

|  |
| --- |
| September 16, 2015 |
| Version 1.0 |
|  |
| Neil Mackenzie, Cloud Solution Architect |
| Michael S. Collier, Cloud Solution Architect |
| Alex Belotserkovskiy, Technical Evangelist |

Azure Automation:

Scheduled Shutdown of Azure Virtual Machines

Table of Contents

1. [Change Log 2](#_Toc430131015)
2. [Executive Summary 2](#_Toc430131016)
3. [Prerequisites 3](#_Toc430131017)
4. [Configure Azure Automation 4](#_Toc430131018)

[Create an Azure Automation Account](#_Toc430131019)

[Configuration Authentication](#_Toc430131020)

[Create an Azure Connection Asset](#_Toc430131021)

[Import and Publish a Connect-Azure Runbook](#_Toc430131022)

1. [Create Your Runbook 16](#_Toc430131023)
2. [Test the Runbook 20](#_Toc430131024)
3. [Publish the Runbook 21](#_Toc430131025)
4. [Link the Runbook to a Schedule 22](#_Toc430131026)
5. [Contacts 24](#_Toc430131027)
6. [Appendix 25](#_Toc430131028)

[Appendix A – Configure via PowerShell](#_Toc430131029)

# Change Log

|  |  |  |
| --- | --- | --- |
| Date | Author | Comments |
| 09/16/2015 | Michael S. Collier  Neil Mackenzie  Alex Belotserkovskiy | Initial version. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Executive Summary

The purpose of this document is to outline an approach for using Azure Automation to schedule the shutdown (deallocation) of all Azure Virtual Machines within a specified Azure subscription. Shutting down unused Azure Virtual Machines is an important activity in controlling (reducing) the cost associated with running services in the Microsoft Azure platform. The sections below walkthrough the steps necessary to create an Azure Automation account, configure it to authenticate to an Azure subscription, iterate over all Azure Virtual Machines in the subscription, and shut down (deallocate) each VM instance.

The guidance contained within this document is expected to evolve as capabilities within the Azure continue to evolve.

Credit and extreme gratitude should be directed towards Keith Mayer for creating the [original blog post](http://blogs.technet.com/b/keithmayer/archive/2014/04/06/step-by-step-getting-started-with-windows-azure-automation.aspx) that inspired this guidance. The majority of content contained in the following pages are a direct result of Keith’s post.

# Prerequisites

A Microsoft Azure subscription is required for all activities.

The process outlined below contains the following restrictions:

* The Azure Management Portal (<http://manage.windowsazure.com>) is used in this document.
* Only Azure Virtual Machines created using the Azure Service Management (ASM) / RDFE API are applicable to process listed below (i.e. only IaaS v1 virtual machines can be managed, not IaaS v2 / Azure Resource Manager (ARM) virtual machines).

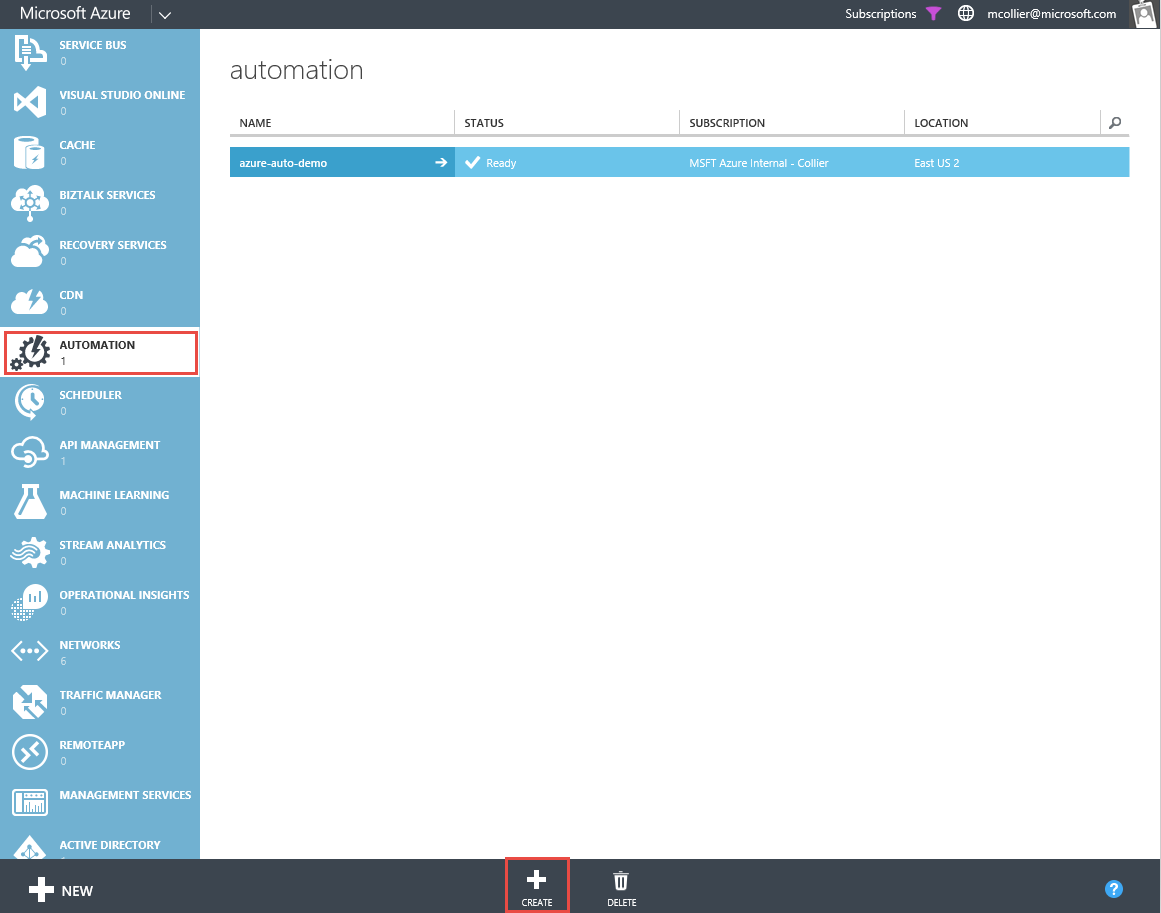
A future version of the document will be updated to include the Azure Preview Portal (<http://portal.azure.com>), as well as support for Azure Virtual Machines created with the Azure Resource Manager (ARM) stack.

# Configure Azure Automation

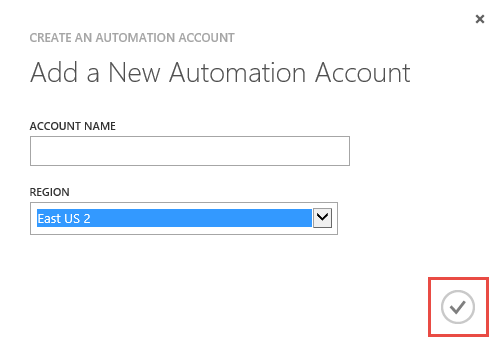
Azure Automation is used to execute a small PowerShell script that will stop (deallocate) all Azure Virtual Machines in a specific Azure subscription. If an Azure Automation account has not been created, the first step is to create a new account.

## Create an Azure Automation Account

1. Sign-in at the Azure Management Portal and scroll down the left blue navigation bar and click **Automation**.



1. Click the **Create** button on the bottom black toolbar to define a new Azure Automation account for managing Runbooks. After specifying an *Account Name* and *Region*, click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to create the account.



## Configuration Authentication

There are two ways within Azure Autoamtion to configure authentication to a Microsoft Azure subscription:

There are two ways within Azure Automation to configure authentication to a Microsoft Azure subscription:

1. Azure Active Directory (Azure AD)

The use of Azure AD currently requires Multi-Factor Authenticaiton (MFA), per corporate Microsoft policy, when using with Microsoft work or school accounts (e.g. [alias@microsoft.com](mailto:alias@microsoft.com)). As such, Microsoft corporate accounts will not work as a means of authentication with Azure Automation. When available, this guidance will be updated with the recommended approach for Microsoft corporate accounts (or suitable alternative).

The use of Azure AD currently requires Multi-Factor Authentication (MFA), per corporate Microsoft policy, when using with Microsoft work or school accounts (e.g. ). As such, Microsoft corporate accounts will not work as a means of authentication with Azure Automation. When available, this guidance will be updated with the recommended approach for Microsoft corporate accounts (or suitable alternative).

