest -Uri  
'https://raw.githubusercontent.com/Azure/AzureStack-  
QuickStart-Templates/master/mysql-standalone-server-  
windows/azuredeploy.json' -UseBasicParsing -OutFile  
$path\sampletemplate1.json
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to import the AzureRM.TemplateValidator PowerShell module:

```
powershell Set-Location -Path 'C:\AzureStack-Tools-az\TemplateValidator' Import-Module .\AzureRM.TemplateValidator.psm1 -Force
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to validate the template:

```
powershell Test-AzureRMTemplate ` -TemplatePath $path ` -TemplatePattern sampletemplate1.json ` -CapabilitiesPath $path\AzureCloudCapabilities.json ` -IncludeComputeCapabilities ` -IncludeStorageCapabilities ` -Report sampletemplate1validationreport.html ` -Verbose
```

- Review the output of the validation and verify that there are no issues.

**Note:** The output should have the following format:

```
Validation Summary: Passed: 1 NotSupported: 0 Exception: 0 Warning: 0 Recommend: 0 Total Templates: 1 Report available at - C:\AzureStack-Tools-az\TemplateValidator\sampletemplate1validationreport.html
```

### Task 3: Run a failed template validation

In this task, you will:

- Run template validator against an Azure Quickstart template
- Within the Remote Desktop session to **AzS-HOST1**, from the web browser displaying the AzureStack QuickStart Templates repository, navigate to the Azure QuickStart Templates repository [MySQL Server 5.6 on Ubuntu VM page](#)
- On the **MySQL Server 5.6 on Ubuntu VM** page, click **azuredeploy.json**.
- On the [azure-quickstart-templates/mysql-standalone-server-ubuntu/azuredeploy.json](#) page, review the content of the template.

- Switch to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and run the following to download the `azuredeploy.json` file and save it as a file named **sampletemplate2.json** in the **C:\Templates** folder.

```
powershell Invoke-WebRequest -Uri
'https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/mysql-standalone-server-ubuntu/azuredeploy.json' -UseBasicParsing -OutFile
$spath\sampletemplate2.json
```

- Within the Remote Desktop session to **AzS-HOST1**, in the web browser, navigate to the REST API reference for [Virtual Machines](#) and identify the latest Azure API version (**2020-12-01** at the time of authoring this content).
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to open the **sampletemplate2.json** file in Notepad.

```
powershell notepad.exe $spath\sampletemplate2.json
```

- In the Notepad window displaying the content of the **sampletemplate2.json** file, locate the section representing the virtual machine resource in the `resources` section of the template, which has the following format:

```
json { "apiVersion": "2017-03-30", "type":
"Microsoft.Compute/virtualMachines", "name": "
[variables('vmName')]", "location": "
[parameters('location')]", "dependsOn": [ "
[concat('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))]", "
[concat('Microsoft.Network/networkInterfaces/',
variables('nicName'))]" ],
```

- Set the value of the `apiVersion` key to the latest Azure REST API version for virtual machines you identified earlier in this task and save the change, leaving the file open.

**Note:** Record the original value before you make the change. You will need to revert to it in the next task.

**Note:** Assuming the REST API version **2020-12-01**, the change should result in the following outcome:

```
json { "apiVersion": "2020-12-01", "type":
"Microsoft.Compute/virtualMachines", "name": "
[variables('vmName')]", "location": "
[parameters('location')]", "dependsOn": [ "
[concat('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))]", "
[concat('Microsoft.Network/networkInterfaces/',
variables('nicName'))]" ],
```

**Note:** This intentionally introduces a configuration which is not yet supported in Azure Stack Hub.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to validate the newly modified template:

```
powershell Test-AzureRMTemplate ` -TemplatePath $path ` -
TemplatePattern sampletemplate2.json ` -CapabilitiesPath
$path\AzureCloudCapabilities.json ` -
IncludeComputeCapabilities ` -IncludeStorageCapabilities
` -Report sampletemplate2validationreport.html ` -Verbose
```

- Review the output of the validation and note that this time there is a supportability issue.

**Note:** The output should have the following format:

```
Validation Summary: Passed: 0 NotSupported: 1 Exception:
0 Warning: 0 Recommend: 0 Total Templates: 1 Report
available at - C:\AzureStack-Tools-
az\TemplateValidator\sampletemplate2validationreport.html
```

- Open the **C:\AzureStack-Tools-az\TemplateValidator\sampletemplate2validationreport.html** file and review the report.

**Note:** The report should contain an entry in the following format: **NotSupported: apiversion (Resource type: Microsoft.Compute/virtualMachines). Not Supported Values - 2020-12-01.**

## Task 4: Remediate template issues

In this task, you will:

- Remediate template issues.
- Within the Remote Desktop session to **AzS-HOST1**, in File Explorer, navigate to the **C:\Templates** folder and open the **AzureCloudCapabilities.json** file.
- In the **AzureCloudCapabilities.json** file, locate the "ResourceTypeName": "virtualMachines", section, which should have the following format:

```
json { "ProviderNamespace": "Microsoft.Compute",
  "ResourceTypeName": "virtualMachines", "Locations": [
    "local" ], "ApiVersions": [ "2020-06-01", "2019-12-01",
    "2019-07-01", "2019-03-01", "2018-10-01", "2018-06-01",
    "2018-04-01", "2017-12-01", "2017-03-30", "2016-08-30",
    "2016-03-30", "2015-11-01", "2015-06-15" ],
  "ApiProfiles": [ "2017-03-09-profile", "2018-03-01-
  hybrid" ] },
```

- Switch to the **sampletemplate2.json** file and change the value of the REST API version you modified in the previous task to its original value. Ensure that this version matches one of the API versions in the **AzureCloudCapabilities.json** listed above and save the change.
- Switch to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and run the following to validate the newly modified template:

```
powershell Test-AzureRMTemplate ` -TemplatePath $path ` -
TemplatePattern sampletemplate2.json ` -CapabilitiesPath
$path\AzureCloudCapabilities.json ` -
IncludeComputeCapabilities ` -IncludeStorageCapabilities
` -Report sampletemplate2validationreport.html ` -Verbose
```

- Review the output of the validation and note that this time there are no issues.

**>Review: In this exercise, you have created a cloud capabilities file and used it to validate Azure Resource Manager templates. You also modified a template based on the result of the validation.**

lab: title: 'Lab: Implement SQL Server Resource Provider in Azure Stack Hub' module: 'Module 2: Provide Services'

---



# **Lab - Implement SQL Server Resource Provider in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

150 minutes

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to allow your tenants to deploy SQL Server databases.

# Objectives

After completing this lab, you will be able to:

- Implement SQL Server resource provider in Azure Stack Hub.

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment has the following configuration:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab.

## Exercise 1: Install the SQL Server resource provider in Azure Stack Hub

In this exercise, you will install the SQL Server resource provider in Azure Stack Hub.

1. Download SQL Server resource provider binaries
2. Install the SQL Server resource provider
3. Verify installation of the SQL Server resource provider

**Note:** To minimize the duration of this exercise, some of the tasks necessary to install the Azure Stack Hub SQL Server resource provider have already been completed, including the following:

- Implementing Azure Marketplace syndication
- Downloading **Microsoft AzureStack Add-On RP Windows Server** from Azure Marketplace

## **Task 1: Download SQL Server resource provider binaries**

In this task, you will:

- Download SQL Server resource provider binaries
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- Within the Remote Desktop session to **AzSHOST-1**, in the web browser displaying the Azure Stack administrator portal, in the hub menu, click **All services**.
- On the **All services** blade, search for and select **Marketplace management**
- On the Marketplace management blade, verify that **Microsoft AzureStack Add-On RP Windows Server** appears in the list of available services.
- Within the Remote Desktop session to **AzSHOST-1**, start another web browser window, download the SQL Resource Provider self-extracting executable from (<https://aka.ms/azshsqlrp11931>) and extract its content into the **C:\Downloads\SQLRP** folder (you will need to create the folder first).

## **Task 2: Install the SQL Server resource provider**

In this task, you will:

- Install SQL Server resource provider
- Within the Remote Desktop session to **AzSHOST-1**, start Windows PowerShell as administrator.

**Note:** Make sure to start a new PowerShell session.

- From the **Administrator: Windows PowerShell** prompt, run the following to configure PowerShell Gallery as a trusted repository

```
powershell Set-PSRepository -Name 'PSGallery' -
InstallationPolicy Trusted
```

- From the **Administrator: Windows PowerShell** prompt, run the following to install the version of the AzureRM.Bootstrapper module required by the SQL Server resource provider:

```
```powershell Get-Module -Name Azs. -ListAvailable | Uninstall-
Module -Force -Verbose Get-Module -Name Azure -ListAvailable |
Uninstall-Module -Force -Verbose
```

```
Install-Module -Name AzureRm.BootStrapper -RequiredVersion
0.5.0 -Force Install-Module -Name AzureStack -RequiredVersion
1.6.0 ```
```

- From the **Administrator: Windows PowerShell** prompt, run the following to register your Azure Stack Hub operator PowerShell environment:

```
powershell Add-AzureRmEnvironment -Name 'AzureStackAdmin'
-ArmEndpoint
'https://adminmanagement.local.azurestack.external' ` -
AzureKeyVaultDnsSuffix
adminvault.local.azurestack.external ` -
AzureKeyVaultServiceEndpointResourceId
https://adminvault.local.azurestack.external
```

- From the **Administrator: Windows PowerShell** prompt, run the following to set the current environment:

```
powershell Set-AzureRmEnvironment -Name 'AzureStackAdmin'
```

- From the **Administrator: Windows PowerShell** prompt, run the following to authenticate to the current environment (when

prompted, sign in as the **CloudAdmin@azurestack.local** user with the **Pa55w.rd1234** as its password):

```
powershell Connect-AzureRmAccount -EnvironmentName  
'AzureStackAdmin'
```

- From the **Administrator: Windows PowerShell** prompt, run the following to verify that the authentication was successful and the corresponding context is set:

```
powershell Get-AzureRmContext
```

- From the **Administrator: Windows PowerShell** prompt, run the following to set the variables necessary to install the SQL Server Resource Provider:

```
```powershell $domain = 'azurestack.local' $privilegedEndpoint =  
'AzS-ERCS01' $downloadDir = 'C:\Downloads\SQLRP'
```



# Set the AzureStack\AzureStackAdmin credentials

```
$serviceAdmin = 'AzureStackAdmin@azurestack.local'  
$serviceAdminPass = ConvertTo-SecureString 'Pa55w.rd1234' -  
AsPlainText -Force $serviceAdminCreds = New-Object  
System.Management.Automation.PSCredential ($serviceAdmin,  
$serviceAdminPass)
```

# Set the AzureStack\CloudAdmin credentials

```
$cloudAdminName = 'AzureStack\CloudAdmin' $cloudAdminPass  
= ConvertTo-SecureString 'Pa55w.rd1234' -AsPlainText -Force  
$cloudAdminCreds = New-Object  
PSCredential($cloudAdminName, $cloudAdminPass)
```

# Set credentials for the new resource provider VM local admin account

```
$vmLocalAdminPass = ConvertTo-SecureString 'Pa55w.rd1234' -  
AsPlainText -Force $vmLocalAdminCreds = New-Object  
System.Management.Automation.PSCredential ('sqlrpadmin',  
$vmLocalAdminPass)
```

**Set a password that will protect  
the private key of a self-signed  
certificate generated to secure  
the SQL Server resource  
provider**

```
$pfxPass = ConvertTo-SecureString 'Pa55w.rd1234pfx' -  
AsPlainText -Force
```

# Update the PowerShell module path environment variable to include the SQL Server resource provider modules

```
$rpModulePath = Join-Path -Path $env:ProgramFiles -ChildPath  
'SqlMySqlPsh' $env:PSModulePath = $env:PSModulePath + ';' +  
$rpModulePath ``
```

- From the **Administrator: Windows PowerShell** prompt, change the current directory to the location of the extracted SQL Server resource provider installation files and run the DeploySQLProvider.ps1 script:

