

Microsoft Azure: Automatic Scaling of Remote Desktop Session Hosts

Reducing Costs of Desktop Hosting on Microsoft Azure Infrastructure Services

Published: November 2016  
Microsoft Corporation

**Copyright information**

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

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.

Microsoft, Active Directory, Hyper-V, SQL Server, Windows PowerShell, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

© 2016 Microsoft Corporation. All rights reserved.

# Introduction

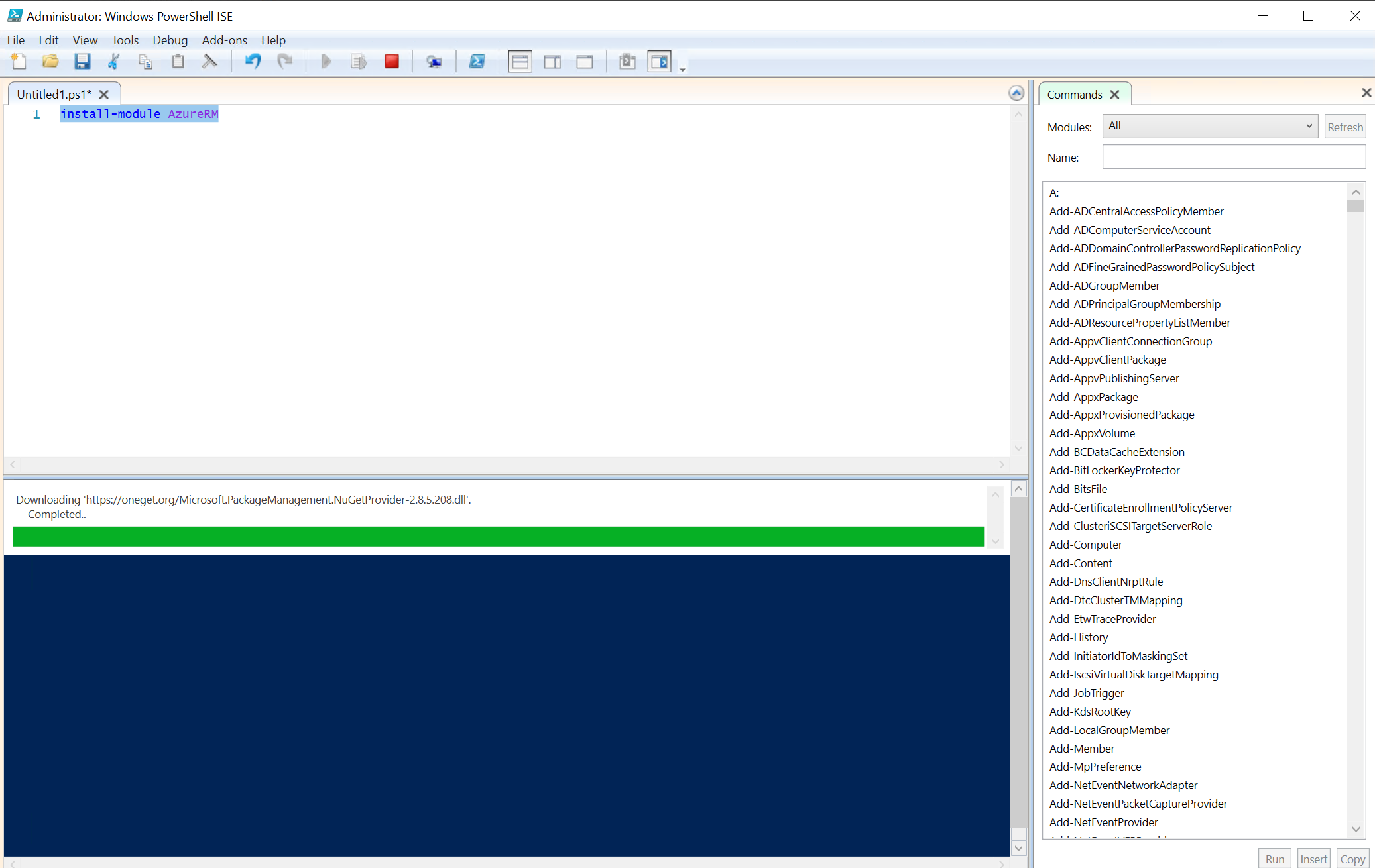
RDSScaler.ps1 is a sample PowerShell script that can be used as a starting point for developing a solution to automatically scale a Remote Desktop Services (RDS) deployment on Microsoft Azure Infrastructure Services.

For many RDS deployments in Azure, the virtual machine costs of the Remote Desktop Session Host (RDSH) servers represent the most significant portion of the total RDS deployment cost. To reduce cost, the script automatically shuts down and de-allocates RDSH server VMs during off-peak usage hours and then restarts them during peak usage hours.