The use of Azure AD currently requires Multi-Factor Authentication (MFA), per corporate Microsoft policy, when using with Microsoft work or school accounts (e.g. ). As such, Microsoft corporate accounts will not work as a means of authentication with Azure Automation. When available, this guidance will be updated with the recommended approach for Microsoft corporate accounts (or suitable alternative).

$thumbprint = (New-SelfSignedCertificate -DnsName "AzureAutomation" -CertStoreLocation Cert:\CurrentUser\My -KeySpec KeyExchange).Thumbprint

$cert = (Get-ChildItem -Path cert:\CurrentUser\My\$thumbprint)

Export-Certificate -Cert $cert -FilePath "c:\AzureAutomation.cer" -Type CERT

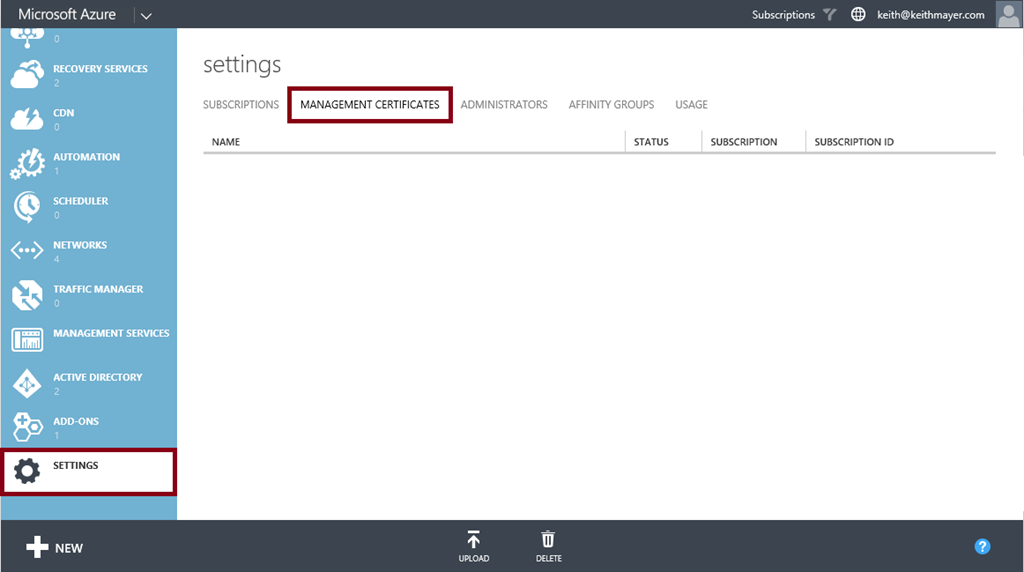
$password = Read-Host -AsSecureString

Export-PfxCertificate -Cert $cert -FilePath "c:\AzureAutomation.pfx" -Password $password

#### Upload the Management Certificate to Microsoft Azure

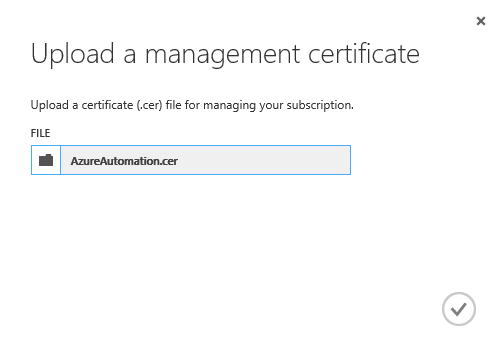
Now that you have a new management certificate created and exported, you’ll need to upload the exported .CER file to your Microsoft Azure subscription.

1. Sign in at the [Microsoft Azure Management Portal](http://manage.windowsazure.com/) with the logon credentials used when you activated your Microsoft Azure subscription.
2. Select **Settings** located on the side navigation panel on the Microsoft Azure Management Portal page. You may need to scroll down the side navigation panel to see this selection.



1. On the *Settings* page, click on the **Management Certificates** tab.

1. On the *Management Certificates* page, click on the **Upload** button located on the bottom black toolbar. When prompted, browse to **c:\AzureAutomation.cer** and click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button.



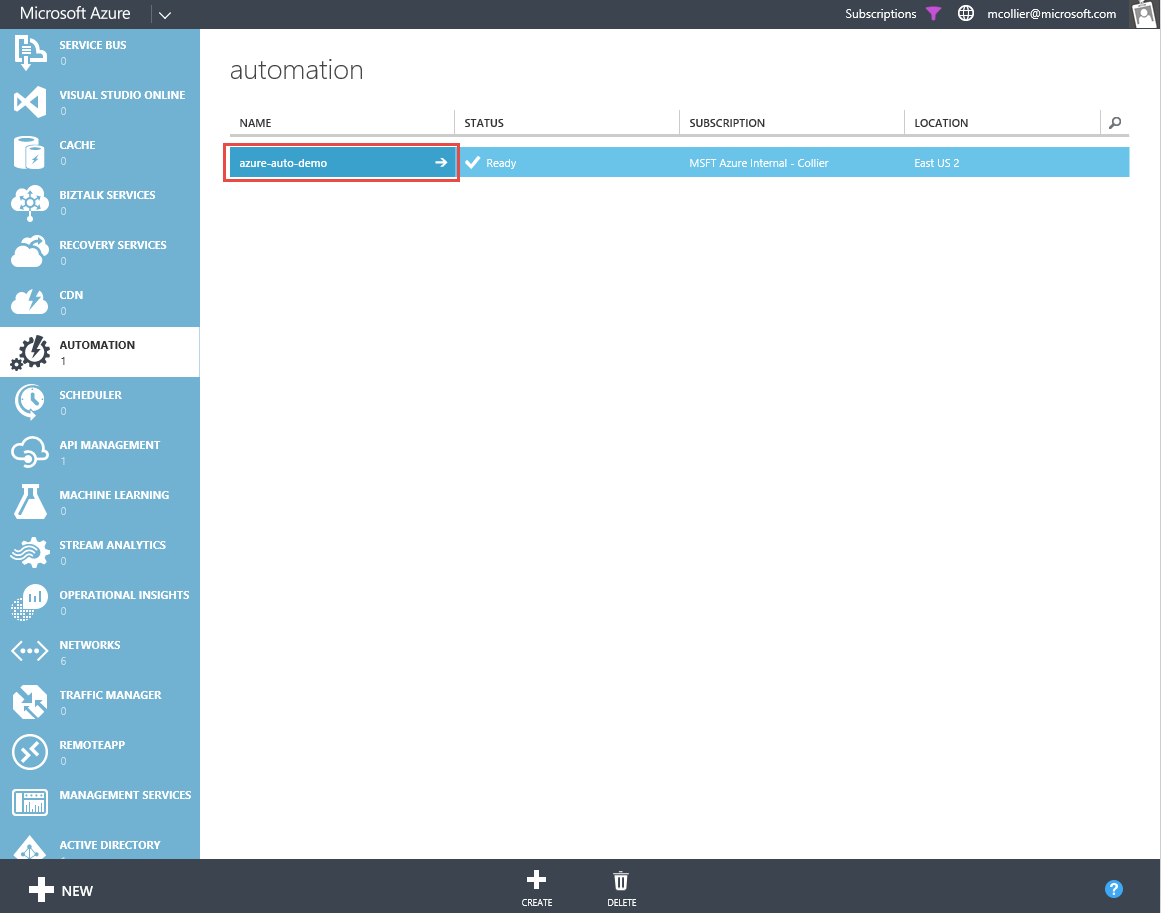
After uploading, your new certificate should now appear in the list of management certificates.

1. For your newly uploaded certificate, record the values listed in the **Subscription** and **Subscription ID** columns for later use in this Step-by-Step article.

