Module 03 - Azure Automation

Pre-requisites

1. [] Sign into the virtual machine by clicking on it and providing the password (Passw0rd)
2. [] Right click on the CopyStudentFiles.ps1 on the desktop and select **Run With PowerShell**
3. [] Press **A** to Agree to All downloads.
4. [] The student lab files and slides will be downloaded to C:\Labs and C:\Slides

Exercise 1: Create an Azure Subscription

Introduction

Use this lab time to create your Microsoft Azure subscription using an Azure Pass or Free Trial if you do not have an Azure Pass provided.

Prerequisites (if applicable)

- Azure Pass code
- PowerShell ISE

Estimated Time to Complete This Lab

20 minutes

Task 1: Create an Azure Pass Subscription

> [!note] The easiest way to create a subscription if you currently have an Azure Subscription tied to your Microsoft Hotmail or Microsoft Outlook account is to create a new Outlook account.

1. [] Go through the steps at: **https://www.microsoftazurepass.com/Home/HowTo**

Exercise 2: Create an Azure Resource Group

Introduction

In this lab, we will provision a new resource group account in our Azure Subscription.

A resource group is a logical container to hold a collection of resources in Azure. All objects in Azure must belong to a resource group including Azure Automation Accounts.

Prerequisites (if applicable)

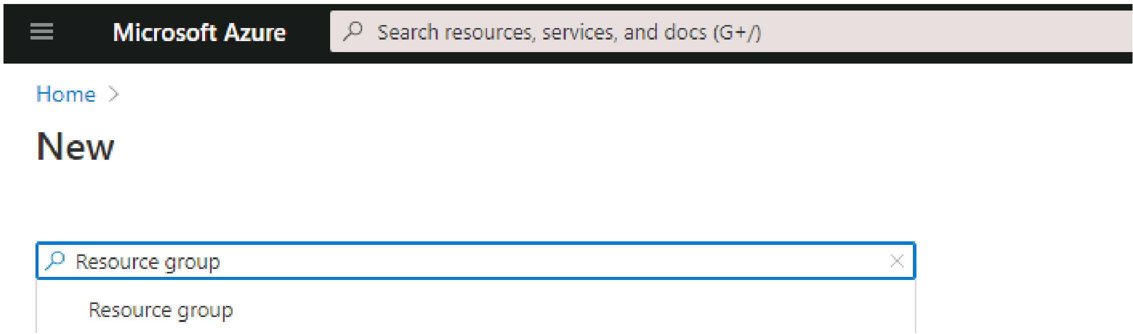
Azure Account

Estimated Time to Complete This Lab

20 minutes

Task Description

1. [] In the Azure Portal click **Create a resource** and search for **Resource Group**

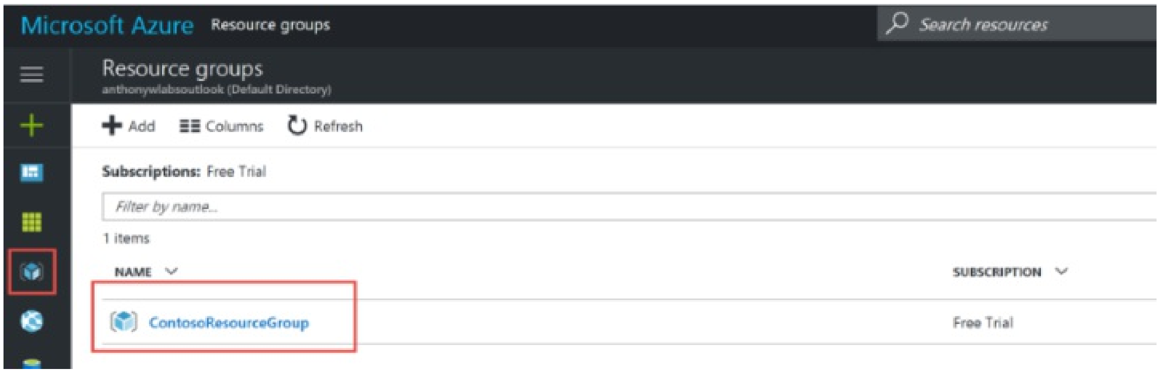


{400}

2. [] Select **Resource Group** and in the new blade select **Create**

> [!note] The required fields for a resource group including resource group name, location and subscription.