# Prerequisites

The environment to be used to execute the script must meet the following requirements.

1. An RDS deployment in Azure under Azure Resource Manager mode with 2 or more RDSH servers in a collection, and each Azure virtual machine name has the same name as the hostname of the operating system running on that virtual machine. More Information on deploying RDS in Azure can be found [here](http://msdn.microsoft.com/en-us/library/azure/dn451351.aspx).
2. Internet access from the Remote Desktop Connection Broker (RD Connection Broker) server.
3. PowerShell 4.0 or higher (default for Windows Server 2012 R2) installed on the RD Connection Broker server. (The script must be run on the VM with the RD Connection Broker role service installed.)
4. Microsoft Azure Resource Manager PowerShell Module installed on the RD Connection Broker server. (Hint: Temporarily turn off IE Enhanced Security on the local server using Server Manager to make this task easier.) You can download and install the Azure Resource Manager PowerShell modules by running Install-Module AzureRM in PowerShell as shown below.



1. An Azure service principal must be created and configured with the appropriate permissions to run the script. Please refer to the following documentation for details regarding how to create and configure the service principal.

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authenticate-service-principal>

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal>

# Script Deployment

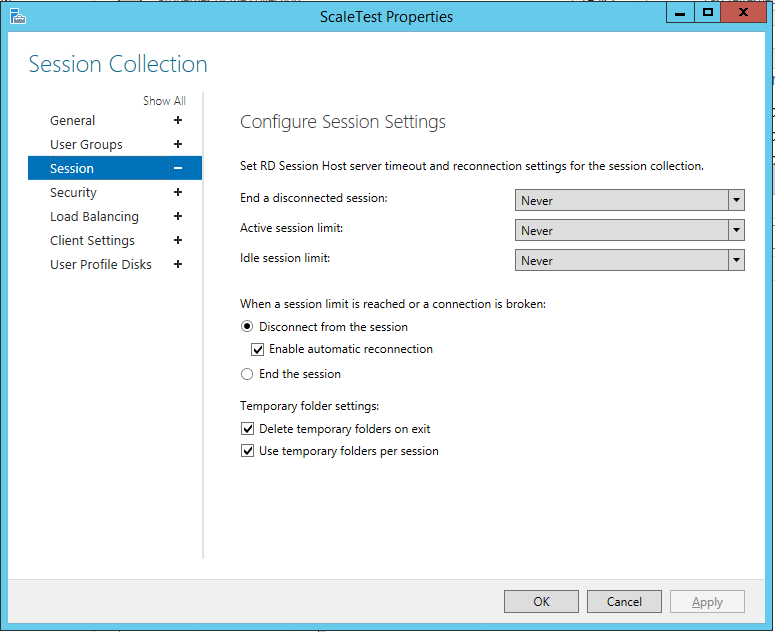
Use the following procedure to deploy the script.

1. Logon to the RD Connection Broker server using an administrative account.
2. Create a folder on the RD Connection Broker server (e.g. c:\DynamicRDSH)
3. Download the RDSScaler.ps1 and Config.xml files from Script Center and copy them to the folder.
4. You must create and configure the service principal with appropriate permissions within your Azure subscription. If you are using the certificate for authenticating the service principal, you need to import the certificate.
5. If you are using certificate based authentication, you need to uncomment the section within the RDSScaler.ps1 file, and comment the section for password based authentication.
6. Update the Config.xml file with the Azure subscription, resource group, and the RD Connection Broker
   1. Open the Config.xml file using Notepad
   2. Replace the AADTenantId value with your Azure Tenant Id.
   3. Replace the AADApplicationId Value with the application id you created and configured for the service principal.
   4. Replace the AADAppCertThumbprint value with the certificate you configured for authenticating the service principal, if you are using certificate based authentication.
   5. Replace the AADServicePrincipalSecret value with the secret/password you configured for the service principal, if you are using password based authentication.
   6. Replace the CurrentAzureSubscriptionName value with your Azure subscription name.
   7. Replace the ResourceGroupName value with the resource group name your RDS environment deployed.
   8. Update additional values in the Config.xml file as desired (See the section below.)
   9. Save the Config.xml file.
7. Configure the Task Scheduler to run the RDSScaler.ps1 file at a regular interval
   1. In **Server Manager**, select **Tools**, and **Task Scheduler**.
   2. In the **Task Scheduler** window, select **Create Task …**
   3. In **the Create Task** dialog, select the **General** tab, enter a **Name: (**e.g. Dynamic RDSH), select **Run whether user is logged on or not** and **Run with highest privileges**
   4. Select the **Triggers** tab and **New…**
   5. In the **New Trigger** dialog, select **Begin the task: At task creation/modification**, check **Repeat task every:** and select the appropriate period and duration (e.g. **15 minutes** and **Indefinitely**)
   6. Select the **Actions** tab and **New…**
   7. In the **New Action** dialog, in the **Program/script:** field, type   
      powershell.exe C:\DynamicRDSH\RDSScaler.ps1
   8. Select **OK** to accept defaults under the **Conditions** and **Settings** tabs
   9. Enter the password for the administrative account used to run the script

# Script Configuration

The behavior of the script can be modified by editing the config.xml file. The config.xml file consists of two sections, the Azure section and the RDSScaleSettings section. Each section contains variables that are read by the script each time it runs.