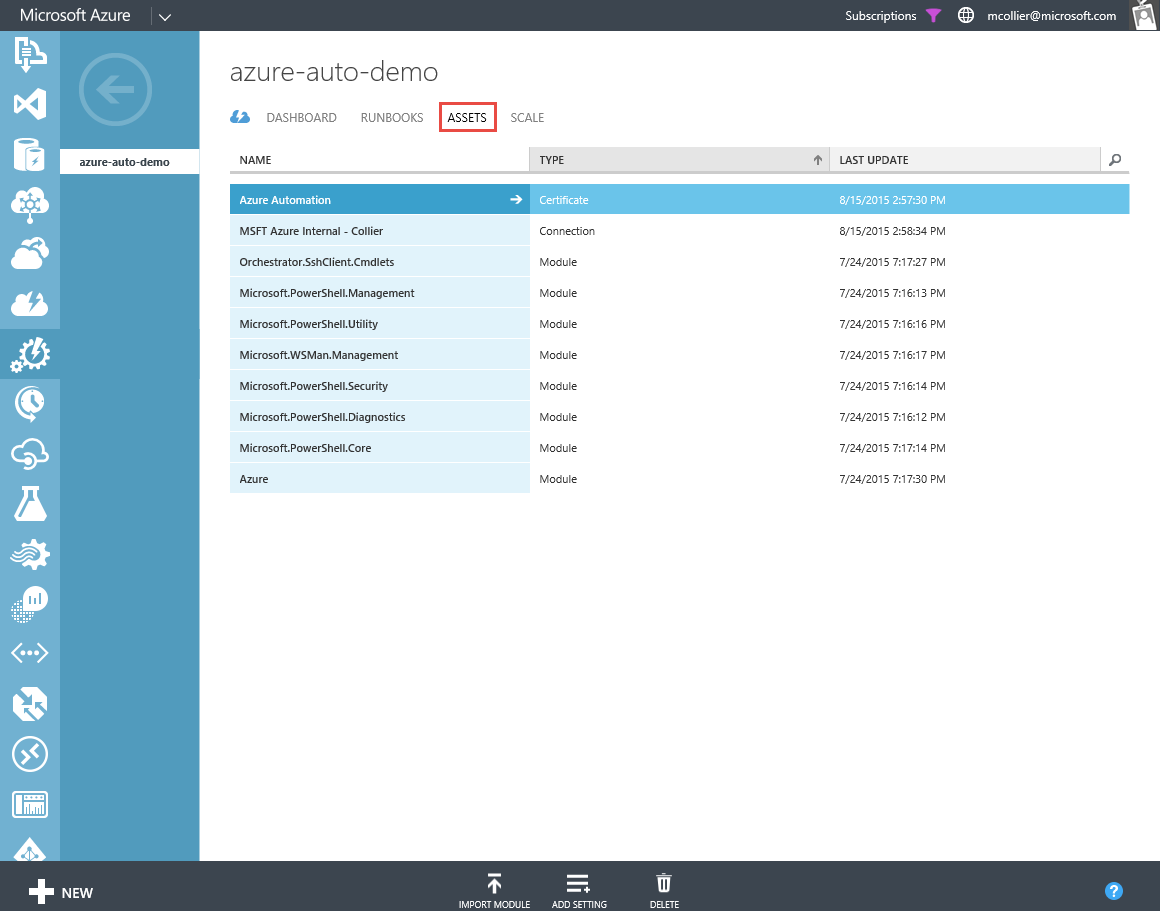
#### Create a Management Certificate Asset

For your Azure Automation account to be able to authenticate to your Microsoft Azure subscription, you’ll also need to upload the certificate .PFX file.  You’ll upload this certificate as an Asset in your Azure Automation account so that it can be consistently leveraged across multiple runbooks.

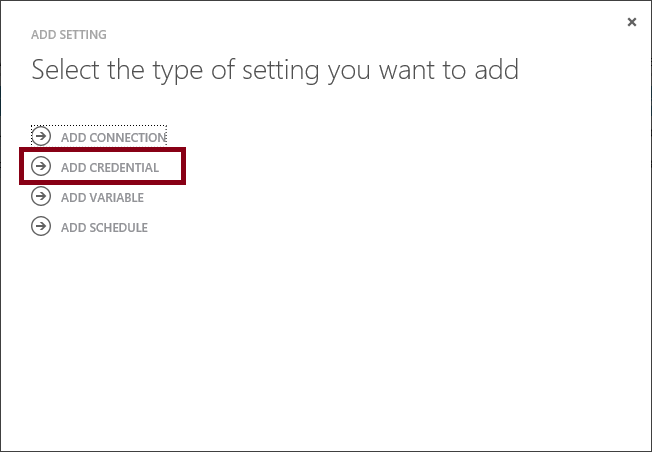
1. On the Automation page in the Azure Management Portal, click on the name on the new Azure Automation account to drill into the account properties.



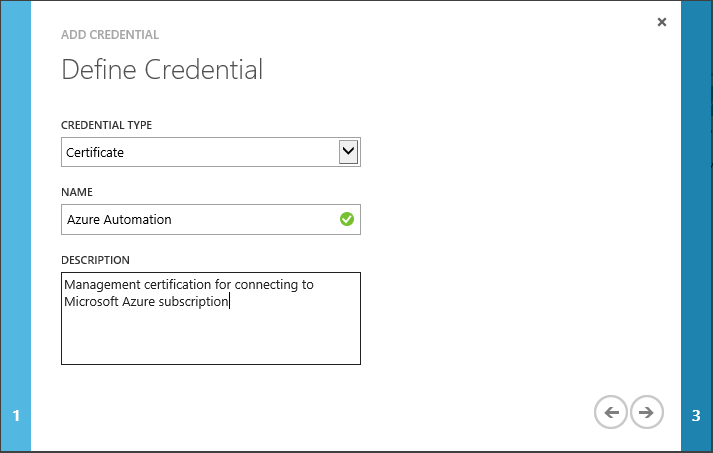
1. Click on the **Assets** tab.



1. Click the **Add Setting** button on the bottom black toolbar.  When prompted, select **Add Credential.**

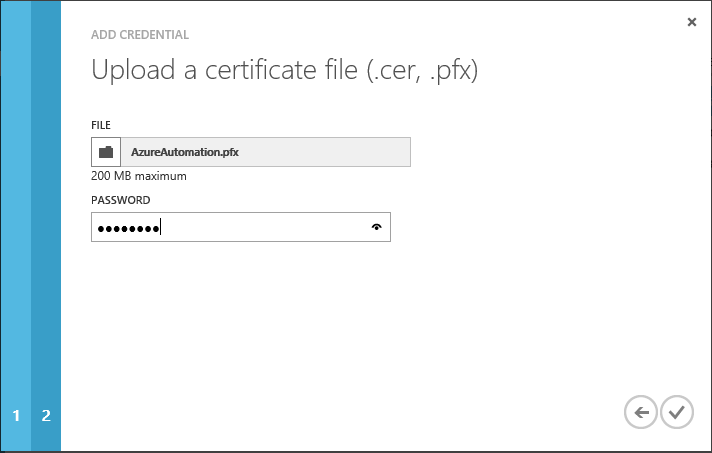


1. On the Define Credential page, select **Certificate** in the Credential Type list and enter a **name** for this new credential.



Click the [image](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/image_5F00_11F7FE3F.png) button to continue.

1. On the Upload a certificate file page, browse to **C:\AzureAutomation.pfx** in the File field and enter the **Password** used to protect the private key when previously exporting this file.



Click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to upload this certificate to your Azure Automation account.

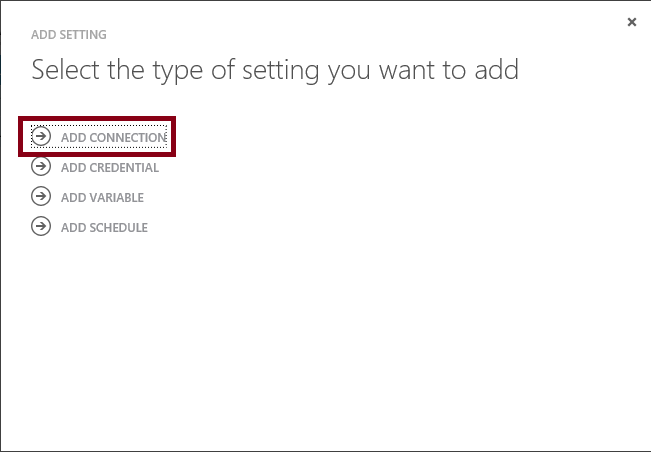
### Configure Azure Active Directory

|  |  |
| --- | --- |
|  | The preferred way to connect to Microsoft Azure is no longer management certifcates but instead Azure Active Directory (Azure AD) authentication.  The use of Azure AD currently requires Multi-Factor Authenticaiton (MFA), per corporate Microsoft policy, when using with Microsoft work or school accounts (e.g. alias@microsoft.com). As such, Microsoft corporate accounts will not work as a means of authentication with Azure Automation. When available, this guidance will be updated with the recommended approach for Microsoft corporate accounts (or suitable alternative). |

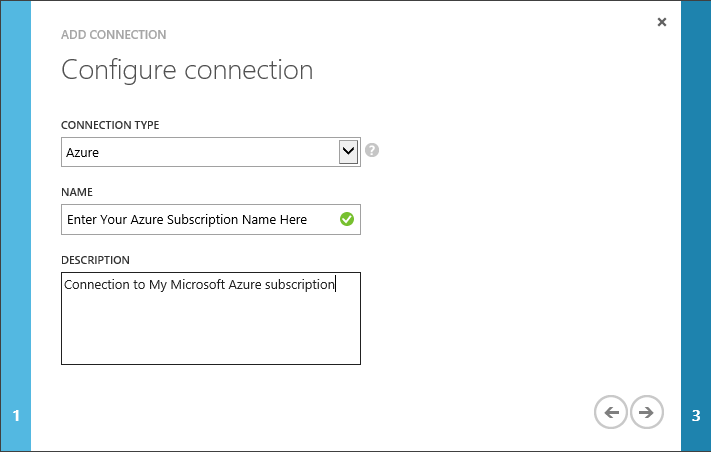
## Create an Azure Connection Asset

You can also define the connection information for your Microsoft Azure subscription as an Asset in your Azure Automation account.  Doing so allows you to easily relate your Microsoft Azure subscription name, subscription ID and management certificate together as a centralized definition for use in all of your runbooks.