```
``powershell Set-Location -Path 'C:\Downloads\SQLRP'
```

```
.\DeploySQLProvider.ps1 -AzCredential $serviceAdminCreds -  
VMLocalCredential $vmLocalAdminCreds -  
CloudAdminCredential $cloudAdminCreds -PrivilegedEndpoint  
$privilegedEndpoint -DefaultSSLCertificatePassword $pfxPass ``
```

**Note:** Wait for the installation to complete. This might take about an hour.

## Task 3: Verify installation of the SQL Server resource provider

In this task, you will:

- Verify installation of the SQL Server resource provider
- Within the Remote Desktop session to **AzS-HOST1**, switch to the web browser window displaying the Azure Stack administrator portal and, in the hub menu, click **Resource groups**.
- On the **Resource groups** blade, click **system.local.sqladapter**.

- On the **system.local.sqladapter** blade, review the **Deployments** entry and verify that all deployments were successful.
- In the Azure Stack administrator portal, navigate to the **Virtual machines** and verify that the SQL resource provider VM was successfully created and is running.
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub tenant portal](#) and, if prompted, sign in as CloudAdmin@azurestack.local.
- In the hub menu of the Azure Stack Hub tenant portal, click **Create a resource**.
- On the **New** blade, select **Data + Storage** and verify that **SQL Database** appears in the list of available resource types.

**Review:** In this exercise, you have installed the SQL Server resource provider in Azure Stack Hub

## Exercise 2: Configure SQL Server resource provider in Azure Stack Hub

In this exercise, you will configure SQL Server resource provider in Azure Stack Hub.

1. Create a plan, offer, and subscription for a SQL Server hosting server (as a cloud operator)
2. Deploy an Azure Stack Hub VM that will become a SQL Server hosting server (as a cloud operator)
3. Add a SQL hosting server (as a cloud operator)
4. Make SQL databases available to users (as a cloud operator)
5. Create a SQL database (as a user)

**Note:** To minimize the duration of this exercise, some of the tasks that facilitate configuration of the Azure Stack Hub SQL Server resource provider have already been completed, including the following:

- Downloading a SQL Server image from Azure Marketplace
- Downloading **Sql IaaS VM extension** from Azure Marketplace

### Task 1: Create a plan, offer, and subscription for a SQL Server hosting server (as a cloud operator)

In this task, you will:

- Create a plan, offer, and subscription for SQL Server hosting server (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **sql-server-hosting-plan1**
  - Resource name: **sql-server-hosting-plan1**
  - Resource group: the name of a new resource group **sql-server-hosting-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Compute**, **Microsoft.Storage**, and **Microsoft.Network** checkboxes.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, specify the following settings:
  - Microsoft.Compute: **Default Quota**
  - Microsoft.Network: **Default Quota**
  - Microsoft.Storage: **Default Quota**
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, back on the **New** blade, click **Offer**.

- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **sql-server-hosting-offer1**
  - Resource name: **sql-server-hosting-offer1**
  - Resource group: **sql-server-hosting-offers-RG**
  - Make this offer public: **No**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **sql-server-hosting-plan1** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, back on the **New** blade, click **Subscription**.
- On the **Create a user subscription** blade, specify the following settings and click **Create**.
  - Name: **sql-server-hosting-subscription1**
  - User: **cloudadmin@azurestack.local**
  - Directory tenant: **ADFS.azurestack.local**
  - Offer name: **sql-server-hosting-offer1**

**Note:** Wait for the deployment to complete. This should take just a few seconds.

- Leave the Azure Stack Hub administrator portal window open.

## **Task 2: Deploy an Azure Stack Hub VM that will become a SQL Server hosting server (as a cloud operator)**

In this task, you will:



- Deploy an Azure Stack Hub VM that will become a SQL Server hosting server (as a cloud operator)

**Note:** An Azure Stack Hub VM operating as a SQL Server hosting server should be created in a billable user subscription.

- Within the Remote Desktop session to **AzS-HOST1**, switch to the web browser window displaying the [Azure Stack Hub tenant portal](#).
- In the hub menu of the Azure Stack Hub tenant portal, click **Create a resource**.
- On the **New** blade, select **Compute** and, in the list of available resource types, select **{WS-BYOL} Free SQL Server License: SQL Server 2017 Express on Windows Server 2016**.
- On the **Basics** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - Name: **sql-host-vm0**
  - VM disk type: **Premium SSD**
  - User name: **sqladmin**
  - Password: **Pa55w.rd**
  - Subscription: **sql-server-hosting-subscription1**
  - Resource group: the name of a new resource group **sql-server-hosting-RG**
  - Location: **local**
- On the **Choose a size** blade, select **DS1\_v2** and click **Select**.
- On the **Settings** pane of the **Create virtual machine** blade, set **Network Security Group** settings to **Advanced** and then click **Network security group (firewall)**.
- On the **Create network security group** blade, click + **Add an inbound rule**.
- On the **Add inbound security rule** blade, specify the following settings and click **Add** (leave others with their default values):
  - Destination port ranges: **1433**
  - Protocol: **TCP**
  - Action: **Allow**
  - Name: **SQL**

- Back on the **Create network security group** blade, click **OK**.
- Back on the **Settings** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - Boot diagnostics: **Disabled**
  - Guest OS diagnostics: **Disabled**
- On the **SQL Server settings** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - SQL connectivity: **Public (Internet)**
  - Port: **1433**
  - SQL Authentication: **Enable**
  - Login name: **SQLAdmin**
  - Password: **Pa55w.rd**
  - Storage configuration: **General**
  - Automated patching: **Disable**
  - Automated backup: **Disable**
  - Azure Key Vault integration: **Disable**
- On the **Summary** pane of the **Create virtual machine** blade, click **OK**.

**Note:** Wait for deployment to complete. This might take about 20 minutes.

- Once the deployment completes, navigate to the **sql-host-vm0** virtual machine blade and, in the **Overview** section, directly under the **DNS name** label, click **Configure**.
- On the **sql-host-vm0-ip | Configuration** blade, in the **DNS name label (optional)** text box, type **sql-host-vm0** and click **Save**.

**Note:** This makes the **sql-host-vm0** available via **sql-host-vm0.local.cloudapp.azurestack.external** DNS name.

- On the **sql-host-vm0-ip | Configuration** blade, set the **Assignment** option to **Static** and click **Save**.

**Note:** This will trigger a restart of the **sql-host-vm0** virtual machine.

### Task 3: Add a SQL hosting server (as a cloud operator)

In this task, you will:

- Add a SQL hosting server (as a cloud operator)
- Within the Remote Desktop session to **AzSHOST-1**, in the the web browser displaying the Azure Stack administrator portal, click **All services** and, in the **Administrative resources** section, click **SQL Hosting Servers**.

**Note:** You might need to refresh the browser page displaying the Azure Stack administrator portal for the **SQL Hosting Servers** resource type to appear.

- On the **SQL Hosting Servers** blade, click **+ Add**.
- On the **Add a SQL Hosting Server** blade, specify the following settings:
  - SQL Server Name: **sql-host-vm0.local.cloudapp.azurestack.external**
  - Username: **sqladmin**
  - Password: **Pa55w.rd**
  - Size of Hosting Server in GB: **50**
  - Always On Availability Group: unchecked
  - Subscription: **Default Provider Subscription**
  - Resource group: the name of a new resource group **sql.resources-RG**
  - Location: **local**
- On the **Add a SQL Hosting Server** blade, click **SKU**, on the **SKUs** blade, click **Create new SKU** and, on the **Create SKU** blade, specify the following settings:
  - Name: **MSSQL2017Exp**
  - Family: **SQL Server 2017**
  - Tier: **Standalone**
  - Edition: **Express**

- On the **Create SKU** blade, click **OK** and back on the **Add a SQL Hosting Server** blade, click **Create**.

**Note:** Wait for the operation to complete. This should take less than 1 minute.

- On the **SQL Hosting Servers** blade, click **Refresh** and verify that the **sqlhost1.local.cloudapp.azurestack.external** appears on the list of servers.

#### **Task 4: Make SQL databases available to users (as a cloud operator)**

In this task, you will:

- Make SQL databases available to users (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **sql-server-2017-express-db-plan1**
  - Resource name: **sql-server-2017-express-db-plan1**
  - Resource group: the name of a new resource group **sqldb-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.SQLAdapter** checkbox.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, next to the **Microsoft.SQLAdapter** entry, click **Create New**.
- On the **Create Quota\*** blade, specify the following settings and click **Create\*\***:
  - Quota Name: **sql-server-2017-express-db-quota1**

- Maximum size of all Databases (GB): **2**
- Maximum number of databases: **20**
- Click **Review + create** and then click **Create**.
 

**Note:** Wait for the deployment to complete. This should take just a few seconds.
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, back on the **New** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **sql-server-2017-express-db-offer1**
  - Resource name: **sql-server-2017-express-db-offer1**
  - Resource group: **sqlldb-offers-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans** >.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **sql-server-2017-express-db-plan1** entry.
- Click **Next: Add-on plans** >.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### **Task 5: Create a SQL database (as a user)**

In this task, you will:

- Create a test user account
- Create a SQL database by using the newly created user account
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the

list of administrative tools, double-click **Active Directory Administrative Center**.

- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, on the Dashboard, click the **Get a subscription** tile.
- On the **Get a subscription** blade, in the **Name** text box, type **t1u1-sqlldb-subscription1**.
- In the list of offers, select **sql-server-2017-express-db-offer1** and click **Create**.
- When presented with the message **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.
- In the Azure Stack Hub tenant portal, in the hub menu, click **All services**.
- In the list of services, click **SQL Databases**.
- On the **SQL Databases** blade, click **+ Add**.
- On the **Create Database** blade, specify the following settings:
  - Database Name: **sqlldb1**
  - Collation: **SQL\_Latin1\_General\_CP1\_CI\_AS**

- Max Size in MB: **200**
- Subscription: **\*\*t1u1-sqlldb-subscription1**
- Resource Group: the name of a new resource group **sqlldb-RG**
- Location: **local**
- SKU: **MSSQL2017Exp**

**Note:** You might have to wait before a newly created SKU becomes available in the tenant portal.

- On the **Create Database** blade, click **Login**.
- On the **Select a Login** blade, click **Create a new login**.
- On the **New Login** blade, specify the following settings and click **OK**:
  - Database login: **dbAdmin**
  - Password: **Pa55w.rd**
- Back on the **Create Database** blade, click **Create**.

**Note:** Wait for the deployment to complete. This should take less than a minute.

**>Review: In this exercise, you have added a SQL Server hosting server to Azure Stack Hub, made it available to tenants, and deployed a SQL database as a tenant user.**

lab: title: 'Lab: Implement App Service Resource Provider in Azure Stack Hub' module: 'Module 2: Provide Services'

---



# **Lab - Implement App Service Resource Provider in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- Implement SQL Server Resource Provider in Azure Stack Hub

# Estimated Time

4 hours

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to allow your tenants to deploy App Service apps and Azure functions.

# Objectives

After completing this lab, you will be able to:

- Implement App Service resource provider in Azure Stack Hub.

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment has the following configuration:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab.

## Exercise 1: Install the App Service resource provider in Azure Stack Hub

In this exercise, you will install the App Service resource provider in Azure Stack Hub.

1. Provision a SQL Server hosting server
2. Provision a file server
3. Install the App Service resource provider
4. Validate the installation of the App Service resource provider

**Note:** To minimize the duration of this exercise, some of the tasks necessary to install the Azure Stack Hub App Service resource provider have already been completed, including the following:

- Implementing Azure Marketplace syndication
- Downloading of the following Azure Marketplace items:
- **[smalldisk] Windows Server 2019 Datacenter Server Core-Bring your own license**
- **Windows Server 2016 Datacenter-Bring your own license**
- **Custom Script Extension**

### **Task 1: Provision a SQL Server hosting server**

In this task, you will:

- Provision a SQL Server hosting server
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the hub menu of the Azure Stack Hub administrator portal, click **Create a resource**.
- On the **New** blade, select **Compute** and, in the list of available resource types, select **{WS-BYOL} Free SQL Server License: SQL Server 2017 Express on Windows Server 2016**.
- On the **Basics** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - Name: **SqlHOST1**
  - VM disk type: **Premium SSD**

- User name: **sqladmin**
  - Password: **Pa55w.rd**
  - Subscription: **Default Provider Subscription**
  - Resource group: the name of a new resource group  
**sql.resources-RG**
  - Location: **local**
- On the **Choose a size** blade, select **DS1\_v2** and click **Select**.
- On the **Settings** pane of the **Create virtual machine** blade, set **Network Security Group** settings to **Advanced** and then click **Network security group (firewall)**.
- On the **Create network security group** blade, click + **Add an inbound rule**.
- On the **Add inbound security rule** blade, specify the following settings and click **Add** (leave others with their default values):
  - Destination port ranges: **1433**
  - Protocol: **TCP**
  - Action: **Allow**
  - Priority: **200**
  - Name: **custom-allow-sql**
- Back on the **Create network security group** blade, click **OK**.
- Back on the **Settings** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - Boot diagnostics: **Disabled**
  - Guest OS diagnostics: **Disabled**
- On the **SQL Server settings** pane of the **Create virtual machine** blade, specify the following settings and click **OK** (leave others with their default values):
  - SQL connectivity: **Public (Internet)**
  - Port: **1433**
  - SQL Authentication: **Enable**
  - Login name: **SQLAdmin**
  - Password: **Pa55w.rd**
  - Storage configuration: **General**

- Automated patching: **Disable**
  - Automated backup: **Disable**
  - Azure Key Vault integration: **Disable**
- On the **Summary** pane of the **Create virtual machine** blade, click **OK**.

**Note:** Wait for deployment to complete. This might take about 20 minutes.

- Once the deployment completes, navigate to the **SqlHOST1** virtual machine blade and, in the **Overview** section, directly under the **DNS name** label, click **Configure**.
- On the **SqlHOST1-ip | Configuration** blade, in the **DNS name label (optional)** text box, type **sqlhost1** and click **Save**.

**Note:** This makes the **sqlhost1** available via **sqlhost1.local.cloudapp.azurestack.external** DNS name.

- On the **sqlhost1-ip | Configuration** blade, set the **Assignment** option to **Static** and click **Save**.

**Note:** This will trigger a restart of the **sqlhost1** virtual machine. Wait until the restart completes before you proceed to the next step.

- Within the Remote Desktop session to **AzSHOST-1**, start a Remote Desktop session to **sqlhost1.local.cloudapp.azurestack.external** and, when prompted, sign in using the following credentials:
  - Username: **SQLAdmin**
  - Password: **Pa55w.rd**
- Within the Remote Desktop session to **SqlHOST1**, right-click **Start** and, in the right-click menu, select **Command Prompt (Admin)**.
- Within the Remote Desktop session to **SqlHOST1**, from the **Administrator: Command Prompt**, run the following to start a SQLCMD session to the local SQL Server instance:

```
sqlcmd
```



- Within the Remote Desktop session to **SqlHOST1**, from the **Administrator: Command Prompt**, run the following to enable contained database authentication for SQL server:

```
sp_configure 'contained database authentication', 1; GO
RECONFIGURE; GO
```

**Note:** This is necessary in order to use this hosting server when implementing App Service Resource Provider later in this lab.

**Note:** Leave the Remote Desktop session to **sqlhost1.local.cloudapp.azurestack.external** open. You will use it later in this lab.

## Task 2: Provision a file server

In this task, you will:

- Provision a file server
- Switch to the Remote Desktop session to **AzSHOST-1** and, in the Azure Stack administrator portal, in the hub menu, click **All services**.
- In the list of services, click **Marketplace management**
- On the **Marketplace management - Marketplace items** blade, search for the [smalldisk] **Windows Server 2019 Datacenter Server Core-Bring your own license** item and ensure that it is available.
- Within the Remote Desktop session to **AzSHOST-1**, open a new tab in a browser window and navigate to (<https://aka.ms/appsvconmasdkfstemplate>).
- On the **AzureStack-QuickStart-Templates / appservice-fileservice-standalone** page, click **azuredeploy.json** and then click **Raw**.
- Select the entire content of the page and copy it to Clipboard.
- Switch back to the Azure Stack administrator portal and click **+ Create a resource**.
- On the **New** blade, click **Custom** and then click **Template deployment**.

- On the **Custom deployment** blade, select **Build your own template in the editor**.
- On the **Edit template** blade, replace the precreated template with the content of Clipboard.
- On the **Custom Deployment** blade, click **Edit template**.
- On the **Edit template** blade, in the **Parameters** section, set the following values:
  - **defaultValue** of **imageReference**: set to **MicrosoftWindowsServer | WindowsServer | 2019-Datacenter-Core-smalldisk | latest**
  - **allowedValues** of **imageReference**: set to **MicrosoftWindowsServer | WindowsServer | 2019-Datacenter-Core-smalldisk | latest**
  - **defaultValue** of **fileServerVirtualMachineSize**: set to **Standard\_A1\_v2**
  - **allowedValues** of **fileServerVirtualMachineSize**: set to **Standard\_A1\_v2**
  - **defaultValue** of **adminPassword**: set to **Pa55w.rd1234**
  - **defaultValue** of **fileShareOwnerPassword**: set to **Pa55w.rd1234**
  - **defaultValue** of **fileShareUserPassword**: set to **Pa55w.rd1234**

**Note:** This will result in the following content of the **Parameters** section:

```
json "parameters": { "fileServerVirtualMachineSize": {
  "type": "string", "defaultValue": "Standard_A1_v2",
  "allowedValues": [ "Standard_A1_v2", ], "metadata": {
    "description": "Size of vm" } }, "imageReference": {
  "type": "string", "defaultValue": "MicrosoftWindowsServer
| WindowsServer | 2019-Datacenter-Core-smalldisk |
latest", "allowedValues": [ "MicrosoftWindowsServer |
WindowsServer | 2019-Datacenter-Core-smalldisk | latest"
], "metadata": { "description": "Please ensure the image
is available. publisher: MicrosoftWindowsServer | offer:
WindowsServer | sku: 2016-Datacenter" } },
"dnsNameForPublicIP": { "type": "string", "defaultValue":
"appservicefileshare", "maxLength": 63, "metadata": {
  "description": "Unique DNS Name for the Public IP used to
access the file share.It must be lowercase. It should
match the following regular expression, or it will raise
an error: ^[a-z][a-z0-9-]{1,61}[a-z0-9]$" } } },
```

```
"adminUsername": { "type": "string", "defaultValue":
"fileshareowner", "metadata": { "description": "File
server Admin user" } }, "adminPassword": { "type":
"securestring", "defaultValue": "Pa55w.rd1234",
"metadata": { "description": "File server Admin password"
} }, "fileShareOwner": { "type": "string",
"defaultValue": "fileshareowner", "metadata": {
"description": "fileshare owner username" } },
"fileShareOwnerPassword": { "type": "securestring",
"defaultValue": "Pa55w.rd1234", "metadata": {
"description": "fileshare owner password" } },
"fileShareUser": { "type": "string", "defaultValue":
"fileshareuser", "metadata": { "description": "fileshare
user" } }, "fileShareUserPassword": { "type":
"securestring", "defaultValue": "Pa55w.rd1234",
"metadata": { "description": "fileshare user password" }
}, "vmExtensionScriptLocation": { "type": "string",
"defaultValue":
"https://raw.githubusercontent.com/Azure/azurestack-
quickstart-templates/master/appservice-fileserver-
standalone", "metadata": { "description": "File Server
extension script Url" } } },
```

- On the **Edit template** blade, in the **Variables** section, replace **"sku": "2016-Datcenter"**, with **"sku": "2019-Datcenter-Core-smalldisk"**, and select **Save**.
- Back on the **Custom Deployment** blade, in the **Subscription** drop-down list, select **Default Provider Subscription** and, in the **Resource group** section, select **sql.resources-RG**.
- On the **Custom deployment** blade, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take about 15 minutes.

### Task 3: Install the App Service resource provider

In this task, you will:

- Install the App Service resource provider
- Within the Remote Desktop session to **AzSHOST-1**, in the Azure Stack administrator portal, in the hub menu, click **All services**.

- In the list of services, click **Marketplace management**.
- On the **Marketplace management - Marketplace items** blade, search for the **Windows Server 2016 Datacenter-Bring your own license** item and ensure that it is available.
- On the **Marketplace management - Marketplace items** blade, search for the **Custom Script Extension** item and ensure that it is available.
- Within the Remote Desktop session to **AzSHOST-1**, start the web browser and navigate to (<https://aka.ms/appsvconmasinstaller>) to download **AppService.exe** and, once the download completes, copy the file to the **C:\Downloads\AppServiceRP** folder (create the folder if needed).
- In the web browser, navigate to (<https://aka.ms/appsvconmashelpers>) to download **AppServiceHelperScripts.zip** and, once the download completes, extract its content to the **C:\Downloads\AppServiceRP** folder.
- Within the Remote Desktop session to **AzSHOST-1**, start Windows PowerShell as administrator.

- From the **Administrator: Windows PowerShell** window, run the following to create certificates required by App Service on Azure Stack:

```
```powershell Set-Location -Path C:\Downloads\AppServiceRP
Get-ChildItem -Path '.' -File -Recurse | Unblock-File
```

```
$pfxPass = ConvertTo-SecureString 'Pa55w.rd1234pfx' -
AsPlainText -Force
```

```
.\Create-AppServiceCerts.ps1 -pfxPassword $pfxPass -
DomainName 'local.azurestack.external' ```
```

- From the **Administrator: Windows PowerShell** window, run the following to create an Azure Resource Manager root certificate for the Azure Stack Hub required to install the App Service provider:

```
``` $domain = 'azurestack.local' $privilegedEndpoint = 'AzS-
ERCS01'
```

# Add the cloudadmin credential that's required for privileged endpoint access.

```
$cloudAdminPass = ConvertTo-SecureString 'Pa55w.rd1234' -
AsPlainText -Force $cloudAdminCreds = New-Object
System.Management.Automation.PSCredential
("CloudAdmin@$domain", $cloudAdminPass)

.\Get-AzureStackRootCert.ps1 -PrivilegedEndpoint
$privilegedEndpoint -CloudAdminCredential $cloudAdminCreds
```
```

**Note:** The script creates in the local folder a file named AzureStackCertificationAuthority.cer, containing the Azure Resource Manager root certificate for the Azure Stack Hub.

- From the **Administrator: Windows PowerShell** window, run the following to create an AD FS app required to install the App Service provider:

```
``` $domain = 'azurestack.local' $privilegedEndpoint = 'AzS-
ERCS01' $adminArmEndpoint =
'adminmanagement.local.azurestack.external' $certificateFilePath =
'C:\Downloads\AppServiceRP\asso.appservice.local.azurestack.exter
nal.pfx' $certificatePassword = ConvertTo-SecureString
'Pa55w.rd1234pfx' -AsPlainText -Force
```

```
.\Create-ADFSIdentityApp.ps1 -AdminArmEndpoint
$adminArmEndpoint -PrivilegedEndpoint $privilegedEndpoint -
CloudAdminCredential $cloudAdminCreds -CertificateFilePath
$certificateFilePath -CertificatePassword $certificatePassword ```
```

- In the output of the script, copy the GUID representing the ID of the generated AD FS application.

**Note:** Make sure to record this GUID. You will need it later in this task.

- From the **Administrator: Windows PowerShell** window and, from the **Administrator: Windows PowerShell** window, run the following to start AppService.exe:

```
.\AppService.exe
```

**Note:** This will start the Microsoft Azure App Service Setup wizard.

- Click **Deploy App Service or upgrade to the latest version**.
- On the **MICROSOFT SOFTWARE SUPPLEMENTARY LICENSE TERMS** page, review the content, click the checkbox **I have read, understood, and agreed to these license terms** checkbox and click **Next**.
- On the page displaying the third party license terms, review the content, click the checkbox **I have read, understood, and agreed to these license terms** checkbox and click **Next**.
- On the page displaying the Admin and Tenant ARM endpoints, verify that the information is correct and click **Next**.
- On the Azure Stack App Service cloud information page, ensure that the **Credential** option is selected and click **Connect**.
- When prompted, sign in as **CloudAdmin@AzureStack.local** with the **Pa55w.rd1234** password.
- Back on the Azure Stack App Service cloud information page, in the **Azure Stack Subscriptions** drop-down list, select the **Default Provider Subscription**, in the **Azure Stack Locations** drop-down list, select **local** and click **Next**.
- On the **Virtual Network Configuration**, accept the default settings and click **Next**.
- On the next page, specify the following information and click **Next**:
  - File Share UNC Path:  
**\\appservicefileshare.local.cloudapp.azurestack.external\websites**
  - File Share Owner: **fileshareowner**
  - File Share Owner Password: **Pa55w.rd1234**
  - File Share User: **fileshareuser**
  - File Share User Password: **Pa55w.rd1234**

- On the next page, specify the settings that identify the application ID you generated earlier in this task and click **Next**:
  - Identity Application Id: the GUID you copied earlier in this task
  - Identity Application certificate file (.pfx):  
*C:\Downloads\AppServiceRP\sso.appservice.local.azurestack.external.pfx\**
  - Identity Application certificate file (.pfx) password:  
*Pa55w.rd1234pfx\**
  - Azure Resource Manager (ARM) root certificate file (.cer):  
*C:\Downloads\AppServiceRP\AzureStackCertificationAuthority.cer\**
- On the next page, specify the settings that identify the certificate files and their respective passwords:
  - App Service default SSL certificate file (.pfx):  
*C:\Downloads\AppServiceRP\_.appservice.local.azurestack.external.pfx\**
  - App Service default SSL certificate (.pfx) password:  
*Pa55w.rd1234pfx\**
  - App Service API SSL certificate file (.pfx):  
*C:\Downloads\AppServiceRP\api.appservice.local.azurestack.external.pfx\**
  - App Service API SSL certificate (.pfx) password:  
*Pa55w.rd1234pfx\**
  - App Service Publisher certificate file (.pfx):  
*C:\Downloads\AppServiceRP\ftp.appservice.local.azurestack.external.pfx\**
  - App Service Publisher SSL certificate (.pfx) password:  
*Pa55w.rd1234pfx\**
- On the next page, specify the SQL Server settings:
  - SQL Server Name:  
**sqlhost1.local.cloudapp.azurestack.external**
  - SQL sysadmin login: **SQLAdmin**
  - SQL sysadmin password: **Pa55w.rd1234**
- On the next page, specify the number and SKU of instances of the App Service deployment:

- Controller Role: **2 instances - Standard\_A1\_v2 - [1 Core(s), 2048 MB]**
- Management Role: **1 instance - Standard\_A2\_v2 - [2 Core(s), 4096 MB]**
- Publisher Role: **1 instance - Standard\_A1\_v2 - [1 Core(s), 2048 MB]**
- FrontEnd Role: **1 instance - Standard\_A1\_v2 - [1 Core(s), 2048 MB]**
- Shared Worker Role: **1 instance - Standard\_A1\_v2 - [1 Core(s), 2048 MB]**
- click **Next**.
- On the next page, in the **Select Platform Image** drop-down list, select the **2016 Datacenter - latest** image and click **Next**.
- On the next page, specify the following Admin credentials for the deployment:
  - Worker Role Virtual Machine(s) Admin: **SAWorkerAdmin**
  - Worker Role Virtual Machine(s) Password: **Pa55w.rd1234**
  - Confirm Password: **Pa55w.rd1234**
  - Other Roles Virtual Machine(s) Admin: **SAORoleAdmin**
  - Other Roles Virtual Machine(s) Password: **Pa55w.rd1234**
  - Confirm Password: **Pa55w.rd1234**
- Click **Next**.
- On the Summary page, click the checkbox **Select and click next to start the deployment** and, to start the deployment, click **Next**.

**Note:** Wait for the installation to complete. This might take 2-3 hours.

- Once the installation completes, click **Exit**.

#### **Task 4: Validate the installation of the App Service resource provider**

In this task, you will:

- Validate the installation of the App Service resource provider



- Within the Remote Desktop session to **AzSHOST-1**, in the web browser displaying the Azure Stack administrator portal, in the hub menu, select **All services**, on the **All services** blade, select **Administration**, and, in the list of services, click **App Service**.

**Note:** You might need to refresh the browser page for the **App Service** entry to become available.

- On the **App Service** blade, in the **Essentials** section, verify that the **All roles are ready** message appears under the **Status** label.

**Note:** Wait until all roles are successfully started. This might take additional 15-20 minutes.

**Review:** In this exercise, you have installed the App Service resource provider on Azure Stack Hub.

## **Exercise 2: Explore management tasks of App Service resource provider on Azure Stack Hub**

In this exercise, you will explore management tasks of App Service resource provider on Azure Stack Hub.

1. Explore the scaling functionality of App Service resources
2. Explore backup settings of App Service resource provider

### **Task 1: Explore the scaling functionality of App Service resources**

In this task, you will:

- Review the scaling functionality of App Service resources
- Review the backup settings of App Service resource provider
- Within the Remote Desktop session to **AzSHOST-1**, in the web browser displaying the Azure Stack administrator portal, on the **App Service** blade, click **Roles**.
- On the **App Service | Roles** blade, review the list of roles and the corresponding instances.

- On the **App Service | Roles** blade, in the **Controller** row, click the ellipsis symbol on the right side and, in the drop-down list, note the **Virtual machine** entry.

**Note:** The Controller role is implemented by using virtual machines, which is the reason for the choice of 2 instances of the Controller during the installation of the App Service resource provider.

- On the **App Service | Roles** blade, in the remaining rows, click the ellipsis symbol on the right side and, in the drop-down list, note the **ScaleSet** entry.

**Note:** All other roles are implemented by using scale sets, which allows for scaling.

- On the **App Service | Roles** blade, note that you currently only have Web Worker roles in the **Shared** worker tier.
- On the **App Service | Roles** blade, in the vertical menu on the left, select **Worker Tiers**.
- On the **App Service | Worker Tiers** blade, click **+ Add**.
- On the **Create** blade, review the available options, including the **Compute Mode** drop-down list that allows you to choose between **Shared** and **Dedicated**.

**Note:** You have the ability to deploy virtual machines in a range of sizes with custom software as virtual machines in the worker tier of your choice.

- Close the **Create** blade without making any changes.

**Note:** The provisioning process might take over an hour.

## **Task 2: Explore the backup settings of the App Service resource provider**

In this task, you will:

- Review backup settings of the App Service resource provider.

**Note:** App Service backup on Azure Stack Hub consists of the following main components

- The resource provider infrastructure
- The resource provider secrets
- The SQL Server instance hosting the metering database
- The user workload content stored on the App Service file share

**Note:** The resource provider infrastructure configuration can be recreated from backup during recovery using App Service recovery PowerShell cmdlets. For details regarding the recovery process, refer to [App Service recovery on Azure Stack Hub](#).

- Within the Remote Desktop session to **AzSHOST-1**, in the web browser displaying the Azure Stack administrator portal, on the **App Service** blade, click **Secrets**.
- On the **App Service | Secrets** blade, click **Download Secrets** and then click **Save**.
- Verify that the **SystemSecrets.json** file was downloaded to the **Downloads** folder on **AzSHOST-1**.

**Note:** You should copy the **SystemSecrets.json** file to a safe location and repeat this process whenever the secrets are rotated.

**Note:** The recommended approach to back up the **Appservice\_hosting** and **Appservice\_metering** databases involves the use of SQL Server maintenance plans of Azure Backup Server, but you can also back them up by using the SQL Server PowerShell module cmdlets.

- Within the Remote Desktop session to **AzSHOST-1**, switch to the Remote Desktop session to **sqlhost1.local.cloudapp.azurestack.external**.
- Within the Remote Desktop session to **SqlHOST1**, start Windows PowerShell as Administrator.

- Within the Remote Desktop session to **SqlHOST1**, from the **Administrator: Windows PowerShell** prompt, run the following to perform a local backup of the App Service databases:

```
powershell $date = Get-Date -Format 'yyyyMMdd' New-Item -
ItemType Directory -Path 'C:\Backups' Backup-SqlDatabase
-ServerInstance 'localhost' -Database
'appservice_hosting' -BackupFile
"C:\Backups\appservice_hosting_$date.bak" -CopyOnly
Backup-SqlDatabase -ServerInstance 'localhost' -Database
'appservice_metering' -BackupFile
"C:\Backups\appservice_metering_$date.bak" -CopyOnly
```

**Note:** App Service stores tenant app information on its designated file share. The recommended approach to back up the file share involves the use of Azure Backup Server, but you can use for this purpose any file copy utility.

- Switch to the Remote Desktop session to **AzSHOST-1** and, within the Remote Desktop session to **AzSHOST-1**, in the web browser displaying the Azure Stack administrator portal, on the **App Service** blade, click **System configuration**.
- In the web browser displaying the Azure Stack administrator portal, on the **App Service | System configuration** blade, note the full path of the **File share**  
(\appservicefileshare.local.cloudapp.azurestack.external\websites)
- Switch to the Remote Desktop session to **AzSHOST-1** and from the **Administrator: Windows PowerShell** window, run the following to copy the content of the App Service file share to the local file system:

```
```powershell $source =
'appservicefileshare.local.cloudapp.azurestack.external\websites'
$date = Get-Date -Format 'yyyyMMdd' $destination =
"C:\Backups\FileShare\$date" New-Item -ItemType Directory -Path
$destination -Force
```

```
$fileshareusername = 'fileshareowner' $fileshareuserpassword =
ConvertTo-SecureString 'Pa55w.rd1234' -AsPlainText -Force
$fileshareusercreds = New-Object
System.Management.Automation.PSCredential
```

```
($fileshareusername, $fileshareuserpassword) New-PSDrive -Name  
'S' -Root $source -PSProvider 'FileSystem' -Credential  
$fileshareusercreds robocopy $source $destination /e /r:1 /w:1  
Remove-PSDrive -Name 'S' ``
```

**Review:** In this exercise, you have explored management tasks of App Service resource provider on Azure Stack Hub.

### Exercise 3: Deliver App Service resources in Azure Stack Hub

In this exercise, you will deliver App Service resources to Azure Stack Hub users.