3. [] Enter the following details in the blade.
    1. **Resource Group Name**: ContosoResourceGroup
    2. **Subscription**: Ensure your subscription is selected.
    3. **Location**: Choose a location nearest to you
4. [] Click **Review + Create**, then click **Create** and the resource group will be deployed.
5. [] In the Azure Portal select **Resource Groups** from the left-hand blade and verify that the group has been created.



{400}

Exercise 3: Create an Azure Automation Account

Introduction

In this lab, we will provision a new automation account in our Azure Subscription.

An Azure Automation account is required to store runbooks, assets, DSC configurations as well as to launch and monitor jobs.

Prerequisites (if applicable)

N/A

Estimated Time to Complete This Lab

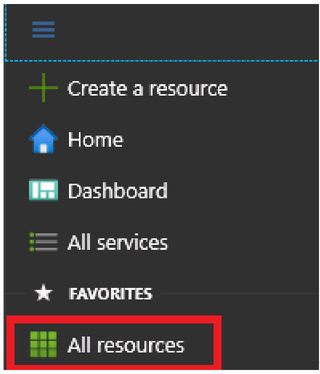
15 minutes

Task Description

1. [] In the search bar at the top of the portal page enter **automation**.
2. [] Select **Automation Accounts**.
3. [] Click on **Create automation account** and enter details as below.
    1. **Subscription**: Ensure your subscription is selected
    2. **ResourceGroup:** Select **Use Existing** and select the **ContosoResourceGroup**
    3. **Name**: ContosoAutomationAccount

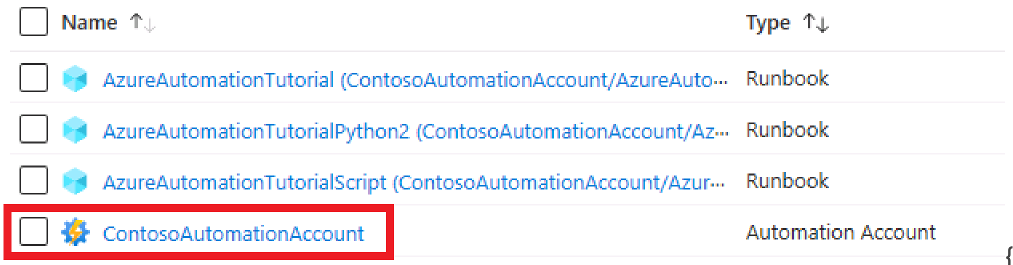
> Make sure you get a green check mark.

1. **Location:** Enter the region of your resource group for the automation account region.
    1. Click **Next**.
2. [] In the advanced tab for managed identities, ensure that **System assigned** check box is checked. Click **Review + Create**.
3. [] Validate the automation account deployment details and click **Create**.
4. [] When complete - in the main dashboard click on **All Resources**