1. On the Assets tab for your Azure Automation account, click the **Add Setting** button on the bottom toolbar.  When prompted, click **Add Connection.**



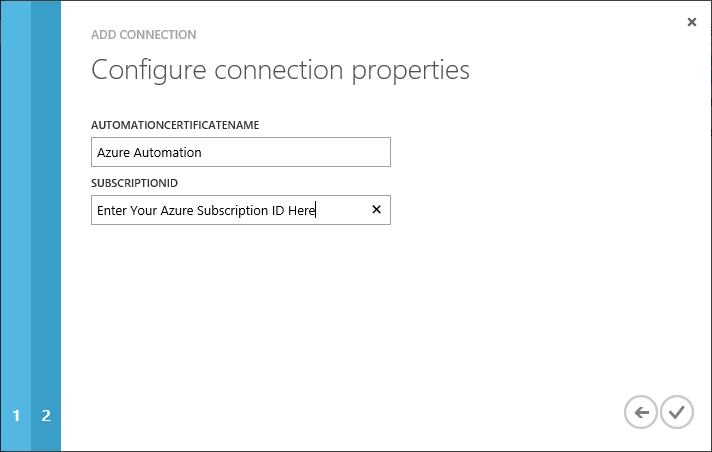
1. On the Configure connection page, select **Azure** as the Connection Type and enter a **Name** that matches your **Microsoft Azure subscription name** recorded earlier.



**Connection Type:** Azure  
**Name:** Enter the name of your Microsoft Azure subscription recorded earlier.

Click the [image](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/image_5F00_11F7FE3F.png) button to continue

1. On the Configure connection properties page, enter the name of the **management certificate asset** that you previously uploaded and enter your **Microsoft Azure subscription ID** that was recorded earlier.



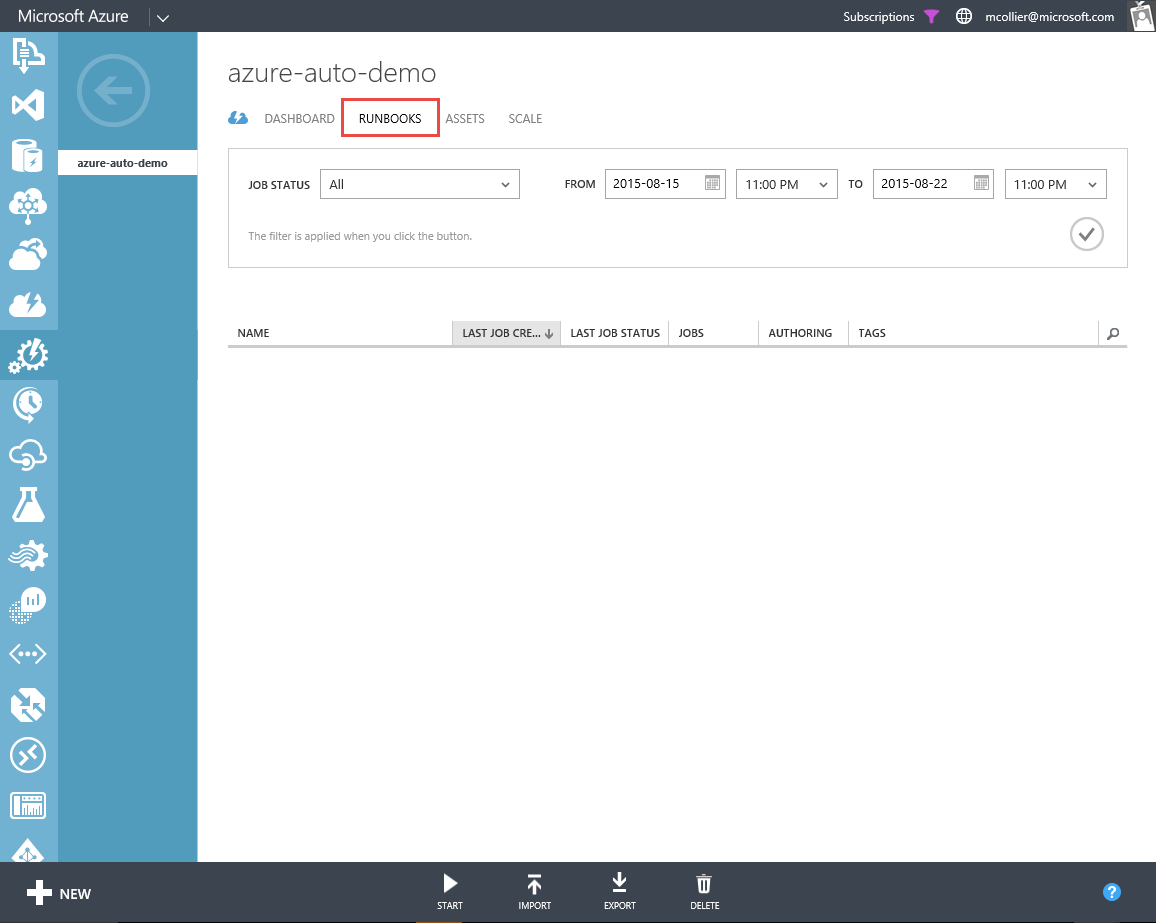
**Automation Certificate Name:** Azure Automation  
**Subscription ID:** Enter the subscription ID of your Microsoft Azure subscription recorded earlier.  
   
Click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to create this connection asset in your Azure Automation account.

## Import and Publish a Connect-Azure Runbook

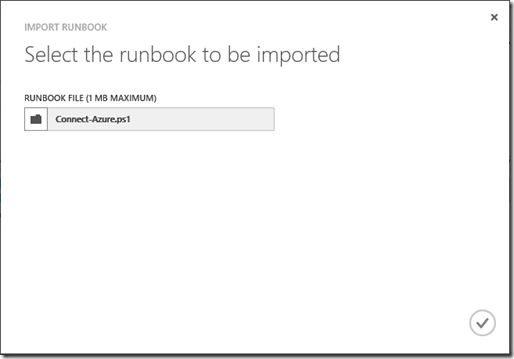
There's a few lines of code that are used to connect a runbook to your Microsoft Azure subscription using the management certificate asset and connection asset that were previously defined. To promote easy maintenance of runbooks, we recommend centralizing this code into one *Connect-Azure* runbook that other  runbooks can reference.

Luckily, the Azure Automation team has made this approach super-easy by providing us with a standard runbook template on the [Azure Automation Script Center](http://gallery.technet.microsoft.com/scriptcenter/site/search?f%5B0%5D.Type=RootCategory&f%5B0%5D.Value=WindowsAzure&f%5B0%5D.Text=Windows%20Azure&f%5B1%5D.Type=SubCategory&f%5B1%5D.Value=WindowsAzure_automation&f%5B1%5D.Text=Automation)*.*

1. Download the [Connect-Azure](http://gallery.technet.microsoft.com/scriptcenter/Connect-to-an-Azure-f27a81bb) runbook template from the Azure Automation Script Center.
2. On the details page of your Azure Automation account, click the **Runbooks** tab.