1. Make App Service resources available to users (as a cloud operator)
2. Create a Web app (as a user)

#### Task 1: Make App Service resources available to users (as a cloud operator)

In this task, you will:

- Make App Service resources available to users (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, switch to the browser window displaying the [Azure Stack Hub administrator portal](#).
- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans** and then click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **app-service-plan1**
  - Resource name: **app-service-plan1**
  - Resource group: the name of a new resource group **app-service-plans-RG**
- Click **Next: Services >**.

- On the **Services** tab of the **New plan** blade, select the **Microsoft.Web** checkbox.
- Click **Next: Quotas>**.
- On the **Quotas** tab of the **New plan** blade, select **Create New** and on the **Create** blade, specify the following settings, and click **OK**:
  - Name: **app-service-quota1**
  - App Service Plans: **Custom 20**
  - Shared App Service Plans: **Custom 10**
  - Dedicated App Service Plans: **Custom 10**
  - Pricing SKUs: **Custom 2 selected (Free and Shared)**
  - Consumption Plan: **Enabled**

**Note:** To offer Azure Functions in the consumption plan model, you need to deploy shared web workers.

- Back on the **Quotas** tab of the **New plan** blade, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, back on the **New** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **app-service-offer1**
  - Resource name: **app-service-offer1**
  - Resource group: **app-service-offers-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **app-service-plan1** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

## Task 2: Create a Web app (as a user)

In this task, you will:

- Create a test user account
- Create a Web app (as a user)
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, on the Dashboard, click the **Get a subscription** tile.
- On the **Get a subscription** blade, in the **Name** text box, type **t1u1-app-service-subscription1**.
- In the list of offers, select **app-service-offer1** and click **Create**.

- When presented with the message **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.
- In the Azure Stack Hub tenant portal, in the hub menu, click **+ Create a resource**.
- In the list of services, click **Web + Mobile** and then click **Web App**.
- On the **Web App** blade, specify the following settings:
  - Subscription: **t1u1-app-service-subscription1**
  - App name: **t1u1webapp1**
  - Resource Group: the name of a new resource group **webapps-RG**
- On the **Web App** blade, click **App Service plan/Location** and, on the **App Service plan** blade, click **+ Create new**.
- On the **New App Service plan** blade, specify the following settings:
  - App Service plan: **appserviceplan1**
  - Location: **local**
- On the **New App Service plan** blade, click **Pricing tier**.
- On the **Spec Picker** blade, select the **F1** pricing tier and click **Apply**.
- Back on the **New App Service plan** blade, click **OK**.
- Back on the **Web App** blade, click **Create**.

**Note:** Wait for the deployment to complete. This should take less than a minute.

- Within the Remote Desktop session to **AzS-HOST1**, in the InPrivate session of the web browser displaying the Azure Stack Hub user portal, in the hub menu, select **All resources**.
- On the **All resources** blade, in the subscription filter drop down list, select the **t1u1-app-service-subscription1** entry and then click **Refresh**.



- On the **All resources** blade, in the list of resources, click the **t1u1webapp1** entry.
- On the **t1u1webapp1** blade, click **Browse**.

**Note:** This should open another browser tab displaying the default home page of the newly provisioned web app.

**>Review: In this exercise, you have made the App Service available to users and created a Web app as a tenant user.**

lab: title: 'Lab: Register Azure Stack Hub with an Azure Subscription'  
module: 'Module 3: Implement Data Center Integration'

---

# **Lab - Register Azure Stack Hub with an Azure Subscription**

## **Student lab manual**

### **Lab dependencies**

- None

# Estimated Time

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to register it with your Azure subscription in order to download Azure Marketplace items and set up data reporting to Azure.

# Objectives

After completing this lab, you will be able to:

- Register Azure Stack Hub with an Azure subscription

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment has the following configuration:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab.

## Exercise 1: Register Azure Stack Hub with an Azure subscription.

In this exercise, you will register Azure Stack Hub with an Azure subscription.

1. Register the Azure Stack Hub resource provider
2. Perform Azure Stack Hub registration
3. Verify Azure Stack Hub registration

## Task 1: Register the Azure Stack Hub resource provider

In this task, you will:

- Register the Azure Stack Hub resource provider in the target Azure subscription.
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as administrator.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to install the PowerShell Az module for Azure Stack Hub:

```
powershell [Net.ServicePointManager]::SecurityProtocol =  
[Net.SecurityProtocolType]::Tls12 Install-Module -Name  
Az.BootStrapper -Force -AllowPrerelease -AllowClobber  
Install-AzProfile -Profile 2019-03-01-hybrid -Force  
Install-Module -Name AzureStack -RequiredVersion 2.0.2-  
preview -AllowPrerelease
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to authenticate to the Azure subscription you will be using in this lab.

```
powershell Connect-AzAccount -EnvironmentName  
'AzureCloud'
```

- When prompted, sign in with the credentials of an Azure Active Directory (Azure AD) user with the Contributor role in the Azure subscription you will be using in this lab.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to register the Azure Stack resource provider:

```
powershell Register-AzResourceProvider -ProviderNamespace  
Microsoft.AzureStack
```



- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to determine whether that the registration is completed:

```
powershell Get-AzResourceProvider -ProviderNamespace
Microsoft.AzureStack | Where-Object {$_.RegistrationState
-eq 'Registered'}
```

**Note:** Wait for the registration to complete. Re-run the **Get-AzResourceProvider** cmdlet in order to verify the status of the registration.

## Task 2: Perform Azure Stack Hub registration

In this task, you will:

- Perform Azure Stack Hub registration.
- Within the Remote Desktop session to **AzSHOST-1**, from the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to establish a privileged endpoint (PEP) session:

```
powershell Enter-PSSession -ComputerName AzS-ERCS01 -
ConfigurationName PrivilegedEndpoint
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, within the PowerShell Remoting session to the privileged endpoint, run the following to display the the summary configuration of the Azure Stack Hub stamp:

```
powershell Get-AzureStackStampInformation
```

- In the output of the command you ran in the previous step, identify and record the value of the **CloudId** property.
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, within the PowerShell Remoting session to the privileged endpoint, run the following to exit the session:

```
powershell Exit-PSSession
```

**Note:** In general, you should not use the **Exit\_PSSession** to terminate the session to the privileged endpoint, but use the **Close-PrivilegedEndpoint** cmdlet instead. We are not following this practice for the sake of simplicity to avoid the need to set up a file share to host the session transcript logs.

**Note:** You can also identify the value of the stamp **Cloud ID** property from the **local | Properties** blade in the Azure Stack Hub administrator portal.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to register the Azure Stack PowerShell environment that targets your Azure Stack instance (make sure to replace the [cloud\_ID] placeholder with the value of the **CloudId** property you identified in the output of the **Get-AzureStackStampInformation** cmdlet):

```
powershell Import-Module  
. \Registration\RegisterWithAzure.psm1 $RegistrationName =  
"[cloud_ID]" Set-AzsRegistration ` -  
PrivilegedEndpointCredential $adminCredentials ` -  
PrivilegedEndpoint 'AzS-ERCS01' ` -BillingModel  
'Development' ` -RegistrationName $RegistrationName ` -  
UsageReportingEnabled:$true
```

- When prompted, sign in with an Azure AD user account with the Contributor role in the Azure subscription.

**Note:** Wait for the registration to complete. This registration might take about 20 minutes.

### Task 3: Verify Azure Stack Hub registration

In this task, you will:

- Verify registration of Azure Stack Hub with the Azure subscription
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser displaying the Azure Stack Hub administrator portal, from the **Dashboard** blade, select the **Region management**

tile.

- On the **local** blade, select **Properties**.
- On the **local | Properties** blade, verify that the **Registration status** is listed as **Registered**.

**Note:** The status can be **Registered**, **Not registered**, or **Expired**.

**>Review: In this exercise, you have registered Azure Stack Hub with an Azure subscription.**

lab: title: 'Lab: Delegate Offer Management in Azure Stack Hub'  
module: 'Module 4: Manage Identity and Access'

---

# **Lab - Delegate Offer Management in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

45 minutes

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to delegate management of offers to your internal technical support staff.

# Objectives

After completing this lab, you will be able to:

- Implement Azure Stack Hub offer delegation.



# Lab Environment

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will create additional user accounts in the course of this lab.

## Exercise 0: Prepare for the lab

In this exercise, you will create Active Directory user accounts that you will be using in this lab:

1. Create a delegated admin user account (as a cloud operator)
2. Create a tenant user account (as a cloud operator)

### Task 1: Create a delegated admin user account (as a cloud operator)

In this task, you will:

- Create a delegated admin user account (as a cloud operator)

- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **DP1**
  - User UPN logon: **dp1@azurestack.local**
  - User SamAccountName: **azurestack\dp1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**

## **Task 2: Create a tenant user account (as a cloud operator)**

In this task, you will:

- Create a tenant user account (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the **Active Directory Administrative Center** console, in the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**

- Password: **Pa55w.rd**
- Password options: **Other password options -> Password never expires**

**Review:** In this exercise, you have created the Active Directory accounts you will use in this lab.

## **Exercise 1: Designate the delegated provider (as a cloud operator)**

In this exercise, you will act as a cloud operator and create a plan consisting of the **Subscription** service and a corresponding offer. Next, you will create a subscription containing the new offer and designate the delegated provider as its subscriber:

1. Create a plan consisting of the Subscription service (as a cloud operator)
2. Create an offer based on the plan (as a cloud operator)
3. Create a new subscription containing the offer (as a delegated provider)

### **Task 1: Create a plan consisting of the Subscription service (as a cloud operator)**

In this task, you will:

- Create an plan consisting of the Subscription service (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:

- Display name: **co-subscription-plan**
  - Resource name: **co-subscription-plan**
  - Resource group: the name of a new resource group **co-subscription-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Subscriptions** checkbox.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, in the **Microsoft.Subscriptions** dropdown list, select **delegatedProviderQuota**.
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

## **Task 2: Create an offer based on the plan (as a cloud operator)**

In this task, you will:

- Create an offer based on the plan (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **cotodp-subscription-offer**
  - Resource name: **cotodp-subscription-offer**
  - Resource group: **co-subscription-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans >**.

- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **co-subscription-plan** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### **Task 3: Create a new subscription containing the offer and add the delegated provider as its subscriber (as a cloud operator)**

In this task, you will:

- Create a new subscription containing the offer and add the delegated provider as its subscriber (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Subscription**.
- On the **Create a user subscription** blade, , specify the following settings:
  - Name: **dp1-subscription1**
  - User: **dp1@azurestack.local**
  - Directory tenant: **ADFS.azurestack.local**
- In the list of public offers, click **cotodp-subscription-offer** and then click **Create**

**Note:** Wait for the deployment to complete. This should take just a few seconds.

**Review:** In this exercise, you have designated the delegated provider.

## Exercise 2: Create and delegate a delegated offer to a delegated provider (as a cloud operator)

In this exercise, you will act as a cloud operator and create an offer containing services, which the delegated provider will then offer to its tenants:

1. Create a plan to delegate (as a cloud operator)
2. Create an offer based on the plan (as a cloud operator)
3. Delegate the offer to the delegated provider (as a cloud operator)

### Task 1: Create a plan to delegate (as a cloud operator)

In this task, you will:

- Create a plan to delegate (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **co-services1-plan**
  - Resource name: **co-services1-plan**
  - Resource group: the name of a new resource group **co-services-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Storage** checkbox.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, next to the **Microsoft.Storage** dropdown list, click **Create New**.
- On the **Create Storage quota** blade, specify the following settings and click **OK**:

- Name: **co-services1-storage-quota**
- Maximum capacity (GB): **200**
- Total number of storage accounts: **1**
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

## Task 2: Create an offer based on the plan (as a cloud operator)

In this task, you will:

- Create an offer based on the plan (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **cotodp-services1-offer**
  - Resource name: **cotodp-services1-offer**
  - Resource group: **co-services-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **co-services1-plan** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### Task 3: Delegate the offer to a delegated provider (as a cloud operator)

In this task, you will:

- Delegate the offer to the delegated provider (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, in the hub menu, click **Offers**.
- On the **Offers** blade, click **cotodp-services1-offer**.
- On the **cotodp-services1-offer** blade, click **Delegated providers**.
- On the **cotodp-services1-offer | Delegated providers** blade, click **+ Add**.
- On the **Delegate offer** blade, specify the following and click **Delegate**:
  - Name: **cotodp-services1-offer**
  - Pick the delegated provider subscription: **dp1-subscription1**

**Note:** Wait for the deployment to complete. This should take just a few seconds.

**Review:** In this exercise, you have created and delegated the offer to a delegated provider.

### Exercise 3: Create a delegated offer to a tenant (as a delegated provider)

In this exercise, you will act as a delegated provider and create an offer (based on the offer from the cloud operator) to which your tenants will be able to subscribe:

1. Create a delegated provider offer for a tenant (as a cloud operator)
2. Identify the URL of the delegated provider portal (as a delegated provider)

### Task 1: Create a delegated provider offer for a tenant (as a delegated provider)



In this task, you will:

- Create an offer for a tenant (as a delegated provider) based on the cloud operator offer for the delegated provider

This will allow tenants to create a subscription based on the offer from the delegated provider

1. Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
2. In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **dp1@azurestack.local** with the password **Pa55w.rd**.
3. In the Azure Stack Hub user portal, in the hub menu, click **All services** and, on the **All services** blade, click **Offers**.
4. On the **Offers** blade, click **+ Add**.
5. On the **Create a new offer** blade, specify the following settings:
  - Display name: **dp1tot-services1-offer**
  - Resource name: **dp1tot-services1-offer**
  - Provider subscription: **dp1-subscription1**
  - Resource group: the name of a new resource group **dp1-RG**
6. Click **Delegated** offer.
7. On the **Delegated offer** blade, click **cotodp-services1-offer**.
8. Back on the **Create a new offer** blade, click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.
9. In the Azure Stack Hub user portal, back on **Offers** blade, click **Refresh** and click the newly created offer **dp1tot-services1-offer**.
10. On the **dp1tot-services1-offer** blade, click **Change state** and, in the drop-down list, click **Public**.