{400}

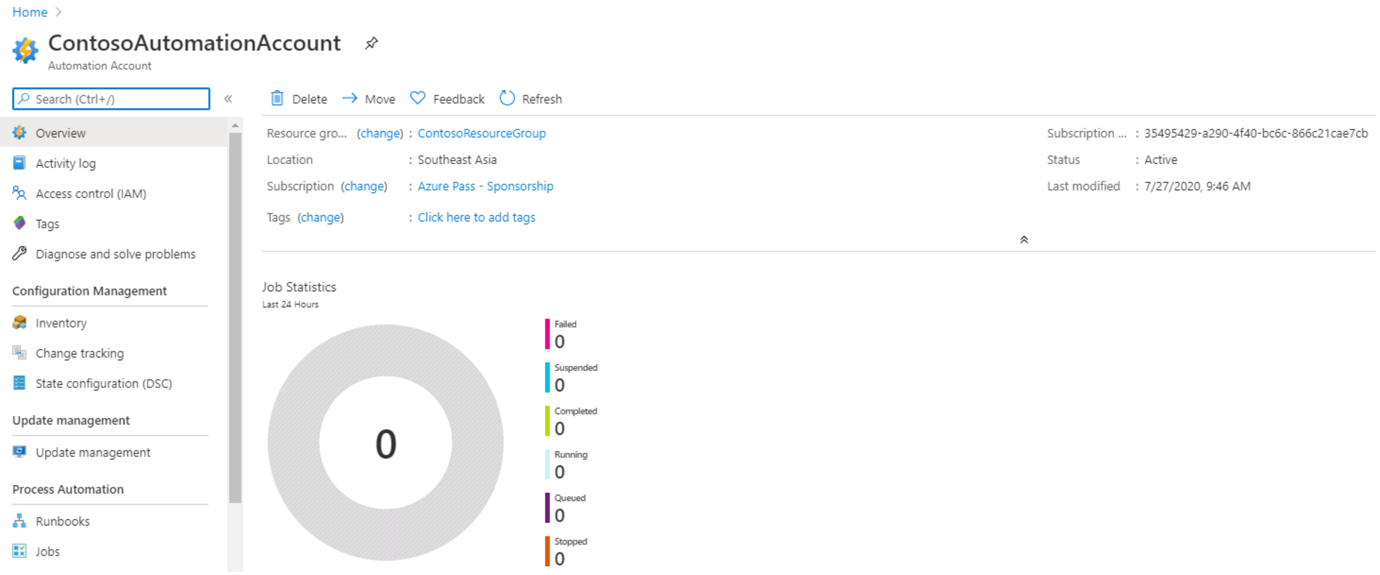
5. [] Click on **ContosoAutomationAccount**



□	Name ↑↓	Type ↑↓
□	AzureAutomationTutorial (ContosoAutomationAccount/AzureAuto...	Runbook
□	AzureAutomationTutorialPython2 (ContosoAutomationAccount/Az...	Runbook
□	AzureAutomationTutorialScript (ContosoAutomationAccount/Azur...	Runbook
□	ContosoAutomationAccount	Automation Account

{400}

6. [] Your automation account has been created.



Resource gro... (change) : ContosoResourceGroup
Location : Southeast Asia
Subscription (change) : Azure Pass - Sponsorship
Tags (change) : Click here to add tags
Subscription ... : 35495429-a290-4f40-bc6c-866c21cae7cb
Status : Active
Last modified : 7/27/2020, 9:46 AM

{400}

Exercise 4: Create a PowerShell Runbook

Introduction

In this lab, we will create a scripted PowerShell Runbook

Runbooks are script files which can be started manually, triggered on a schedule, triggered by a webhook or called via Windows PowerShell. They can be run in Azure or on-premise.

Prerequisites (if applicable)