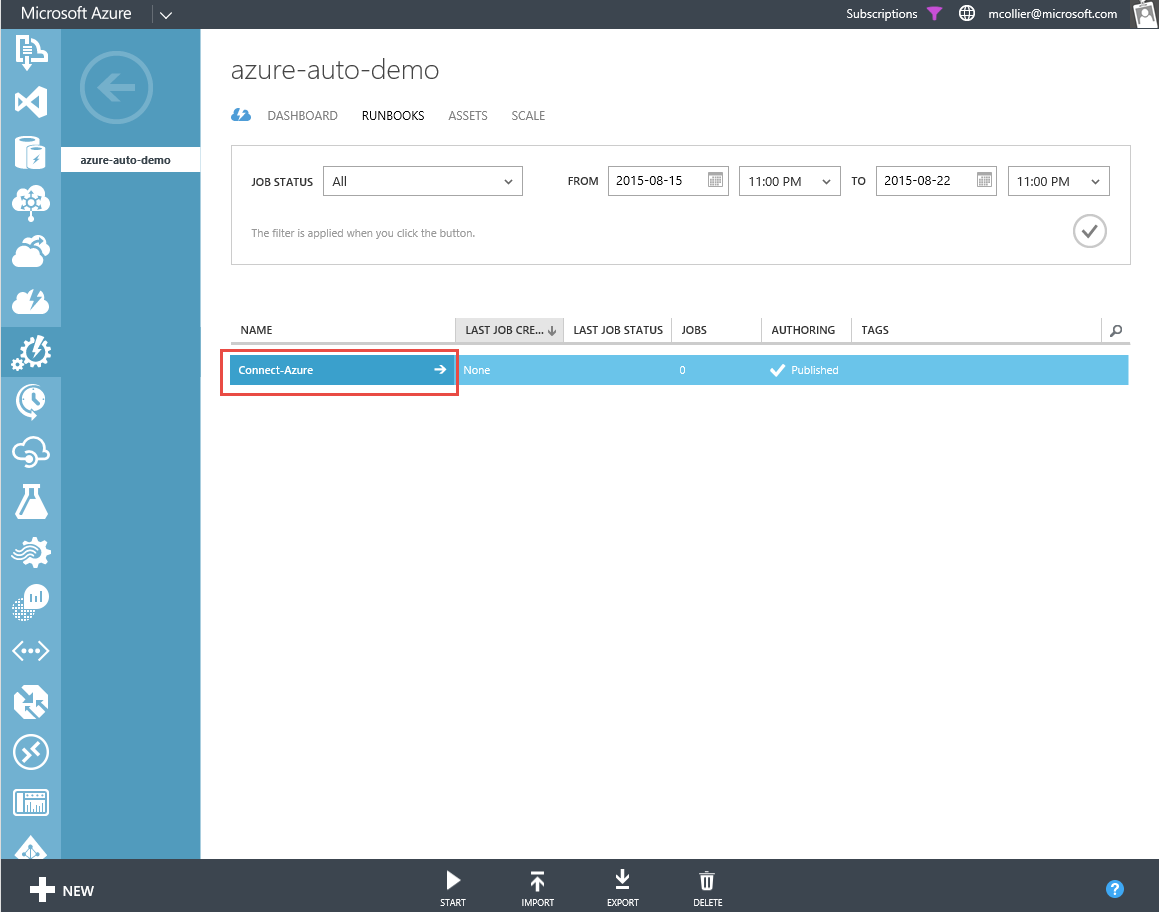


1. On the bottom toolbar, click the **Import** button. When prompted, browse to the **Connect-Azure.ps1** runbook template that you previously downloaded.

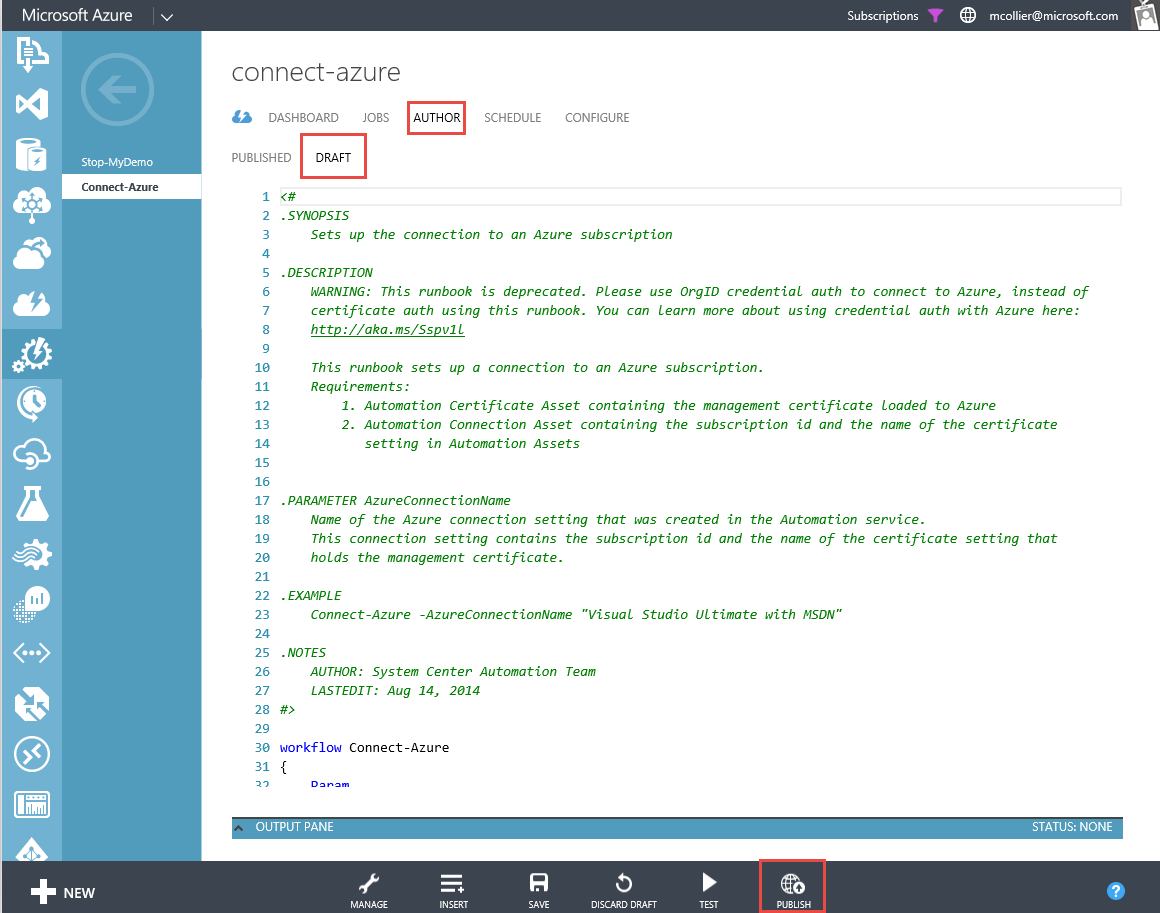
[](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/image_5F00_0632260B.png)

Click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to import this runbook template.

1. On the Runbooks tab, click on **Connect-Azure** to drill into the detailed property pages of the imported runbook.



1. On the Connect-Azure page, click on the **Author** tab followed by the **Draft** tab.



1. Click the **Publish** button on the bottom toolbar to publish the imported runbook. When prompted, click the **Yes** button to confirm publishing this runbook.

