# Module 11: Implementing Azure-based management and automation

# Lab: Implementing Automation

### Scenario

A. Datum Corporation wishes to minimize administrative overhead as much as possible, especially for tasks such as deploying and deprovisioning VMs. For this reason, as part of A. Datum's evaluation of Microsoft Azure, you have been asked to test the new Azure Automation features and, as part of your tests, to deploy Azure VMs by using runbook automation.

### Objectives

After completing this lab, you will be able to:

* Configure Automation accounts.
* Create runbooks.

### Lab Setup

Estimated Time: 40 minutes

Virtual Machine: **20533C-MIA-CL1**

User Name: **Student**

Password: **Pa$$w0rd**

Before starting this lab, ensure that you have performed the "Preparing the Azure environment" demonstration tasks at the beginning of the first lesson in this module, and that the setup script has completed. However, before running the setup script, first identify the Azure regions where the students can create Azure Automation accounts. Make sure to choose one of these regions when prompted during execution of the setup script.

To identify the Azure region that students can use for this lab, sign-in to the Azure portal. In the hub menu of the Azure portal, click New, and then click Management. In the Management blade, click Automation. In the Add Automation Account blade, click the down arrow on the right hand side of the Location drop down list. The drop down list will contain the list of Azure regions where you can provision an Automation account. Choose the region closest to the lab location and provide it to students.

## Exercise 1: Configuring Automation accounts

### Scenario

Administrators at A. Datum Corporation spend considerable time creating storage accounts and Azure VMs. You want to increase administrator productivity by using Automation to execute these tasks and freeing administrators to continue with other tasks.

The main tasks for this exercise are as follows:

1. Create an Automation account
2. Create an Azure AD user
3. Create Automation assets

#### Task 1: Create an Automation account

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa$$w0rd**, and that the setup script you ran in the previous demonstration to prepare the environment has completed.
2. Sign in to the Azure portal by using the Microsoft account that is the Service Administrator or Co-Administrator of your Azure subscription.
3. Create a new Azure Automation account with the following settings:

* Name: **LabAutomationAccount**
* Subscription: your current subscription
* Resource group: create a new resource group named **AutomationLabRG**
* Location: the Azure region that you chose when running the provisioning script
* Create Azure Run As account: No
* Pin to dashboard: Enable the checkbox

1. Wait for the Automation account to be provisioned. This should take less than a minute.

#### Task 2: Create an Azure AD user

1. From the Azure classic portal, create a new user in the default directory of your Azure subscription, with the following settings:

* USER NAME: **LabAutomationUser**
* @: leave the default value
* ROLE: **user**
* Multi Factor Authentication: disabled
* DISPLAY NAME: **Lab Automation User**

1. From the Azure classic portal, configure the new user as a Co-Administrator of the current subscription.
2. Change the temporary password of the newly created user to **Pa$$w0rd**.

#### Task 3: Create Automation assets

1. In the Azure portal, in the **Automation Account** blade, add a Credential asset:

* Name: **PSCredential**
* Description: **Lab Automation User (Co-Administrator)**
* User name: the name of the newly created AutomationUser account that you copied to Notepad
* Password: **Pa$$w0rd**
* Confirm password: **Pa$$w0rd**

1. In the same Automation account, create the following Automation unencrypted Variable assets of the String type:

* SubscriptionName: name of your subscription
* AdminName: **Student**
* AdminPassword: **Pa$$w0rd**
* Location: *the name of Azure* *region* *that* *you used when running the provisioning script at the beginning of this module*
* Network: **ADATUM-HQ-VNET**
* Subnet: **Subnet-1**

1. In the same Automation account, create the following Schedule asset:

* Name: **EndOfDay**
* Description: **End of Day**
* Starts: tomorrow's date at **18:00:00**
* Recurrence: **Recurring**
* Runs every : **1 Day**
* Set expiration: **No**

**Result**: After completing this exercise, you should have configured a new Microsoft Azure Automation account, and created a new Microsoft Azure Active Directory (Azure AD) organizational account to use as an Automation Credential asset.

## Exercise 2: Creating runbooks

### Scenario

As part of your tests of the new Azure Automation features, you will now deploy Azure virtual machines by using an Automation runbook.

The main tasks for this exercise are as follows:

1. Import a runbook
2. Publish and execute a runbook
3. Reset the environment

#### Task 1: Import a runbook

1. From the Azure portal, import the PowerShell workflow script **New-StorageVNetAndVMs.ps1** residing in the D:11folder into your Automation account.
2. Review the content of the runbook.

#### Task 2: Publish and execute a runbook

1. Publish the **New-StorageAndVMs** runbook.
2. Start the newly published runbook.
3. View the progress of the runbook execution. Wait until the job completes.

#### Task 3: Reset the environment

1. Launch Windows PowerShell as **Administrator**.
2. From the Windows PowerShell prompt, run:

Reset-Azure

1. When prompted (twice), sign in by using the Microsoft account associated with your Azure subscription.
2. If you have multiple Azure subscriptions, select the one you want to target with the script.
3. When prompted for confirmation, type **y**.

**Note:** This script will remove Azure services in your subscription. We therefore recommend that you use an Azure trial pass that was provisioned specifically for this course, and not your own Azure account. The script will take 5-10 minutes to reset your Microsoft Azure environment to be ready for the next lab. The script removes all storage, VMs, virtual networks, cloud services, and resource groups. **Important**: The script might not be able to get exclusive access to a storage account to delete it (if this occurs, you will see an error). If you find remaining objects after the reset script is complete, you can rerun the **Reset-Azure** script, or use the Azure portal and Azure classic portal to delete all the objects in your Azure subscription manually â€”with the exception of the default director.

**Result**: After completing this exercise, you should have imported, published, and executed a PowerShell workflow-based runbook that deploys two virtual machines in parallel.

**Question** Why did you have to create an Azure AD account in the lab?

**Question** What should you consider when testing the execution of an Automation runbook?

©2016 Microsoft Corporation. All rights reserved.

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

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