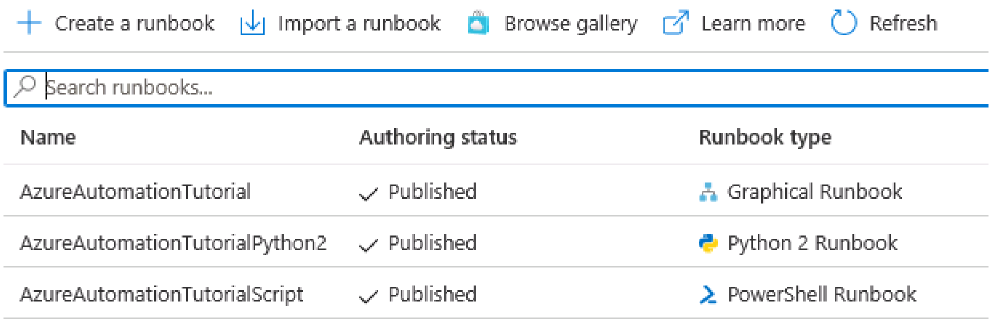
Automation Account created

Estimated Time to Complete This Lab

10 minutes

Task Description

1. [] Open your automation account.
2. [] Click on **Runbooks**



Name	Authoring status	Runbook type
AzureAutomationTutorial	✓ Published	Graphical Runbook
AzureAutomationTutorialPython2	✓ Published	Python 2 Runbook
AzureAutomationTutorialScript	✓ Published	PowerShell Runbook

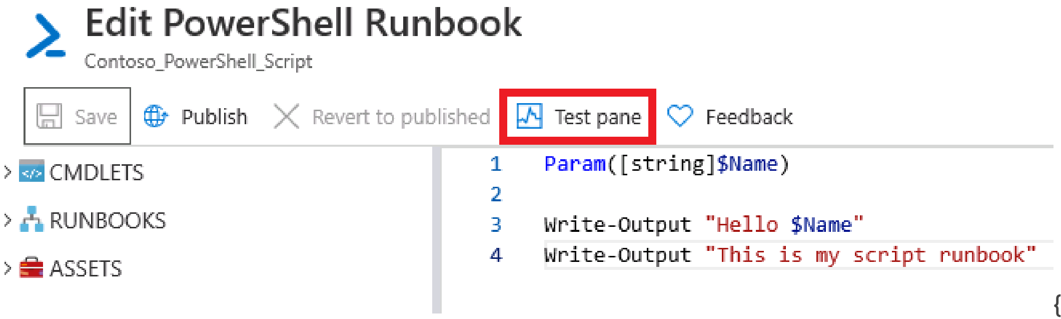
{400}

3. [] Select **Create a runbook** and in the new blade enter the details as below.
    1. **Name**: `Contoso_PowerShell_Script`
    2. **Runbook type**: PowerShell
    3. **Description**: Runs a basic PowerShell script.
4. [] Click **Create** when complete.
5. [] When complete the script pane for the new runbook will open in the browser. Type the following lines into the script pane and click **Save**

```
Param([string]$Name)

Write-Output "Hello $Name"
Write-Output "This is my script runbook"
```

6. [] Click on the **Test pane**

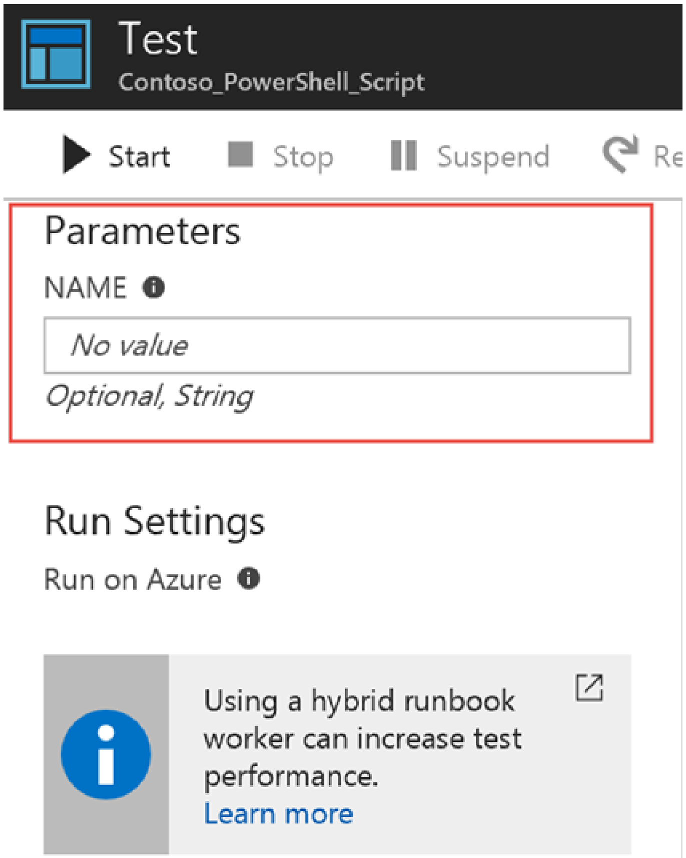


```
Param([string]$Name)

Write-Output "Hello $Name"
Write-Output "This is my script runbook"
```

{400}

7. [] Notice that the parameter from the script now appears as input to the Runbook. Enter your name and click **Start**.