# Create Your Runbook

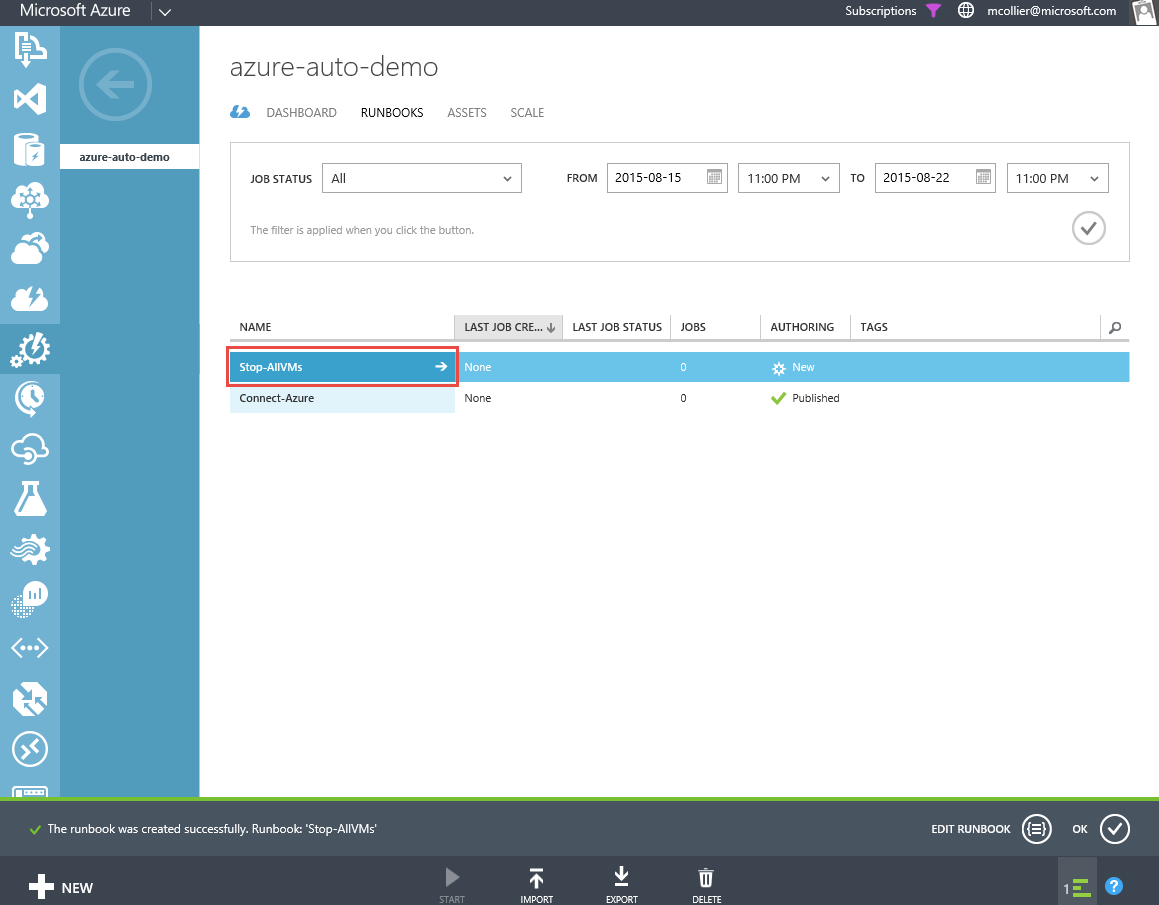
We are ready to create a runbook for automating the provisioning and management of cloud resources in your Microsoft Azure subscription. In the following section, the runbook we will create will be used to automate the safe shutdown of all Azure Virtual Machines in a specified Azure subscription at the end of each day. This runbook will stop and deallocate each virtual machine so that compute charges for these VMs do not continue to accrue when they are not being used.

1. On the bottom toolbar, click **New | App Services | Automation | Runbook | Quick Create** to create a new runbook.

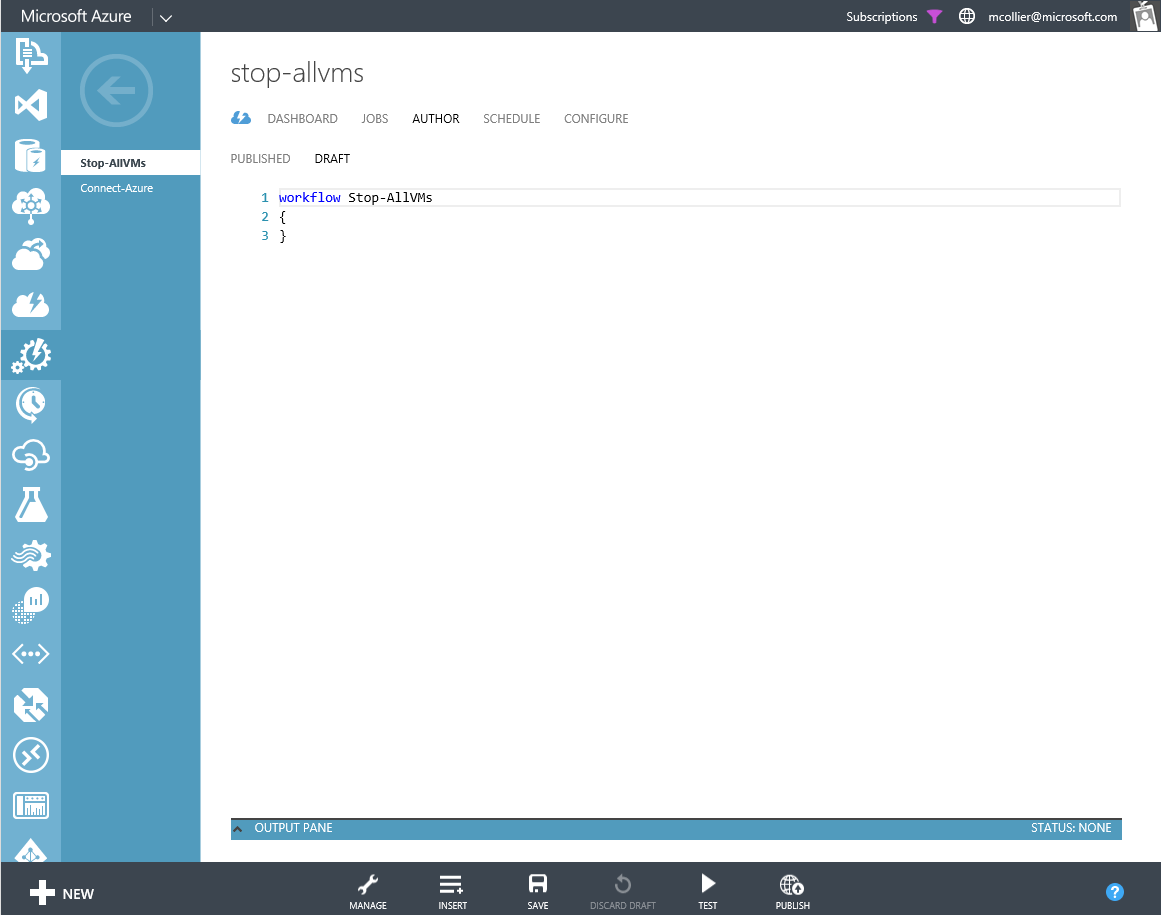


Use the following field values when creating this runbook:  
   
**Runbook Name:** Stop-AllVMs  
   
Note that runbook automation scripts are defined using PowerShell workflows.  As such, it is a best practice to name runbooks using a PowerShell [verb-noun](http://msdn.microsoft.com/en-us/library/ms714428(v=vs.85).aspx) cmdlet naming convention.  
   
**Automation Account:** Select your previously created Azure Automation Account.  
   
Click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to create this new runbook.

1. On your Azure Automation account page, click on **Stop-AllVMs** to drill into the detailed property pages for the new runbook.



1. Click the **Author** tab, and then click the **Draft** tab to begin editing the PowerShell code for this new runbook.



1. Inside the **Workflow** code block, insert the PowerShell code that leverages the Connect-Azure runbook to connect to your Microsoft Azure subscription.  Be sure to replace the values for the **$subName** variable with the value that you recorded earlier in this article.

**workflow Stop-AllVMs  
{  
  # Specify Azure Subscription Name**

**$subName = 'Enter Your Azure Subscription Name'**

**# Connect to Azure Subscription**

**Connect-Azure -AzureConnectionName $subName**

**Select-AzureSubscription -SubscriptionName $subName   
}**

You will use the above PowerShell code at the beginning of each Azure Automation runbook that provisions or manages cloud resources in a Microsoft Azure subscription.

1. Inside the **Workflow** code block, add the PowerShell code that you will use to automate cloud resources in your Microsoft Azure subscription.  In this runbook, we will insert the code (new code highlighted in yellow) used to stop and deallocate all Microsoft Azure Virtual Machines in a subscription.

**workflow Stop-AllVMs  
{  
    # Specify Azure Subscription Name  
    $subName = 'Enter Your Azure Subscription Name'  
      
    # Connect to Azure Subscription  
    Connect-Azure -AzureConnectionName $subName  
          
    Select-AzureSubscription -SubscriptionName $subName**

**Write-output "Starting shutdown of Azure VMs now!"**

**Get-AzureService | select ServiceName | foreach {**

**Get-AzureVM -ServiceName $\_.ServiceName | foreach {**

**if ($\_.InstanceStatus -eq 'ReadyRole') {**

**$currentTime = Get-Date**

**$vmName = $\_.Name**

**$svcName = $\_.ServiceName**

**Write-output "[$currentTime] - Shutting down VM [$vmName] in service [$svcName]."**

**Stop-AzureVM -ServiceName $\_.ServiceName -Name $\_.Name -Force**

**}**

**}**

**}**

**}**

1. When finished, click the **Save** button on the bottom toolbar to save your draft runbook.

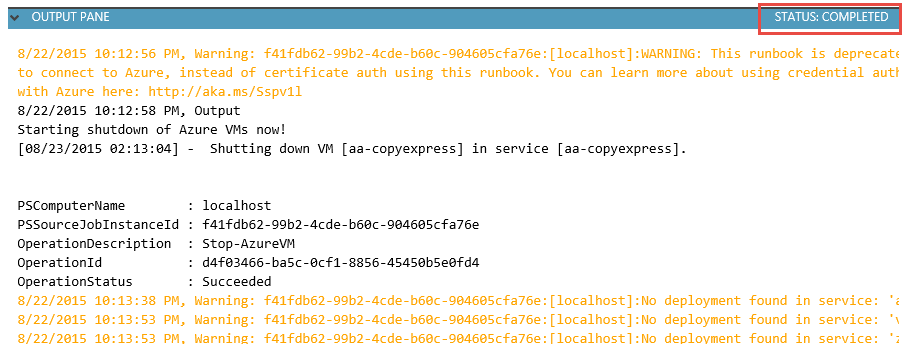
# Test the Runbook

After saving the draft runbook, you can test the new runbook to confirm that it executes successfully.