1. Azure section:
   1. AADTenantId: Specifies the Azure Active Directory Tenant Id your azure subscription associated with.
   2. AADApplicationId: The GUID for the Azure Active Directory Application you create for service principal.
   3. AADAppCertThumbprint: The thumbprint of the certificate you configured for authenticating your service principal.
   4. AADServicePrinicpalSecret: The secret/password you created for your Azure service principal.
   5. CurrentAzureSubscriptionName: The name of your Azure subscription
   6. ResourceGroupName: The Azure resource group name your RDS environment deployed.
2. RDSScaleSettings section:
   1. BeginPeakTime: Starting time of the business work day (peak), in 24-hour notation
   2. EndPeakTime: Ending time of the business day, in 24-hour notation
   3. TimeDifferenceInHours: The difference in hours between your local time zone and Coordinated Universal Time (UTC). Microsoft Azure virtual machines use UTC time by default. You can use this value to allow BeginPeakTime and EndPeakTime to be entered in the local time zone notation.
   4. SessionThresholdPerCPU: Maximum number of sessions per Azure CPU allowed during peak time before a new RDSH server is started.
   5. MinimumNumberofRDSH: The minimum number of RDSH servers that remain active during off-peak time.
   6. LimitSecondsToForceLogOffUser: The number of seconds the script will wait until forcing user logoff during off-peak hours. If this parameter is set to 0, the script will not force user logoffs. It will use the session configuration in the **Session Collection** properties configured separately from Server Manager or PowerShell. The Server Manager UI is shown below.



* 1. LogOffMessageTitle: The title of the notification message sent to a user before forcing the user to log off.
  2. LogOffMessageBody: The body of the message sent to a user before forcing the user to log off.

# Log Files

The script creates two log files, RDSScale.log and RDSUsage.log. The RDSScale.log will log the events and errors (if any) during each execution of the script.

The RDSUsage.log file will record the active number of cores and active number of virtual machines at each execution of the script. You can use this information to estimate the actual usage of Microsoft Azure VMs and the cost. The file is formatted as comma separated values, with each line containing the following information.

time, collection, cores, VMs

The file name can be modified to have a .csv extension, loaded into Microsoft Excel, and analyzed.

# Detailed Description

The script reads settings from a config.xml file, including the start and end of the peak usage period during the day.

During the peak usage time, the script checks the current number of sessions and the current running RDSH capacity for each collection. It calculates if the running RDSH servers have sufficient capacity to support existing sessions based on the SessionThresholdPerCPU parameter defined in the config.xml file. If not, the script starts additional RDSH servers in the collection.

During the off-peak usage time, the script determines which RDSH servers should be shutdown based on the MinimumNumberOfRDSH parameter in the config.xml file. The script will set the RDSH servers to drain mode to prevent new sessions connecting to the hosts. If the LimitSecondsToForceLogOffUser parameter in the config.xml file is set to a non-zero positive value, the script will notify any logged on users to save work, wait the configured amount of time, and then force the users to logoff. Once there are no user sessions on an RDSH server, it will shut down the RDSH server.

If LimitSecondsToForceLogOffUser parameter in the config.xml file is set to zero, the script will allow the session configuration setting in the collection properties to handle the logoff of user sessions. If there are any sessions on an RDSH server, it will leave the RDSH server running. If there are no sessions, the script will shut down the RDSH server.

The script is designed to run periodically on the Remote Desktop (RD) Connection Broker server using Task Scheduler. You should select the appropriate time interval based on the size of your RDS environment since starting and shutting down virtual machines can take some time.

If you have two RD Connection Brokers in a high availability configuration, you can deploy the script on both RD Connection Brokers. The script will automatically determine which RD Connection Broker is the active management server. Only the script running on the active management server will affect the RDSH servers. The script that is not running on the active management server will simply exit.

Version History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| v1 | September 2014 | First version |
| v1.1 | March 2015 | Fixed comment syntax error in the xml file.  Fixed collection name comparison error. Script now works with collection names with spaces and changed collection names. |
| V2.0 | November 2016 | Updated the script to support Azure Resource Manager |