Parameters

NAME

No value

Optional, String

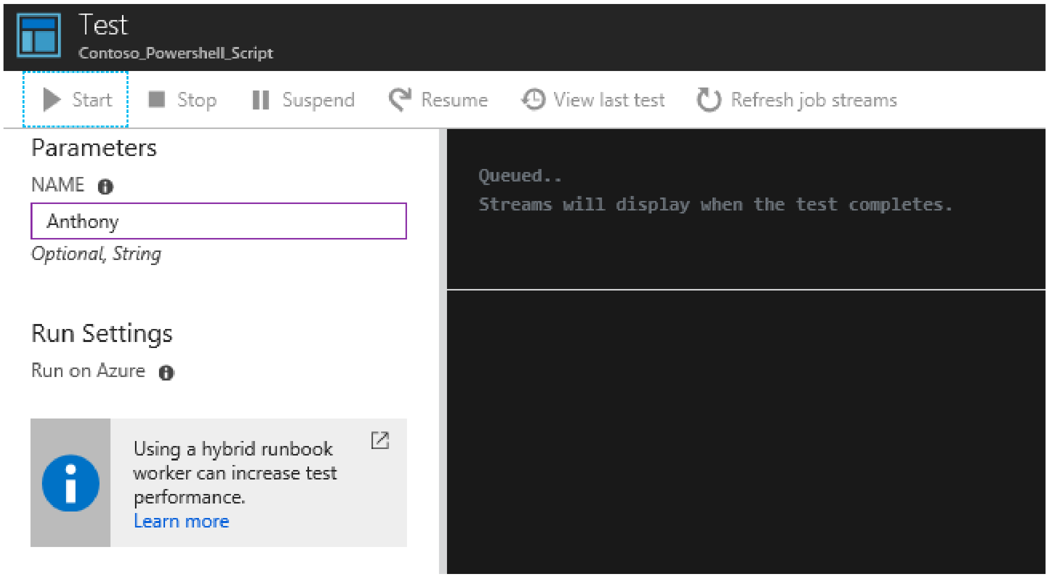
Run Settings

Run on Azure

Using a hybrid runbook worker can increase test performance.
Learn more

{400}

8. [] The runbook is queued in the Azure Automation service until a worker picks up the job.