**Task 2: Identify the URL of the delegated provider portal (as a delegated provider)**

In this task, you will:

- Identify the URL of the delegated provider portal (as a delegated provider)
- In the Azure Stack Hub user portal, while signed in as **dp1@azurestack.local**, in the hub menu, click **All services** and, on the **All services** blade, in the list of services, click **Subscriptions**.
- On the **Subscriptions** blade, click **dp1-subscription1**.
- On the **dp1-subscription1** delegated provider subscription blade, click **Properties**.
- On the properties blade, copy the value of the **Portal URL** to clipboard. You will need it in the next exercise of this lab,
- In the Azure Stack Hub user portal, in the upper right corner, click the user avatar icon and, in the drop-down menu, click **Sign out**.

**Note:** Tenants need to subscribe to the offer from the delegated provider portal URL.

**Review:** In this exercise, you have created a delegated provider offer.

#### **Exercise 4: Sign up for a delegated provider's offer (as a tenant user)**

In this exercise, you will act as a tenant user who signs up for a delegated provider's offer and creates a resource in the new subscription. The exercise consists of the following tasks:

1. Sign up for the delegated provider's offer (as a tenant user)
2. Create a resource in the new subscription (as a tenant user)

##### **Task 1: Sign up for the delegated provider's offer (as a tenant user)**

In this task, you will:

- Sign up for the delegated offer (as a tenant user)

- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the delegated provider portal URL you identified in the previous exercise and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.

**Note:** Make sure not to use the standard Azure Stack Hub user portal URL in this case.

- In the Azure Stack Hub user portal, click the **Get a subscription** tile.
- On the **Get a subscription** blade, in the **Name** text box, type **t1u1-subscription1**.
- On the **Get a subscription** blade, select the **dp1tot-services1-offer** offer and click **Create**.
- When presented with the message **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.

## **Task 2: Create a resource in the new subscription (as a tenant user)**

In this task, you will:

- Create a resource in the new subscription (as a tenant user)
- Within the Remote Desktop session to **AzS-HOST1**, in the Azure Stack Hub user portal, while signed in as **t1u1@azurestack.local**, in the hub menu, click **All services** and, on the **All services** blade, in the list of services, click **Subscriptions**.
- On the **Subscriptions** blade, click **t1u1-subscription1**.
- On the **t1u1-subscription1** blade, click **Resources**.
- On the **t1u1-subscription1 | Resources** blade, click **+ Add**.
- On the **New** blade, click **Data + Storage** and, in the list of resource types, click **Storage account**.
- On the **Basics** tab of the **Create storage account** blade, specify the following settings:
  - Subscription: **t1u1-subscription1**

- Resource group: the name of a new resource group **delegated-RG**
  - Storage account name: any unique name consisting of between 3 and 24 lower case letters or digits
  - Location: **local**
  - Performance: **Standard**
  - Account kind: **Storage (general purpose v1)**
  - Replication: **Locally-redundant storage (LRS)**
- On the **Basics** tab of the **Create storage account** blade, click **Next: Advanced >**.
  - On the **Advanced** tab of the **Create storage account** blade, leave the default settings in place and click **Review + create**.
  - On the **Review + create** tab of the **Create storage account** blade, click **Create**.

**Note:** Wait until the storage account is provisioned. This should take about one minute.

- Verify that the storage account has been successfully created.

**>Review: In this exercise, you have subscribed to a delegated provider's offer, creating this way a new subscription and provisioned a storage account in the new subscription.**

lab: title: 'Lab: Manage Service Principals in Azure Stack Hub' module: 'Module 4: Manage Identity and Access'

---

# **Lab - Manage Service Principals in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You plan to use an internally develop application to manage Azure Stack Hub. To allow for the application to authenticate, you need to create a service principal and assign to it the Contributor role in the Default Provider Subscription.

**Note:** An application that needs to deploy or configure resources through Azure Resource Manager must be represented by its own identity. Just as a user is represented by a security principal called a user principal, an app is represented by a service principal. The service principal provides an identity for your app, allowing you to delegate only the necessary permissions to the app.

An app must present credentials during authentication. This authentication consists of two elements:

- An Application ID, sometimes referred to as a Client ID. A GUID that uniquely identifies the app's registration in your Active Directory tenant.
- A secret associated with the application ID. You can either generate a client secret string (equivalent to a password), or specify an X509 certificate.

In this lab, you will use a secret. For details regarding authentication by using certificates, refer to [Use an app identity to access Azure Stack Hub resources](#).



# Objectives

After completing this lab, you will be able to:

- Create and manage service principals in Azure Stack Hub AD FS integrated scenarios.

# Lab Environment

This lab uses an ASDK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

The approach described in this lab is specific to the AD FS integrated scenarios. For information regarding implementing service principal-based authentication when using Azure Active Directory (Azure AD) as the identity provider, refer to [Use an app identity to access Azure Stack Hub resources](#).

## Exercise 1: Create and configure a service principal in Azure Stack Hub

In this exercise, you will establish a PowerShell Remoting session to the privileged endpoint to create a service principal and use the Azure Stack

Hub administrator portal to assign to it the Contributor role. The exercise consists of the following tasks:

1. Create a service principal
2. Assign the Contributor role to the service principal

### Task 1: Create a service principal

In this task, you will:

- Connect to the privileged endpoint via PowerShell and create a service principal.
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as administrator.
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to establish a PowerShell Remoting session to the privileged endpoint:

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

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to create the new app registration (and service principal object) and store the reference to it in the **\$spObject** variable:

```
powershell $spObject = Invoke-Command -Session $session -ScriptBlock {New-GraphApplication -Name 'azsmgmt-app1' -GenerateClientSecret}
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to retrieve the Azure Stack Hub stamp information and store the reference to it in the **\$azureStackInfo** variable:

```
powershell $azureStackInfo = Invoke-Command -Session $session -ScriptBlock {Get-AzureStackStampInformation}
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to terminate the PowerShell Remoting session of the privileged endpoint:

```
powershell $session | Remove-PSSession
```

**Note:** In general, you should use the **Close-PrivilegedEndpoint** cmdlet to close the privileged endpoint session. We are not following this practice for the sake of simplicity to avoid the need to set up a file share to host the session transcript logs.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to use the Azure Stack Hub stamp information you retrieved in earlier in this task to set values of the variables you will use to configure the service principal, referencing, respectively, the Azure Stack Hub endpoint used for Azure Resource Manager user operations, audience for acquiring an OAuth token used to access Graph API, and GUID of the identity provider:

```
powershell $armUseEndpoint = $azureStackInfo.TenantExternalEndpoints.TenantResourceManager $graphAudience = "https://graph." + $azureStackInfo.ExternalDomainFQDN + "/" $tenantID = $azureStackInfo.AADTenantID
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to register and set the Azure Stack Hub user environment:

```
powershell Add-AzEnvironment -Name 'AzureStackUser' -ArmEndpoint $armUseEndpoint
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to sign in as the service principal into the AzureStackUser environment:

```
powershell $securePassword = $spObject.ClientSecret | ConvertTo-SecureString -AsPlainText -Force $credential = New-Object -TypeName
```

```
System.Management.Automation.PSCredential -ArgumentList
$spObject.ClientId, $securePassword $spUserSignIn =
Connect-AzAccount -Environment 'AzureStackUser' -
ServicePrincipal -Credential $credential -TenantId
$tenantID
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to verify that the sign in was successful:

```
powershell $spUserSignIn
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to remove the current authentication context:

```
powershell Remove-AzAccount -Username
$credential.UserName
```

**Note:** Now you will repeat the equivalent sequence of steps to validate that you can authenticate to the Azure Stack Hub administrator environment:

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to use the Azure Stack Hub stamp information you retrieved in earlier in this task to set values of the variables you will use to configure the service principal, referencing, respectively, the Azure Stack Hub endpoint used for Azure Resource Manager administrative operations, audience for acquiring an OAuth token used to access Graph API, and GUID of the identity provider:

```
powershell $armAdminEndpoint =
$azureStackInfo.AdminExternalEndpoints.AdminResourceManag
er $graphAudience = "https://graph." +
$azureStackInfo.ExternalDomainFQDN + "/" $tenantID =
$azureStackInfo.AADTenantID
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to register and set the Azure Stack Hub admin environment:

```
powershell Add-AzEnvironment -Name 'AzureStackAdmin' -
ArmEndpoint $armAdminEndpoint
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to sign in as the service principal into the AzureStackAdmin environment::

```
powershell $spAdminSignIn = Connect-AzAccount -
Environment 'AzureStackAdmin' -ServicePrincipal -
Credential $credential -TenantId $tenantID
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to verify that the sign in was successful:

```
powershell $spAdminSignIn
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to display the properties of the new service principal:

```
powershell $spObject
```

**Note:** The output should have the following format:

```
ApplicationIdentifier : S-1-5-21-2657257302-3827180852-
1812683747-1510 ClientId : 6a3fb4ad-838a-47b1-a93c-
f3e4a1b683c8 Thumbprint : ApplicationName : Azurestack-
azsmgmt-app2-53508ec5-d7d9-4761-876a-3602542c2965
ClientSecret : 3fKPtUg37YraCk1IaFtdqeyTpVplXDqc25Dj3bUs
PSComputerName : AzS-ERCS01 RunspaceId : 6b142339-b67f-
490e-a258-40983c0cd8ea
```

**Note:** Record the value of the **ApplicationName** property. You will need it in the next task. In addition, you should record the value of the **ClientSecret** property and provide it to developers who implement the management application.

## Task 2: Assign the Contributor role to the service principal

In this task, you will:

- Assign the Contributor role to the service principal by using the Azure Stack Hub administrator portal.
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator](#)

[portal](#) and sign in as CloudAdmin@azurestack.local.

- In the web browser displaying the Azure Stack Hub administrator portal, in the hub menu, select **All services**.
- On the **All services** blade, select **General** and, in the list of services, select **Subscriptions**.
- On the **Subscriptions** blade, select **Default Provider Subscription**.
- On the **Default Provider Subscription** blade, select **Access control (IAM)**.
- On the **Default Provider Subscription | Access control (IAM)** blade, click **+ Add** and, in the drop-down menu, select **Add role assignment**.
- On the **Add role assignment** blade, specify the following settings and click **Save**:
  - Role: **Contributor**
  - Assign access to: **Azure AD user, group, or service principal**
  - Select: search for and select the value of the **ApplicationName** property of the service principal you identified in the previous task.
- Verify that the role assignment was successful.

**>Review: In this exercise, you have created a service principal and assigned to it the Contributor role.**

lab: title: 'Lab: Connect to Azure Stack Hub via PowerShell' module: 'Module 5: Manage Infrastructure'

---



# **Lab - Connect to Azure Stack Hub via PowerShell**

## **Student lab manual**

### **Lab dependencies**

- None

## Estimated Time

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to be able to manage your environment by using PowerShell. You also need to be able to occasionally connect to the Azure Stack Hub via PowerShell as a user.

# Objectives

In this lab, you will be able to:

- Connect to the ASDK operator and user environments via PowerShell

# Lab Environment

This lab uses an ASDK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

**Note:** For information regarding connecting to Azure Stack Hub integrated with Azure Active Directory (Azure AD), refer to [Connect to Azure Stack Hub with PowerShell](#).

The lab environment has the following configuration:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab.

# Instructions

## Exercise 1: Connecting to ASDK via Azure PowerShell

In this exercise, you will connect to the Admin ARM endpoint of ASDK from the ASDK host via PowerShell:

1. Install Azure Stack Hub compatible Az PowerShell modules
2. Download Azure Stack Hub tools.
3. Configure and connect to the Azure Stack Hub operator environment via PowerShell
4. Configure and connect to the Azure Stack Hub user environment via PowerShell

### Task 1: Install Azure Stack Hub compatible Azure PowerShell modules

In this task, you will:

- Remove any pre-existing Azure and Az PowerShell modules.
- Install and configure prerequisites for Azure Stack Hub compatible Az PowerShell modules.
- Install Azure Stack Hub compatible Az PowerShell modules.

**Note:** All versions of the Azure Resource Manager (AzureRM) PowerShell module are outdated, although not out of support. However, the Az PowerShell module is now the recommended PowerShell module for interacting with Azure and Azure Stack Hub.

1. If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
2. Within the Remote Desktop session to **AzS-HOST1**, start **Windows PowerShell** as Administrator.

3. From the **Administrator: Windows PowerShell** prompt, run the following to remove all existing versions of Azure PowerShell and Az PowerShell modules:

```
powershell Get-Module -Name Azure* -ListAvailable |  
Uninstall-Module -Force -Verbose -ErrorAction Continue  
Get-Module -Name Azs.* -ListAvailable | Uninstall-Module  
-Force -Verbose -ErrorAction Continue Get-Module -Name  
Az.* -ListAvailable | Uninstall-Module -Force -Verbose -  
ErrorAction Continue
```

**Note:** If you receive an error message regarding in-use modules, close the Windows PowerShell session, re-open it, and rerun the above commands.

4. From the **Administrator: Windows PowerShell** prompt, run the following to delete all the folders which names start with **Azure**, **Az** or **Azs** from the **C:\Program Files\WindowsPowerShell\Modules** and **C:\Users\AzureStackAdmin\Documents\PowerShell\Modules** folders.

```
powershell Get-ChildItem -Path 'C:\Program  
Files\WindowsPowerShell\Modules' -Include 'Az*' -Recurse  
-Force | Remove-Item -Force -Recurse Get-ChildItem -Path  
'C:\Users\AzureStackAdmin\Documents\PowerShell\Modules' -  
Include 'Az*' -Recurse -Force | Remove-Item -Force -  
Recurse
```

**Note:** If you receive an error message regarding in-use modules, close the Windows PowerShell session, re-open it, and rerun the above commands.