1. Click the **Test** button on the bottom toolbar.

**Caution!** When “testing” a runbook, the runbook is actually executed against your Microsoft Azure subscription. Be certain that you really do want to execute this runbook against provisioned cloud resources before clicking the **Test** button.

1. As the runbook executes, the Output Pane located at the bottom of the page will be refreshed with execution status and output.



When the runbook has completed, the **Status** value in the Output Pane will display **Completed.**

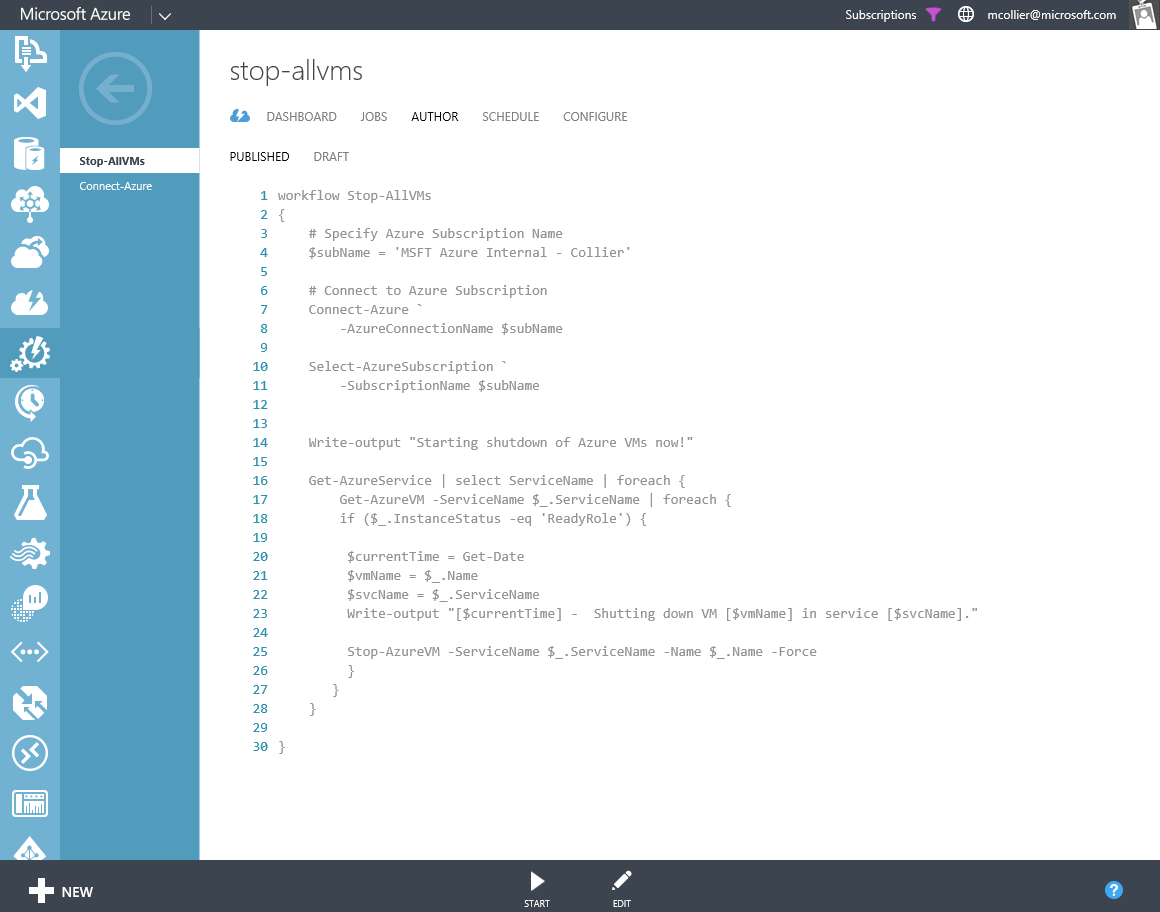
**Note**: When using a management certificate to connect to the Microsoft Azure subscription you will receive the warning message below. This warning message is hard-coded into the [Connect-Azure](https://gallery.technet.microsoft.com/scriptcenter/Connect-to-an-Azure-f27a81bb) script used previously. This is a reminder that the preferred way to connect to Microsoft Azure is no longer management certifcates but instead Azure Active Directory authentication. The script will continue to execute with a management certificate.

**Note**: When using a management certificate to connect to the Microsoft Azure subscription you will receive the warning message below. This warning message is hard-coded into the ﷟HYPERLINK "https://gallery.technet.microsoft.com/scriptcenter/Connect-to-an-Azure-f27a81bb"Connect-Azure script used previously. This is a reminder that the preferred way to connect to Microsoft Azure is no longer management certificates but instead Azure Active Directory authentication. The script will continue to execute with a management certificate. *with Azure here: http://aka.ms/Sspv1l*

# Publish the Runbook

When you’ve tested the runbook and confirmed that it executes successfully, you can publish the new runbook for running on a scheduled basis.

1. Click the **Publish** button on the bottom toolbar.
2. After the new runbook is published, click the **Published** tab to confirm that it has been published successfully.

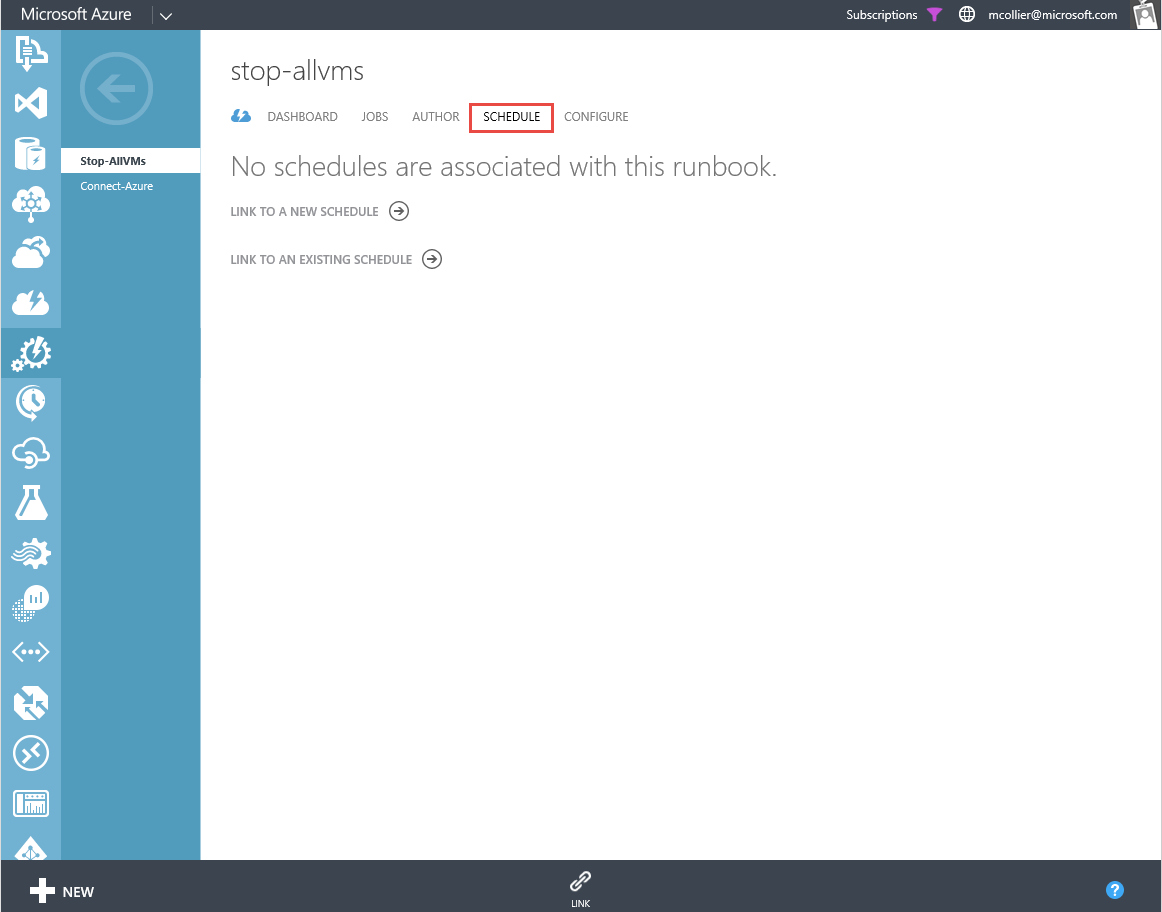


With the runbook published, you are now ready to link the runbook to a schedule. Linking to a schedule will allow the runbook to execute automatically at a configured time.