Parameters

NAME

Anthony

Optional, String

Run Settings

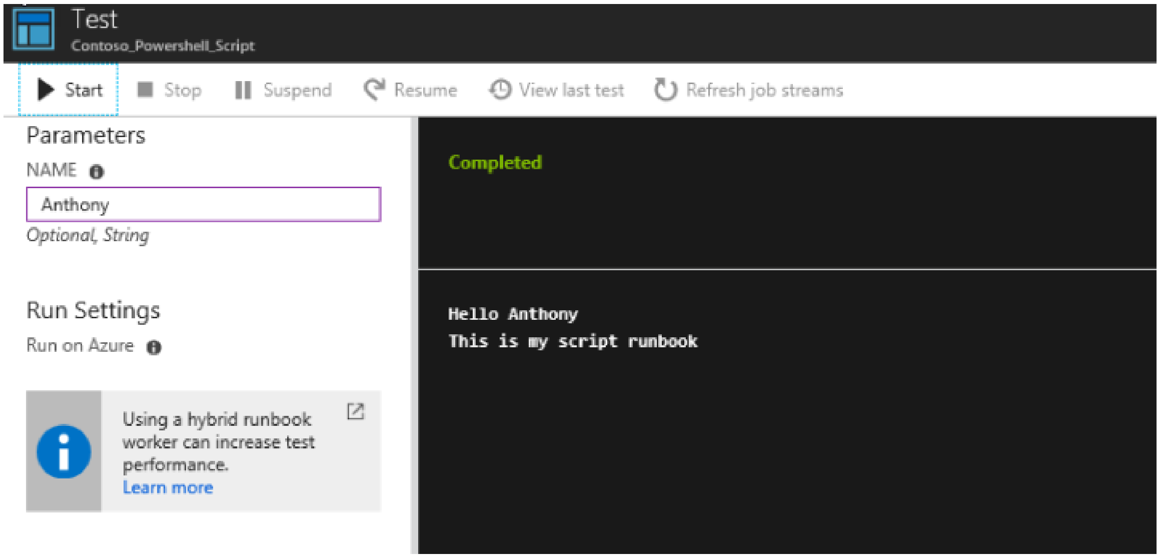
Run on Azure

Using a hybrid runbook worker can increase test performance.
Learn more

```
Queued..
Streams will display when the test completes.
```

{400}

9. [] Review the output in the console pane once complete.



Test

Contoso_Powershell_Script

Parameters

NAME: Anthony

Optional, String

Run Settings

Run on Azure

Using a hybrid runbook worker can increase test performance. Learn more

Completed

```
Hello Anthony
This is my script runbook
```

{400}

10. [] Close the test pane and click **Publish**. Select **Yes** to override the previous version.

Exercise 5: Import a Gallery Runbook

Introduction

In this lab, we will import a runbook from the gallery

The PowerShell gallery contains several runbooks which other people have authored. You can load these into your account and modify them for your environment.

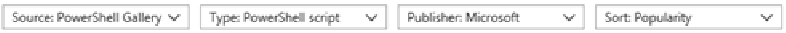
Prerequisites (if applicable)

Estimated Time to Complete This Lab

10 minutes

Task Description

1. [] Open your automation account.
2. [] Select **Runbooks**
3. [] Click **Browse Gallery**
4. [] Adjust the filters as in the image below and click **OK**.



Source: PowerShell Gallery

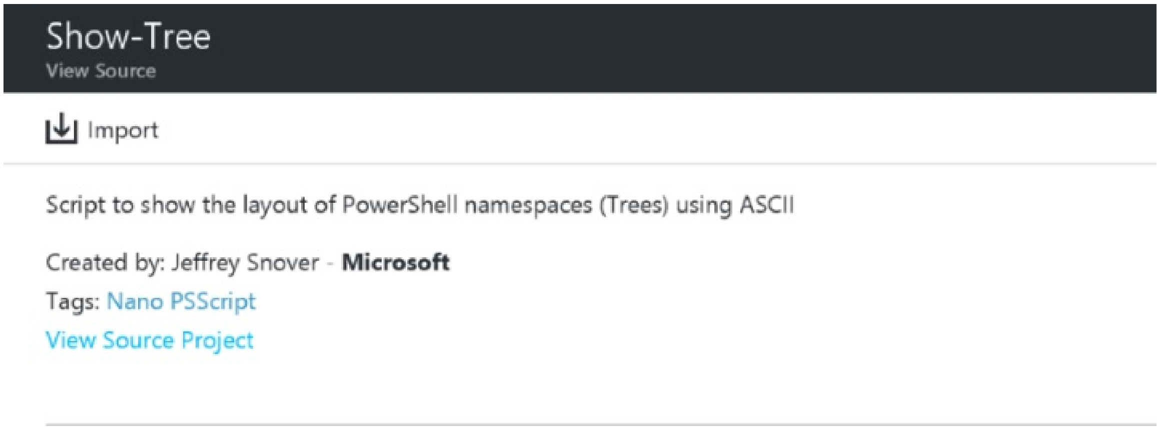
Type: PowerShell script

Publisher: Microsoft

Sort: Popularity

{400}

5. [] Review the runbooks which are presented. Clicking them will open a new blade which has more information about the runbook including the sample code.
6. [] Select any runbook and click **Import**.