5. Within the Remote Desktop session to **AzS-HOST1**, start Microsoft Edge, navigate to the [PowerShell releases page](#).
6. From the [PowerShell releases page](#) page, download and install the latest release of PowerShell.
7. Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as administrator.
8. From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to configure PowerShell Gallery as a trusted repository

```
powershell Set-PSRepository -Name 'PSGallery' -
InstallationPolicy Trusted
```

## 9. From the **Administrator: C:\Program**

**Files\PowerShell\7\pwsh.exe** prompt, run the following to install PowerShellGet:

```
powershell Install-Module PowerShellGet -MinimumVersion
2.2.3 -Force
```

**Note:** Disregard any warning messages regarding in-use modules.

## 10. From the **Administrator: C:\Program**

**Files\PowerShell\7\pwsh.exe** window, run the following to install the PowerShell Az module for Azure Stack Hub:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Install-Module -Name
Az.BootStrapper -Force -AllowPrerelease -AllowClobber
Install-AzProfile -Profile 2019-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 2.0.2-
preview -AllowPrerelease
```

**Note:** Disregard any error messages regarding already available commands.

## Task 2: Download Azure Stack Hub tools

In this task, you will:

- Download Azure Stack Hub tools from GitHub
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, run the following to download the Azure Stack Tools:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Set-Location -Path
'C:\' Invoke-WebRequest
https://github.com/Azure/AzureStack-Tools/archive/az.zip
-OutFile az.zip Expand-Archive az.zip -DestinationPath .
-Force Set-Location -Path '\AzureStack-Tools-az'
```



**Note:** This step copies the archive containing the GitHub repository hosting the Azure Stack Hub tools to the local computer and expands the archive to the **C:\AzureStack-Tools-master** folder. The tools contain PowerShell modules that offer a range of features, including identifying Azure Stack Hub capabilities, managing Azure Stack Hub VM infrastructure and images, configuring Resource Manager policies, registering Azure Stack Hub with Azure, Azure Stack Hub deployment, connectivity to Azure Stack Hub, Azure Stack Hub tenant management, and validation of Azure Stack Hub Resource Manager templates.

### Task 3: Configure and connect to the Azure Stack Hub operator environment via PowerShell

In this task, you will:

- Configure the Azure Stack Hub operator environment via PowerShell
- Connect to the Azure Stack Hub operator environment via PowerShell
- Verify connection to the Azure Stack Hub operator environment via PowerShell
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to register your Azure Stack Hub operator PowerShell environment:

```
powershell Add-AzEnvironment -Name 'AzureStackAdmin' -
ArmEndpoint
'https://adminmanagement.local.azurestack.external' ` -
AzureKeyVaultDnsSuffix
adminvault.local.azurestack.external ` -
AzureKeyVaultServiceEndpointResourceId
https://adminvault.local.azurestack.external
```

**Note:** Verify that the command returns the following output:

```
powershell Name Resource Manager Url ActiveDirectory
Authority ----
---- AzureStackAdmin
https://adminmanagement.local.azurestack.external
https://adfs.local.azurestack.external/adfs/
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to sign in to your Azure Stack Hub operator PowerShell environment with the AzureStack\CloudAdmin credentials:

```
powershell Connect-AzAccount -EnvironmentName  
'AzureStackAdmin' -UseDeviceAuthentication
```

- In the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, review the resulting message, open a web browser window, navigate to the [adfs.local.azurestack.external](https://adfs.local.azurestack.external) page, type the code included in the reviewed message, and click **Continue**.
- When prompted, sign in by using the following credentials:
  - Username: **CloudAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Switch back to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and verify that you have successfully authenticated as **CloudAdmin@azurestack.local**.
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to list the Azure Stack Hub admin subscriptions

```
powershell Get-AzSubscription
```

**Note:** Verify that the output includes **Default Provider Subscription, Metering Subscription, and Consumption Subscription**.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to verify the corresponding PowerShell environment context:

```
powershell Get-AzContext
```

## **Task 4: Configure and connect to the Azure Stack Hub user environment via PowerShell**

In this task, you will:

- Configure the Azure Stack Hub user environment via PowerShell
- Connect to the Azure Stack Hub user environment via PowerShell
- Verify connection to the Azure Stack Hub user environment via PowerShell
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7.
- From the **C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to register an Azure Resource Manager environment that targets your Azure Stack Hub user environment:

```
powershell Add-AzEnvironment -Name 'AzureStackUser' -
ArmEndpoint
'https://management.local.azurestack.external'
```

**Note:** Verify that the command returns the following output:

```
powershell Name Resource Manager Url ActiveDirectory
Authority ----
-----
---- AzureStackUser
https://management.local.azurestack.external
https://adfs.local.azurestack.external/adfs/
```

- From the **C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to sign in to your Azure Stack Hub PowerShell environment.

```
powershell Connect-AzAccount -EnvironmentName
'AzureStackUser'
```

**Note:** This will automatically open a web browser window prompting you to authenticate.

- When prompted, sign in by using the following credentials:
  - Username: **CloudAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to list the Azure Stack Hub admin subscriptions

```
powershell Get-AzSubscription
```

**Note:** Verify that the output does **not** include **Default Provider Subscription, Metering Subscription, and Consumption Subscription.**

**>Review: In this exercise, you have connected to the Azure Stack Hub operator and user environments via PowerShell.**

lab: title: 'Lab: Access the Privileged Endpoint in Azure Stack Hub'  
module: 'Module 5: Manage Infrastructure'

---

# **Lab - Access the Privileged Endpoint in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to identify the method of accessing the privileged endpoint.



# Objectives

After completing this lab, you will be able to:

- Access the Azure Stack Hub privileged endpoint

# Lab Environment

This lab uses an ASDK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

## Exercise 1: Manage Azure Stack Hub via the privileged endpoint

In this exercise, you will establish a PowerShell Remoting session to the privileged endpoint to run Windows PowerShell cmdlets accessible via the Remoting session. The exercise consists of the following tasks:

1. Connect to the privileged endpoint via Windows PowerShell
2. Review the functionality available via the privileged endpoint
3. Close the connection to the privilege endpoints and collect the session transcript

## Task 1: Connect to the privileged endpoint via Windows PowerShell

In this task, you will:

- Connect to the privileged endpoint via Windows PowerShell
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell ISE as administrator.
- From the **Administrator: Windows PowerShell ISE** console, run the following to identify the IP address of the infrastructure VM running the privileged endpoint:

```
powershell $ipAddress = (Resolve-DnsName -Name AzS-ERCS01).IPAddress
```

- From the **Administrator: Windows PowerShell ISE** window, run the following to add the IP address of the infrastructure VM running the privileged endpoint to the list of WinRM trusted hosts (unless all hosts are already allowed):

```
powershell $trustedHosts = (Get-Item -Path WSMAN:\localhost\Client\TrustedHosts).Value If ($trustedHosts -ne '*') { If ($trustedHosts -ne '') { $trustedHosts += ",ipAddress" } else { $trustedHosts = "$ipAddress" } } Set-Item WSMAN:\localhost\Client\TrustedHosts -Value $trustedHosts -Force
```

- From the **Administrator: Windows PowerShell ISE** window, run the following to store the Azure Stack Hub admin credentials in a variable:

```
powershell $adminUserName = 'CloudAdmin@azurestack.local' $adminPassword = 'Pa55w.rd1234' | ConvertTo-SecureString -Force -AsPlainText $adminCredentials = New-Object PSCredential($adminUserName,$adminPassword)
```

- From the **Administrator: Windows PowerShell ISE** window, run the following to establish a PowerShell Remoting session to the privileged endpoint:

```
powershell Enter-PSSession -ComputerName $ipAddress -
ConfigurationName PrivilegedEndpoint -Credential
$adminCredentials
```

- Verify that the PowerShell Remoting session has been successfully established. The console pane in the Windows PowerShell ISE window should be displaying the prompt starting with the IP address of the infrastructure VM running the privileged endpoint enclosed in square brackets.

## Task 2: Review the functionality available via the privileged endpoint

In this task, you will:

- Review the functionality available via the privileged endpoint.
- Within the Remote Desktop session to **AzS-HOST1**, from the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, in the console pane, run the following to identify all available PowerShell cmdlets:

```
powershell Get-Command
```

- From the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, run the following to identify current Cloud Admin user accounts:

```
powershell Get-CloudAdminUserList
```

**Note:** The list should include only two accounts - CloudAdmin and AzureStackAdmin.

- From the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, run the following to validate update readiness of Azure Stack Hub and review the results:

```
powershell Test-AzureStack -Group UpdateReadiness
```

**Note:** Keep in mind that ASDK does not support updates, so this is strictly for demonstration purposes.

**Note:** For an introduction to the **Test-AzureStack** functionality, refer to [Validate Azure Stack Hub system state](#).

**Note:** During support scenarios, a Microsoft support engineer might need to elevate the privileged endpoint PowerShell session to access the internals of the Azure Stack Hub infrastructure. This process is referred to as unlocking the privileged endpoint. The session elevation process is a two step, two people, two organization authentication process. The unlock procedure is initiated by the Azure Stack Hub operator, who retains control of their environment at all times. You will step through the emulated scenario illustrating this process next.

- From the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, run the following to validate that the privileged endpoint is currently locked:

```
powershell Get-SupportSessionInfo
```

- From the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, run the following to generate the support session token:

```
powershell Get-SupportSessionToken
```

**Note:** In a support scenario, you would pass the request token to a Microsoft support engineer via a medium of their choice, such as chat or email. The Microsoft support engineer then would use the request token to generate a support session authorization token and relay its value to you. Within the same PowerShell Remoting session, you would next run the **Unlock-SupportSession** cmdlet and, when prompted, provide the value of the support session authorization token. At that point, the PowerShell Remoting session would become elevated, with full admin capabilities and full reachability into the infrastructure.

**Task 3: Close the session to the privilege endpoints and collect the session transcript**

In this task, you will:

- Close the session to the privilege endpoints and collect the session transcript.

**Note:** Privileged endpoint logs every action and its output. To collect the logs, close the session by using the **Close-PrivilegedEndpoint** cmdlet. This closes the endpoint and transfers the log files to an external file share for retention.

**Note:** You will start by creating a file share to store the privileged endpoint logs.

1. Within the Remote Desktop session to **AzS-HOST1**, start another PowerShell ISE as administrator.
2. From the **Administrator: Windows PowerShell ISE** console, run the following to create and configure share that will store the privileged endpoint session logs:

```
powershell $pepGroup = 'AZURESTACK\CloudAdmins' New-Item -Path 'C:\PEPLogs' -ItemType Directory -Force $pepShare = New-SmbShare -Name 'PEPLogs' -Description 'PEPLogs' -Path 'C:\PEPLogs' Grant-SmbShareAccess -Name $pepShare.Name -AccountName $pepGroup -AccessRight Full -Force Revoke-SmbShareAccess -Name $pepShare.Name -AccountName 'Everyone' -Force
```

3. Switch back to the PowerShell Remoting session in the **Administrator: Windows PowerShell ISE** window, run the following to close the privileged endpoint session and transfers the session log files to an external file share for retention:

```
powershell Close-PrivilegedEndpoint -TranscriptsPathDestination '\\AzS-HOST1.azurestack.local\PEPLogs' -Credential $using:adminCredentials
```

4. Wait until the cmdlet completes and, in File Explorer, review the content of the **C:\PEPLogs** folder.

**>Review: In this exercise, you have established a PowerShell Remoting session to the privileged endpoint, reviewed its functionality, and closed the session.**

lab: title: 'Lab: Manage Log Collection in Azure Stack Hub' module: 'Module 5: Manage Infrastructure'

---

# **Lab - Manage Log Collection in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None



## **Estimated Time**

30 minutes

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to identify methods to perform log collection.

# Objectives

After completing this lab, you will be able to:

- Perform Azure Stack Hub log collection

# Lab Environment

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

## Exercise 1: Explore Azure Stack Hub diagnostic logs collection capabilities

In this exercise, you will explore different options for managing diagnostic logs collection options.

1. Enable proactive diagnostic log collection
2. Send diagnostic logs on demand
3. Copy diagnostic logs to a local file share

### Task 1: Enable proactive diagnostic log collection

In this task, you will:

- Enable proactive log collection.

**Note:** Proactive log collection automatically collects and sends diagnostic logs from Azure Stack Hub to Microsoft before you open a support case. These logs are only collected when a system health alert is raised and are only accessed by Microsoft Support in the context of a support case.

1. If needed, sign in to **AzSHOST-1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
2. Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
3. In the web browser window displaying the Azure Stack Hub administrator portal, in the hub menu, click **Help + support**.
4. On the **Overview** blade, click **Log Collection**.
5. On the **Overview | Log Collection** blade, click **Enable proactive log collection**.

**Note:** If the **Enable proactive log collection** option is not available, proceed directly to the next step.
6. On the **Settings** blade, specify the following settings and click **Save**.
  - Proactive log collection: **Enable**
  - Log location: **Azure (Recommended)**
7. Back on the **Overview | Log Collection** blade, click **Settings** to verify that the proactive log collection is configured according to settings you specified.

## **Task 2: Send diagnostic logs on demand**

In this task, you will:

- Send diagnostic logs on demand by using the Azure Stack Hub administration portal.

- Send diagnostic logs on demand by using Azure Stack Hub PowerShell.

**Note:** This functionality is referred to as *Send logs now*

**Note:** You will start by sending diagnostic logs on demand by using the Azure Stack Hub administration portal.

1. Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the [Azure Stack Hub administrator portal](#), in the hub menu, click **Help + support**.
2. On the **Overview** blade, click **Log Collection**.
3. On the **Overview | Log Collection** blade, click **Send logs now**.
4. On the **Send logs now** blade, specify the following settings and click **Collect + upload**.
  - Start: current date and time - 3 hours
  - End: current date and time

**Note:** Do not wait for the upload to complete but instead proceed to the next step. The log collection will fail. This is expected since the target Azure subscription is not directly accessible from the lab environment.

**Note:** Now you will configure the equivalent functionality by using Azure Stack Hub PowerShell. This requires connecting to the privileged endpoint.

5. Within the Remote Desktop session to **AzS-HOST1**, start PowerShell ISE as administrator.
6. From the **Administrator: Windows PowerShell ISE** console, run the following to identify the IP address of the infrastructure VM running the privileged endpoint:

```
powershell $ipAddress = (Resolve-DnsName -Name AzS-ERCS01).IPAddress
```

7. From the **Administrator: Windows PowerShell ISE** window, run the following to add the IP address of the infrastructure VM running the privileged endpoint to the list of WinRM trusted hosts (unless all hosts are already allowed):

```
powershell $trustedHosts = (Get-Item -Path
WSMan:\localhost\Client\TrustedHosts).Value If
($trustedHosts -ne '*') { If ($trustedHosts -ne '') {
$trustedHosts += ",ipAddress" } else { $trustedHosts =
"$ipAddress" } } Set-Item
WSMan:\localhost\Client\TrustedHosts -Value $TrustedHosts
-Force
```

8. From the **Administrator: Windows PowerShell ISE** window, run the following to store the Azure Stack Hub admin credentials in a variable:

```
powershell $adminUserName = 'CloudAdmin@azurestack.local'
$adminPassword = 'Pa55w.rdl234' | ConvertTo-SecureString
-Force -AsPlainText $adminCredentials = New-Object
PSCredential($adminUserName,$adminPassword)
```

9. From the **Administrator: Windows PowerShell ISE** window, run the following to establish a PowerShell Remoting session to the privileged endpoint:

```
powershell Enter-PSSession -ComputerName $ipAddress -
ConfigurationName PrivilegedEndpoint -Credential
$adminCredentials
```

**Note:** Verify that the PowerShell Remoting session has been successfully established. The console pane in the Windows PowerShell ISE window should be displaying the prompt starting with the IP address of the infrastructure VM running the privileged endpoint enclosed in square brackets.

10. From the PowerShell Remoting session in the console pane of the **Administrator: Windows PowerShell ISE** window, run the following to send Azure Stack Hub storage diagnostic logs on demand:

```
powershell Send-AzureStackDiagnosticLog -FilterByRole
Storage
```

**Note:** Do not wait for the upload to complete but instead proceed to the next step. The log collection will fail. This is expected since the target Azure subscription is not directly accessible from the lab environment.

**Note:** You have the option of filtering by role as well as specify the time window for which logs should be collected.

For details, refer to [Diagnostic log collection](#).

### Task 3: Copy diagnostic logs to a local file share

In this task, you will:

- Copy diagnostic logs to a local file share by using the Azure Stack Hub administration portal.
- Copy diagnostic logs to a local file share by using Azure Stack Hub PowerShell.

**Note:** You will start by creating a file share to store logs.

1. Within the Remote Desktop session to **AzS-HOST1**, start File Explorer.
2. In File Explorer, create a new folder **C:\AzSHLogs**.
3. In File Explorer, right-click the **AzSHLogs** folder and, in the right-click menu, click **Properties**.
4. In the **AzSHLogs Properties** window, click the **Sharing** tab and then click **Advanced Sharing**.
5. In the **Advanced Sharing** dialog box, click **Share this folder** and then click **Permissions**.
6. In the **Permissions for AzSHLogs** window, ensure that the **Everyone** entry is selected and then click **Remove**.
7. Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **CloudAdmins** and click **OK**.
8. Ensure that the **CloudAdmins** entry is selected and click the **Full Control** checkbox in the **Allow** column.
9. Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, click **Locations**.
10. In the **Locations** dialog box, click the entry representing the local computer (**AzS-HOST1**) and click **OK**.
11. In the **Enter the object names to select** text box, type **Administrators** and click **OK**.
12. Ensure that the **Administrators** entry is selected, click the **Full Control** checkbox in the **Allow** column, and then click **OK**.
13. Back in the **Advanced Sharing** dialog box, click **OK**.
14. Back in the **AzSHLogs Properties** window, click the **Security** tab and click **Edit**.
15. Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **CloudAdmins** and click **OK**.



16. In the **Permissions for AzSHLogs** dialog box, in the list of entries in the **Groups or user names** pane, click **CloudAdmins**, in the **Permissions for CloudAdmins** pane, click **Full Control** in the **Allow** column and then click **OK**.

17. Back in the **AzSHLogs Properties** window, click **Close**.

**Note:** Next, you can configure diagnostic log collection to a file share from either the Azure Stack Hub administration portal or via the privileged endpoint.

18. Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the [Azure Stack Hub administrator portal](#), in the hub menu, click **Help + support**.

19. On the **Overview** blade, click **Log Collection**.

20. On the **Overview | Log Collection** blade, click **Settings**.

21. On the **Settings** blade, change the **Log location** option from **Azure (Recommended)** to **Local file share** and specify the following settings:

- SMB fileshare path: **\\AzS-HOST1.azurestack.local\\AzSHLogs**
- Username: **AZURESTACK\\AzureStackAdmin**
- Password: **Pa55w.rd1234**

22. On the **Settings** blade, click **Save**.

23. Back on the **Overview | Log Collection** blade, click **Send logs now**.

24. On the **Send logs now** blade, specify the following settings and click **Collect + upload**.

- Start: current date and time - 3 hours
- End: current date and time

**Note:** To view the progress of the log collection and upload, on the **Overview | Log Collection** blade, click **Refresh**.

**Note:** Do not wait for the upload to complete but instead proceed to the next step. The log collection should succeed but

it might take about 15 minutes for it to complete.

**Note:** Now you will configure the equivalent functionality by using Azure Stack Hub PowerShell. This requires connecting to the privileged endpoint.

25. Within the Remote Desktop session to **AzS-HOST1**, switch back to the **Administrator: Windows PowerShell** console from which you connected to the privileged endpoint in the previous task.

26. From the PowerShell Remoting session in the console pane of the **Administrator: Windows PowerShell** window, run the following to copy Azure Stack Hub storage diagnostic logs to a local file share:

```
powershell Get-AzureStackLog -OutputSharePath '\\AzS-  
HOST1.azurestack.local\AzSHLogs' -OutputShareCredential  
$using:adminCredentials -FilterByRole Storage
```

27. Wait until the cmdlet completes and, in File Explorer, review the content of the **C:\AzSHLogs** folder.

**Note\*:** *The folder should contain folders corresponding to each individual copy you initiated. The folders should have the names in the format AzureStackLogs-YYYYMMDDHHMMSS-AZS-ERCS01, where YYYYMMDDHHMMSS\*\*\* represents the timestamp of the copy.*

28. From the PowerShell Remoting session prompt in the **Administrator: Windows PowerShell** window, run the following to close the session:

```
powershell Close-PrivilegedEndpoint -  
TranscriptsPathDestination '\\AzS-  
HOST1.azurestack.local\AzSHLogs' -Credential  
$using:adminCredentials
```

**>Review: In this exercise, you have established a PowerShell Remoting session to the privileged endpoint and used PowerShell cmdlets available from that endpoint to collect diagnostics logs.**

lab: title: 'Lab: Configure and manage Azure Stack Hub Storage Accounts' module: 'Module 5: Manage Infrastructure'

---

# **Lab - Configure and manage Azure Stack Hub Storage Accounts**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

30 minutes

## Lab scenario

You are an operator of an Azure Stack Hub environment. You need to configure and manage its storage accounts.

# Objectives

After completing this lab, you will be able to:

- Configure Azure Stack Hub storage
- Manage Azure Stack Hub storage accounts

# Lab Environment

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will create an additional user accounts in the course of this lab.

## Exercise 0: Prepare for the lab

In this exercise, you will create an Active Directory user account that you will be using in this lab:

1. Create a tenant user account (as a cloud operator)

### Task 1: Create a tenant user account (as a cloud operator)

In this task, you will:

- Create a tenant user account (as a cloud operator)



- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**

**Review:** After completing this exercise, you have created the Active Directory account you will use in this lab.

## **Exercise 1: Configure Azure Stack Hub storage**


In this exercise, you will configure Azure Stack Hub storage:

1. Configure storage account retention
2. Create a plan containing Storage services
3. Create an offer based on the plan

### **Task 1: Configure storage account retention**

In this task, you will:

- Configure storage account retention

- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the web browser window displaying the Azure Stack Hub administrator portal, click **All services**.
- On the **All services** blade, select the **Administration** section and click **Region management**.
- On the **local** blade, click **Resource providers**.
- On the **Resource providers** blade, click **Storage**.
- On the **Storage** blade, click **Configuration**.
- On the **Storage**  **Configuration** blade, in the **Retention period for deleted storage accounts (days)** text box, type 10 and click **Save**.

## Task 2: Create a plan containing Storage services

In this task, you will:

- Create a plan containing Storage services.
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **storage-plan1**
  - Resource name: **storage-plan1**
  - Resource group: the name of a new resource group **storage-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Storage** checkbox.
- Click **Next: Quotas>**.

- On the **Quotas** tab of the **New plan** blade, next to the **Microsoft.Storage** drop-down list, click **Create New**.
- On the **Create Storage quota** blade, specify the following settings and click **OK**:
  - Name: **storage-plan1-storage-quota**
  - Maximum capacity (GB): **200**
  - Total number of storage accounts: **5**
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### Task 3: Create an offer based on the plan

In this task, you will:

- Create an offer based on the plan.
- Within the Remote Desktop session to **AzS-HOST1**, in the web browser window displaying the Azure Stack Hub administrator portal, click + **Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **storage-offer1**
  - Resource name: **storage-offer1**
  - Resource group: **storage-offers-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **storage-plan1** entry.
- Click **Next: Add-on plans >**.

- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

**Review:** After completing this exercise, you have configured the deleted storage account retention setting and created a public offer containing a **Microsoft.Storage** service-based plan.

## Exercise 2: Manage Azure Stack Hub storage accounts.

In this exercise, you will manage Azure Stack Hub storage accounts:

1. Create and delete Azure storage accounts (as a tenant user)
2. Recover a deleted storage account (as a cloud operator)
3. Reclaim storage capacity (as a cloud operator)

### Task 1: Create and delete Azure storage accounts (as a tenant user)

In this task, you will:

- Create and delete Azure storage accounts (as a tenant user)
- Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, on the Dashboard, click the **Get a subscription** tile.
- On the **Get a subscription** blade, in the **Name** text box, type **t1u1-storage-subscription1**.
- In the list of offers, select **storage-offer1** and click **Create**.
- When presented with the message **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.
- In the Azure Stack Hub tenant portal, in the hub menu, click **All services**.
- In the list of services, click **Subscriptions**.
- On the **Subscriptions** blade, click **t1u1-storage-subscription1**.

- On the **t1u1-base-subscription1** blade, click **Resources**.
- On the **t1u1-base-subscription1 - Resources** blade, click **+ Add**.
- On the **New** blade, click **Data + Storage** and then click **Storage account**.
- On the **Basics** tab of the **Create storage account** blade, specify the following settings (leave others with their default values):
  - Resource group: the name of a new resource group **storage-RG**
  - Name: a unique name consisting of between 3 and 24 lower case letters or digits
  - Location: **local**
  - Performance: **Standard**
  - Account kind: **Storage (general purpose v1)**
  - Replication: **Locally-redundant storage (LRS)**
- On the **Basics** tab of the **Create storage account** blade, click **Next: Advanced >**.
- On the **Advanced** tab of the **Create storage account** blade, leave the default settings in place and click **Review + create**.
- On the **Review + create** tab of the **Create storage account** blade, click **Create**.

**Note:** Wait until the storage account is provisioned. This should take about one minute.

- Create another storage account with the same settings in the same resource group by following the steps 11-16.

**Note:** Next, you will delete both storage accounts. Make sure to record their names first. You will need them later in this lab.

- In the Azure Stack Hub user portal, in the hub menu, click **All services** and then, on the **All services** blade, click **Storage accounts**.
- On the **Storage accounts** blade, ensure that the subscription filter includes the **t1u1-storage-subscription1**, in the list of storage accounts, select both storage accounts you created earlier in this task, and click **Delete**.

- On the **Delete Resources** blade, in the **Confirm delete** text box, type **yes** and click **Delete**.

## Task 2: Recover a deleted storage account (as a cloud operator)

In this task, you will:

- Recover a deleted storage account (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, switch to the web browser window displaying the Azure Stack Hub administrator portal, click **All services**, on the **All services** blade, click **Administration**, and, in the list of administration services, click **Storage**.
- On the **Storage** blade, click **Storage accounts**.
- On the **Storage | Storage accounts** blade, in the **Account name** filter text box, type the name of the first storage account you created in the previous task and verify that the storage account name is listed with the **Deleted** status.
- Click the entry representing the first deleted storage account.
- On the **Storage account** blade, click **Recover**. When prompted to confirm, click **Yes**.

**Note:** Wait for the operation to complete. This should take less than a minute. Keep in mind that it might take some extra time for a recovered storage account to appear in the user portal.

- Close the **Storage account** blade.

## Task 3: Reclaim storage capacity (as a cloud operator)

In this task, you will:

- Reclaim storage capacity (as a cloud operator)
- Within the Remote Desktop session to **AzS-HOST1**, in the Azure Stack Hub administrator portal, while logged on as CloudAdmin@azurestack.local, on the **Storage | Storage accounts** blade, click **Refresh**.

- With the name of the first storage account in the **Account name** filter text box, verify that the storage account is now listed with the **Active** status.
- On the **Storage | Storage accounts** blade, in the **Account name** filter text box, type the name of the second storage account you created in the previous task and verify that the storage account name is listed with the **Deleted** status.
- On the **Storage | Storage accounts** blade, click **Reclaim space** and, when prompted to confirm, click **Yes**.

**Note:** Wait for the operation to complete. This should take less than a minute.

- Refresh the browser page.
- Back on the **Storage | Storage accounts** blade, in the **Account name** filter text box, type the name of the second storage account you created in the previous task and verify that its name no longer appears in the list of search results.

**>Review: In this exercise, you have created and deleted storage accounts as a tenant user as well as recovered a deleted storage account and reclaimed storage capacity as a cloud operator.**

lab: title: 'Lab: Manage Public IP Addresses in Azure Stack Hub'  
module: 'Module 5: Manage Infrastructure'

---



# **Lab - Manage Public IP Addresses in Azure Stack Hub**

## **Student lab manual**

### **Lab dependencies**

- None

## **Estimated Time**

30 minutes

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to manage public IP address resources.

# Objectives

After completing this lab, you will be able to:

- Manage public IP address resources

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment consists of the following components:

- ASDK deployment running on the **AzS-HOST1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab. You will also create additional user accounts.

# Instructions

## Exercise 0: Prepare for the lab

In this exercise, you will create an Active Directory user account that you will be using in this lab:

1. Create a user account (as a cloud operator)

## Task 1: Create a user account (as a cloud operator)

In this task, you will:

- Create a user account (as a cloud operator)
- If needed, sign in to **AzS-HOST1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **T1U1**
  - User UPN logon: **t1u1@azurestack.local**
  - User SamAccountName: **azurestack\t1u1**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**

**Review:** In this exercise, you have created the Active Directory account you will use in this lab.

## **Exercise 1: Create an offer (as a cloud operator)**

In this exercise, you will act as a cloud operator. First, you will review the public IP address usage, then you will create a plan consisting of the network services and an offer containing this plan. Next, you will make the offer public, allowing users to create subscriptions based on this offer. The exercise consists of the following tasks:

1. Review public IP address usage (as a cloud operator)
2. Create a plan consisting of the network services (as a cloud operator).
3. Create an offer based on the plan and make the offer public (as a cloud operator)

### **Task 1: Review public IP address usage (as a cloud operator)**

In this task, you will:

- Review public IP address usage (as a cloud operator).
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the Azure Stack Hub administrator portal, on the **Dashboard** page, in the **Resource providers** tile, click **Network**.
- On the **Network** blade, note the **Public IP pools usage** graph and the numbers of used and free IP addresses.

### **Task 2: Create a plan consisting of the network services (as a cloud operator)**

In this task, you will:

- Create a plan consisting of the network services (as a cloud operator).

- In the web browser window displaying the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Plan**.
- On the **Basics** tab of the **New plan** blade, specify the following settings:
  - Display name: **Network-plan1**
  - Resource name: **network-plan1**
  - Resource group: the name of a new resource group **network-plans-RG**
- Click **Next: Services >**.
- On the **Services** tab of the **New plan** blade, select the **Microsoft.Network** checkbox.
- Click **Next: Quotas >**.
- On the **Quotas** tab of the **New plan** blade, select **Create New**.
- On the **Create Network quota** blade, specify the following settings and click **OK**:
  - Name: **Network-plan1-quota**
  - Max virtual networks: **2**
  - Max virtual network gateways: **2**
  - Max network connections: **2**
  - Max public IPs: **20**
  - Max NICs **20\*\***
  - Max load balancers: **5**
  - Max network security groups: **20**
- Click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

### **Task 3: Create an offer based on the plan (as a cloud operator)**

In this task, you will:



- Create an offer based on the plan (as a cloud operator)
- In the Azure Stack Hub administrator portal, click **+ Create a resource**.
- On the **New** blade, click **Offers + Plans**.
- On the **Offers + Plans** blade, click **Offer**.
- On the **Basics** tab of the **Create a new offer** blade, specify the following settings:
  - Display name: **Network-offer1**
  - Resource name: **network-offer1**
  - Resource group: a new resource group named **network-offers-RG**
  - Make this offer public: **Yes**
- Click **Next: Base plans >**.
- On the **Base plans** tab of the **Create a new offer** blade, select the checkbox next to the **Network-plan1** entry.
- Click **Next: Add-on plans >**.
- Leave **Add-on plans** settings with their default values, click **Review + create** and then click **Create**.

**Note:** Wait for the deployment to complete. This should take just a few seconds.

**Review:** In this exercise, you have created a plan and a public offer based on that plan.

## **Exercise 2: Create public IP address resources (as a user)**

In this exercise, you will act as users who signs up for the offer you created in the first exercise, creates a new subscription, and creates public IP address resources in that subscription. The exercise consists of the following tasks:

1. Sign up for the offer (as a user)
2. Connect to Azure Stack Hub user Azure Resource Manager endpoint (as a user)

### 3. Create IP address resources (as a user)

#### **Task 1: Sign up for the offer (as a user)**

In this task, you will:

- Sign up for the offer (as a user)
- 1. Within the Remote Desktop session to **AzS-HOST1**, start an InPrivate session of the web browser.
- In the web browser window, navigate to the [Azure Stack Hub user portal](#) and sign in as **t1u1@azurestack.local** with the password **Pa55w.rd**.
- In the Azure Stack Hub user portal, on the dashboard, click **Get a subscription**.
- On the **Get a subscription** blade, in the **Display name** text box, type **T1U1-network-subscription1**.
- In the list of offers, select **Network-offer1** and then click **Create**.
- In the message box **Your subscription has been created. You must refresh the portal to start using your subscription**, click **Refresh**.

#### **Task 2: Connect to Azure Stack Hub Azure Resource Manager user endpoint (as a user)**

In this task, you will:

- Connect to Azure Stack Hub Azure Resource Manager user endpoint (as a user)
- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as Administrator.

**Note:** For detailed instructions on setting up PowerShell connectivity to Azure Stack Hub, follow instructions in the lab **Connect to Azure Stack Hub via PowerShell**.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to install the Azure Stack Hub PowerShell modules required for this lab:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Install-Module -Name
Az.BootStrapper -Force -AllowPrerelease -AllowClobber
Install-AzProfile -Profile 2019-03-01-hybrid -Force
Install-Module -Name AzureStack -RequiredVersion 2.0.2-
preview -AllowPrerelease
```