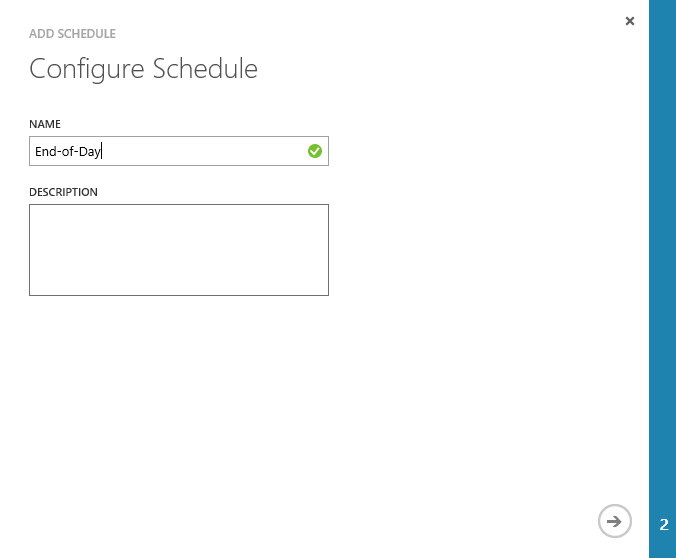
# Link the Runbook to a Schedule

We want this new runbook to execute at the end of every day, to make sure that our lab VM’s are automatically stopped and deallocated when not being used.  To execute a runbook on a scheduled basis, we can link the runbook to a recurring schedule.

1. Click on the **Schedule** tab.

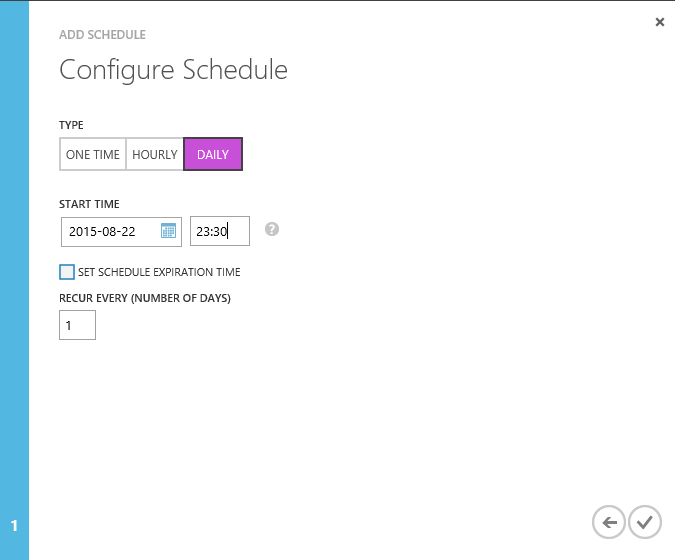


1. Click **Link to a New Schedule** and enter **End-of-Day** as the name of the new schedule.

Click the [image](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/image_5F00_0F58C3AB.png) button to continue.

1. On the Configure Schedule page, define a scheduled **Start Time** and **Recur Every** value. Be sure to set the **Type** to DAILY to execute this runbook each day.



**Note**: The time used is local time (not UTC).

Click the [clip_image001](http://blogs.technet.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-00-94-09-metablogapi/3302.image_5F00_41419357.png) button to add this new schedule and link it to the runbook.

# Contacts

The purpose of this section is to identify key contacts involved in the creation and review of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Contact Info | Role |
| Michael S. Collier | Cloud Solution Architect | [michael.collier@microsoft.com](mailto:michael.collier@microsoft.com) | Author |
| Neil Mackenzie | Cloud Solution Architect | [neimac@microsoft.com](mailto:neimac@microsoft.com) | Author |
| Alex Belotserkovskiy | Technical Evangelist | [albe@microsoft.com](mailto:albe@microsoft.com) | Reviewer |
| Simon Gurevich | Principal Consultant | [Simon.Gurevich@microsoft.com](mailto:Simon.Gurevich@microsoft.com) | WW Communities – Azure, Lead |

# Appendix

## Appendix A – Configure via PowerShell

Below is a PowerShell script which will perform many of the above tasks. There are a few important details to be aware of prior to executing the script:

1. It does not upload the management certificate. That must be done in the Azure Management Portal.
2. It requires the two runbooks, Stop-AllVMs.ps1 and Connect-Azure.ps1, to be in the current executing directory. The two scripts are embedded below.
3. The Stop-AllVMs.ps1 script contains the hardcoded Azure subscription name. You will need to update this to be the name of the Azure subscription to look at for stopping the VMs.



The PowerShell script below must be executed from an administrative PowerShell console (or ISE).

CLS

Add-AzureAccount

# Name of Azure Automation account

$automationAccountName = '[YOUR-NEW-AZURE-AUTOMATION-ACCOUNT-NAME]'

# Location for Automation Account (seems to be a PS bug - since it actually goes in East US 2)

$location = 'East US'

# Name of Azure Automation and Management certificate

$certificateName = 'AzureAutomationPS'

$stopRunbook = 'Stop-AllVMs'

$connectRunbook = 'Connect-Azure'

$scheduleName = 'EveryNight'

$startTime = '9:00:00 PM'

# Set a base path to use for uploading the runbooks

$scriptPath = $MyInvocation.MyCommand.Path

$scriptPath = Split-Path $scriptPath

# Select the appropriate subscription

$subscriptionName = (Get-AzureSubscription).SubscriptionName | Out-GridView -Title "Select Azure Subscription" -PassThru

Select-AzureSubscription -SubscriptionName $subscriptionName

$subscriptionId = (Get-AzureSubscription -Current).SubscriptionId

# Create the Automation/Management certificate and export it in CER and PFX formats

$thumbprint = (New-SelfSignedCertificate -DnsName "$certificateName" -CertStoreLocation Cert:\CurrentUser\My -KeySpec KeyExchange).Thumbprint

$cert = (Get-ChildItem -Path cert:\CurrentUser\My\$thumbprint)

Export-Certificate -Cert $cert -FilePath "$scriptPath\$certificateName.cer" -Type CERT

$password = Read-Host -Prompt "Please enter the certificate password." -AsSecureString

Export-PfxCertificate -Cert $cert -FilePath "$scriptPath\$certificateName.pfx" -Password $password

# Create the Automation account

New-AzureAutomationAccount -Name $automationAccountName -Location $location -Verbose

# Upload the Automation certificate

New-AzureAutomationCertificate -AutomationAccountName $automationAccountName -Name $certificateName -Password $password -Path "$certificateName.pfx"

# Create the Automation connection

$connectionFieldValues = @{"AutomationCertificateName"=$certificateName; "SubscriptionID"=$subscriptionId};

New-AzureAutomationConnection -AutomationAccountName $automationAccountName -Name $subscriptionName -ConnectionTypeName 'Azure' -ConnectionFieldValues $connectionFieldValues

# Create, upload and publish the Stop-AllVMs runbook

New-AzureAutomationRunbook -AutomationAccountName $automationAccountName -Path "$scriptPath\$stopRunbook.ps1"

Publish-AzureAutomationRunbook -AutomationAccountName $automationAccountName -Name $stopRunbook

# Create, upload and publish the Connect-Azure runbook

New-AzureAutomationRunbook -AutomationAccountName $automationAccountName -Path "$scriptPath\$connectRunbook.ps1"

Publish-AzureAutomationRunbook -AutomationAccountName $automationAccountName -Name $connectRunbook

# Create the schedule

New-AzureAutomationSchedule -AutomationAccountName $automationAccountName -Name $scheduleName -StartTime $startTime -DayInterval 1

# Connect the schedule to the runbook

Register-AzureAutomationScheduledRunbook -AutomationAccountName $automationAccountName -RunbookName $stopRunbook -ScheduleName $scheduleName