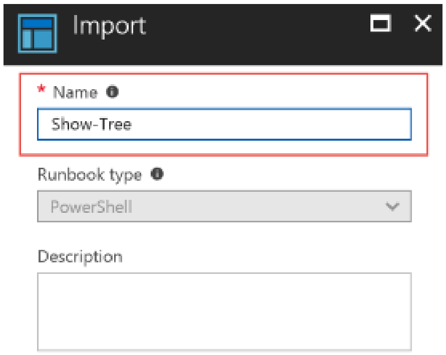
Show-Tree
View Source

Script to show the layout of PowerShell namespaces (Trees) using ASCII

Created by: Jeffrey Snover - **Microsoft**
Tags: Nano PSScript
View Source Project

{400}

7. [] In the new blade, you can rename the Runbook. Just click **OK**.



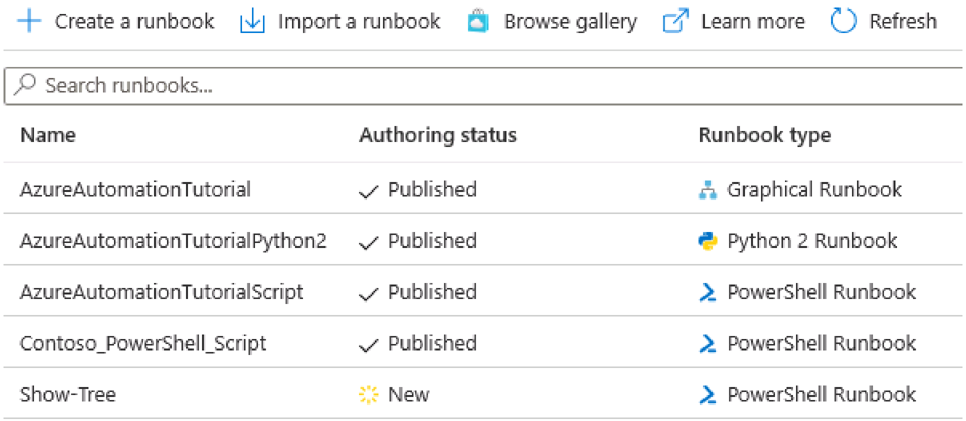
* Name: Show-Tree

Runbook type: PowerShell

Description:

{400}

8. [] Close the gallery and verify that the Runbook has been imported into your automation account.



Name	Authoring status	Runbook type
AzureAutomationTutorial	✓ Published	Graphical Runbook
AzureAutomationTutorialPython2	✓ Published	Python 2 Runbook
AzureAutomationTutorialScript	✓ Published	PowerShell Runbook
Contoso_PowerShell_Script	✓ Published	PowerShell Runbook
Show-Tree	New	PowerShell Runbook

{400}

Exercise 6: Creating Automation Account Assets

Introduction

In this exercise, we will create variable and credential assets which can be used later.

Variables and credentials can be created in an Automation Account and accessed from Runbooks or DSC.

Prerequisites (if applicable)