**Note:** Disregard any error messages regarding already available commands.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to download and extract the Azure Stack Hub tools:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 Set-Location -Path
'C:\' Invoke-WebRequest
https://github.com/Azure/AzureStack-Tools/archive/az.zip
-OutFile az.zip Expand-Archive az.zip -DestinationPath .
-Force Set-Location -Path '\AzureStack-Tools-az'
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to register your Azure Stack Hub user environment:

```
powershell Add-AzEnvironment -Name 'AzureStackUser' -
ArmEndpoint
'https://management.local.azurestack.external'
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to initiate authentication to the Azure Stack Hub user environment via a browser session as the **t1u1@azurestack.local** user:

```
powershell Connect-AzAccount -EnvironmentName
'AzureStackUser' -UseDeviceAuthentication
```

- In the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window, review the resulting message, open another web browser window in the InPrivate mode, navigate to the [adfs.local.azurestack.external](https://adfs.local.azurestack.external) page, and type in the code included in the reviewed message. If prompted, sign in again as the **t1u1@azurestack.local** user.
- Switch back to the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** window and verify that you have

successfully authenticated as the **t1u1@azurestack.local** user.

### Task 3: Create IP address resources (as a user)

In this task, you will:

- Create IP address resources (as a user)
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to verify that you are using the newly provisioned subscription:

```
powershell (Get-AzSubscription).Name
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to register the Network Resource Provider within the current subscription:

```
powershell Register-AzResourceProvider -ProviderNamespace Microsoft.Network
```

**Note:** You have to register a resource provider in order to create resources managed by that provider.

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to create a resource group that will host public IP address resources:

```
powershell $rg = New-AzResourceGroup -Name publicIPs-RG -Location local
```

- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to create public IP address resources:

```
powershell 1..5 | ForEach-Object {New-AzPublicIpAddress -Name "publicIP$_" -ResourceGroupName $rg.ResourceGroupName -AllocationMethod Static -Location local}
```

- Wait until all IP address resources are provisioned.

**Review:** After completing this exercise, you have created public IP address resources in a user's subscription.

## Exercise 4: Manage public IP address usage (as a cloud operator)

In this exercise, you will act as a cloud operator, reviewing and managing the public IP address usage. The exercise consists of the following tasks:

1. Review public IP address usage
2. Add a public IP address pool

### Task 1: Review public IP address usage (as a cloud operator)

In this task, you will:

- Review public IP address usage (as a cloud operator).
- Switch to the web browser window displaying the Azure Stack Hub administrator portal, where you are signed in as CloudAdmin@azurestack.local.
- In the Azure Stack Hub administrator portal, in the hub menu, click **Dashboard** and, on the **Resource providers** tile, click **Network**.
- On the **Network** blade, review again the **Public IP pools usage** graph and the numbers of used and free IP addresses.

**Note:** The numbers should have changed, reflecting additional 5 public IP addresses you created in the user subscription (as the user).

### Task 2: Add a public IP address pool (as a cloud operator)

In this task, you will:

- Add a public IP address pool
- In the web browser window displaying the Azure Stack Hub administrator portal, on the **Network** blade, click the **Public IP pools usage** tile.

**Note:** If the **Public IP pools** blade displays the message **The exclusive operation 'Startup' is in progress. Add node and add IP pool operations are disabled while the operation is running. Click here to view the activity log**, then you will need to wait until the 'Startup' operation completes before you proceed to the next step.

- On the **Public IP pools** blade, click **+ Add IP pool**.
- On the **Add IP pool** blade, specify the following settings and click **Add**.
  - Name: **Public Pool 1**
  - Region: **local**
  - Address range (CIDR block): **192.168.110.0/24**
- Wait for the change to take effect and navigate back to the **Network** blade.
- Review the **Public IP pools usage** graph and note the changes in the number of free IP addresses.

**>Review: In this exercise, you have reviewed and configured public IP address pools.**

lab: title: 'Lab: Configure Azure Stack Hub Infrastructure Backup'  
module: 'Module 5: Manage Infrastructure'

---

# **Lab - Configure Azure Stack Hub Infrastructure Backup**

## **Student lab manual**

### **Lab depedndencies**

- None



## **Estimated Time**

30 minutes

## **Lab scenario**

You are an operator of an Azure Stack Hub environment. You need to prepare it for infrastructure backup.

# Objectives

After completing this lab, you will be able to:

- Configure Azure Stack Hub infrastructure backup

# Lab Environment

This lab uses an ADSK instance integrated with Active Directory Federation Services (AD FS) (backed up Active Directory as the identity provider).

The lab environment consists of the following components:

- ASDK deployment running on the **AzSHOST-1** server with the following access points:
- Administrator portal: <https://adminportal.local.azurestack.external>
- Admin ARM endpoint:  
<https://adminmanagement.local.azurestack.external>
- User portal: <https://portal.local.azurestack.external>
- User ARM endpoint: <https://management.local.azurestack.external>
- Administrative users:
- ASDK cloud operator username: **CloudAdmin@azurestack.local**
- ASDK cloud operator password: **Pa55w.rd1234**
- ASDK host administrator username:  
**AzureStackAdmin@azurestack.local**
- ASDK host administrator password: **Pa55w.rd1234**

You will install software necessary to manage Azure Stack Hub via PowerShell in the course of this lab. You will also create additional user accounts.

## Exercise 1: Configure Azure Stack Hub infrastructure backup

In this exercise, you will prepare configure Azure Stack Hub infrastructure backup:

1. Create a backup user
2. Create a backup share
3. Generate an encryption key

#### 4. Configure backup controller

##### Task 1: Create a backup user

In this task, you will:

- Create a backup user
- If needed, sign in to **AzSHOST-1** by using the following credentials:
  - Username: **AzureStackAdmin@azurestack.local**
  - Password: **Pa55w.rd1234**
- Within the Remote Desktop session to **AzS-HOST1**, click **Start**, in the Start menu, click **Windows Administrative Tools**, and, in the list of administrative tools, double-click **Active Directory Administrative Center**.
- In the **Active Directory Administrative Center** console, click **azurestack (local)**.
- In the details pane, double-click the **Users** container.
- In the **Tasks** pane, in the **Users** section, click **New -> User**.
- In the **Create User** window, specify the following settings and click **OK**:
  - Full name: **AzS-BackupOperator**
  - User UPN logon: **AzS-BackupOperator@azurestack.local**
  - User SamAccountName: **azurestack\AzS-BackupOperator**
  - Password: **Pa55w.rd**
  - Password options: **Other password options -> Password never expires**

##### Task 2: Create a backup share

In this task, you will:

- Create a backup share.

**Note:** In non-lab scenarios, this share would be external to the Azure Stack Hub deployment. You will create it directly on the

Azure Stack Hub host for the simplicity sake.

- Within the Remote Desktop session to **AzS-HOST1**, start File Explorer.
- In File Explorer, create a new folder **C:\Backup**.
- In File Explorer, right-click the **Backup** folder and, in the right-click menu, click **Properties**.
- In the **Backup Properties** window, click the **Sharing** tab and then click **Advanced Sharing**.
- In the **Advanced Sharing** dialog box, click **Share this folder** and then click **Permissions**.
- In the **Permissions for Backup** window, ensure that the **Everyone** entry is selected and then click **Remove**.
- Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **AzS-BackupOperator** and click **OK**.
- Ensure that the **AzS-BackupOperator** entry is selected and click the **Full Control** checkbox in the **Allow** column.
- Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, click **Locations**.
- In the **Locations** dialog box, click the entry representing the local computer (**AzS-HOST1**) and click **OK**.
- In the **Enter the object names to select** text box, type **Administrators** and click **OK**.
- Ensure that the **Administrators** entry is selected, click the **Full Control** checkbox in the **Allow** column, and then click **OK**.
- Back in the **Advanced Sharing** dialog box, click **OK**.
- Back in the **Backup Properties** window, click the **Security** tab and click **Edit**.
- Click **Add**, in the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **AzS-BackupOperator** and click **OK**.
- In the **Permissions for Backup** dialog box, in the list of entries in the **Groups or user names** pane, click **AzS-BackupOperator**, in the **Permissions for AzS-BackupOperator** pane, click **Full Control** in the **Allow** column and then click **OK**.
- Back in the **Backup Properties** window, click **Close**.

### Task 3: Generate an encryption key

In this task, you will:

- Generate an encryption key.

**Note:** All infrastructure backups must be encrypted, so to configure an infrastructure backup you must provide a certificate corresponding to the encryption key pair. You will use Windows PowerShell to generate a key.

- Within the Remote Desktop session to **AzS-HOST1**, start PowerShell 7 as Administrator.
- From the **Administrator: C:\Program Files\PowerShell\7\pwsh.exe** prompt, run the following to generate the encryption key pair and the corresponding certificate:

```
`powershell $cert = New-SelfSignedCertificate -DnsName  
"azsh.contoso.com" -CertStoreLocation 'cert:\LocalMachine\My'
```

```
New-Item -Path 'C:\' -Name 'CertStore' -ItemType 'Directory'
```

```
Export-Certificate -Cert $cert -FilePath  
C:\CertStore\AzSHIBPK.cer ``
```

**Note:** Azure Stack Hub supports self-signed certificate for the purpose of infrastructure backup. Keep in mind that you need to provide the private key during recovery, so in production environments, make sure to store it in a secure location.

## Task 4: Configure backup controller

In this task, you will:

- Configure backup controller
- Within the Remote Desktop session to **AzS-HOST1**, open the web browser window displaying the [Azure Stack Hub administrator portal](#) and sign in as CloudAdmin@azurestack.local.
- In the Azure Stack Hub administrator portal, click **All services**.
- On the **All services** blade, select **Administration** and then select **Infrastructure backup**.
- On the **Infrastructure backup** blade, click **Configure**.
- On the **Backup controller settings** blade, specify the following settings and click **OK**:

- Backup storage location: **\AzS-HOST1.azurestack.local\Backup**
  - Username: **AzS-BackupOperator@azurestack.local**
  - Password: **Pa55w.rd**
  - Confirm password: **Pa55w.rd**
  - Backup frequency in hours: **12**
  - Retention period in days: **7**
  - Certificate .cer file: **C:\CertStore\AzSHIBPK.cer**
- To verify that the infrastructure backup has been enabled, refresh the web browser page displaying the Azure Stack Hub administrator portal and navigate back to the **Infrastructure backup** blade.
  - On the **Infrastructure backup** blade, review the backup settings.
  - Optionally, click **Backup now** to initiate on-demand backup.

**Note:** The backup process takes more than 15 minutes to complete and cannot be paused or stopped.

**Review:** In this exercise, you have created a share that will host Azure Stack Hub infrastructure backups and an Active Directory user account that will be used by the Azure Stack Hub infrastructure backup to connect to that share, generated encryption keys, and configured backup controller.



# Table of Contents

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.	7
{{ activity.lab.title }}{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %}	1
{{ activity.demo.title }}	1