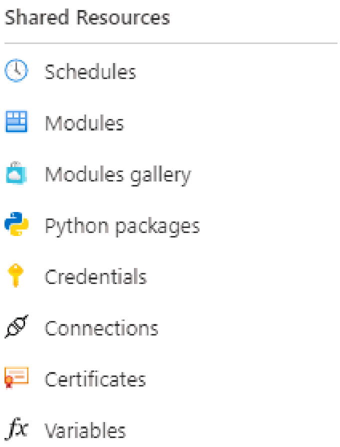
N/A

Estimated Time to Complete This Lab

15 minutes

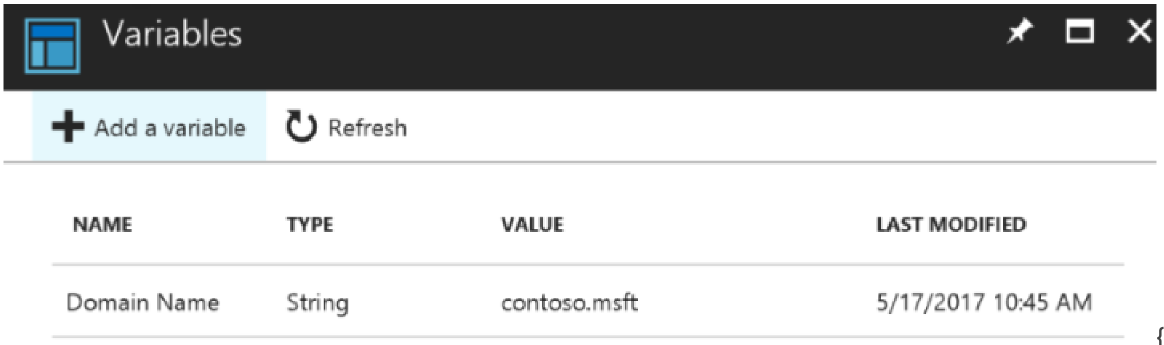
Task Description

1. [] Open your automation account.
2. [] Under Shared Resources, Click **Variables**



{400}

3. [] Click **Add a Variable**.
4. [] Review the data types available and note that a variable can be encrypted. Enter the following value in the blade and click **Create**.
    1. **Name**: Domain Name
    2. **Type**: String
    3. **Value**: contoso.msft
    4. **Encrypted**: No
5. [] Refresh the variables blade and ensure the variable has been created.

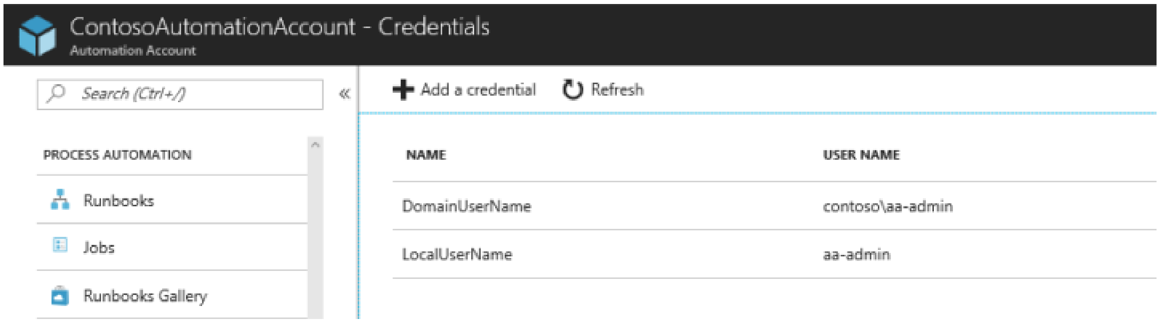


NAME	TYPE	VALUE	LAST MODIFIED
Domain Name	String	contoso.msft	5/17/2017 10:45 AM

{400}

6. [] Open the **Credentials** under SHARED RESOURCES and click **Add a Credential**. Review the field available for a credential object. We will create two credential objects for use in a later lab. Enter the following details and click **Create**.
    1. **Name**: LocalUserName
    2. **User Name**: aa-admin
    3. **Password**: R3dDwarf2017
    4. **Confirm Password**: R3dDwarf2017

7. [] Refresh the credentials blade and verify the object has been created. Add another credential object with the values below.
    1. **Name**: DomainUserName
    2. **User Name**: contoso\aa-admin
    3. **Password**: R3dDwarf2017
    4. **Confirm Password**: R3dDwarf2017
8. [] Refresh the credentials blade and verify the object has been created.



NAME	USER NAME
DomainUserName	contoso\aa-admin
LocalUserName	aa-admin

{400}