# The Queen's College IT Open Morning

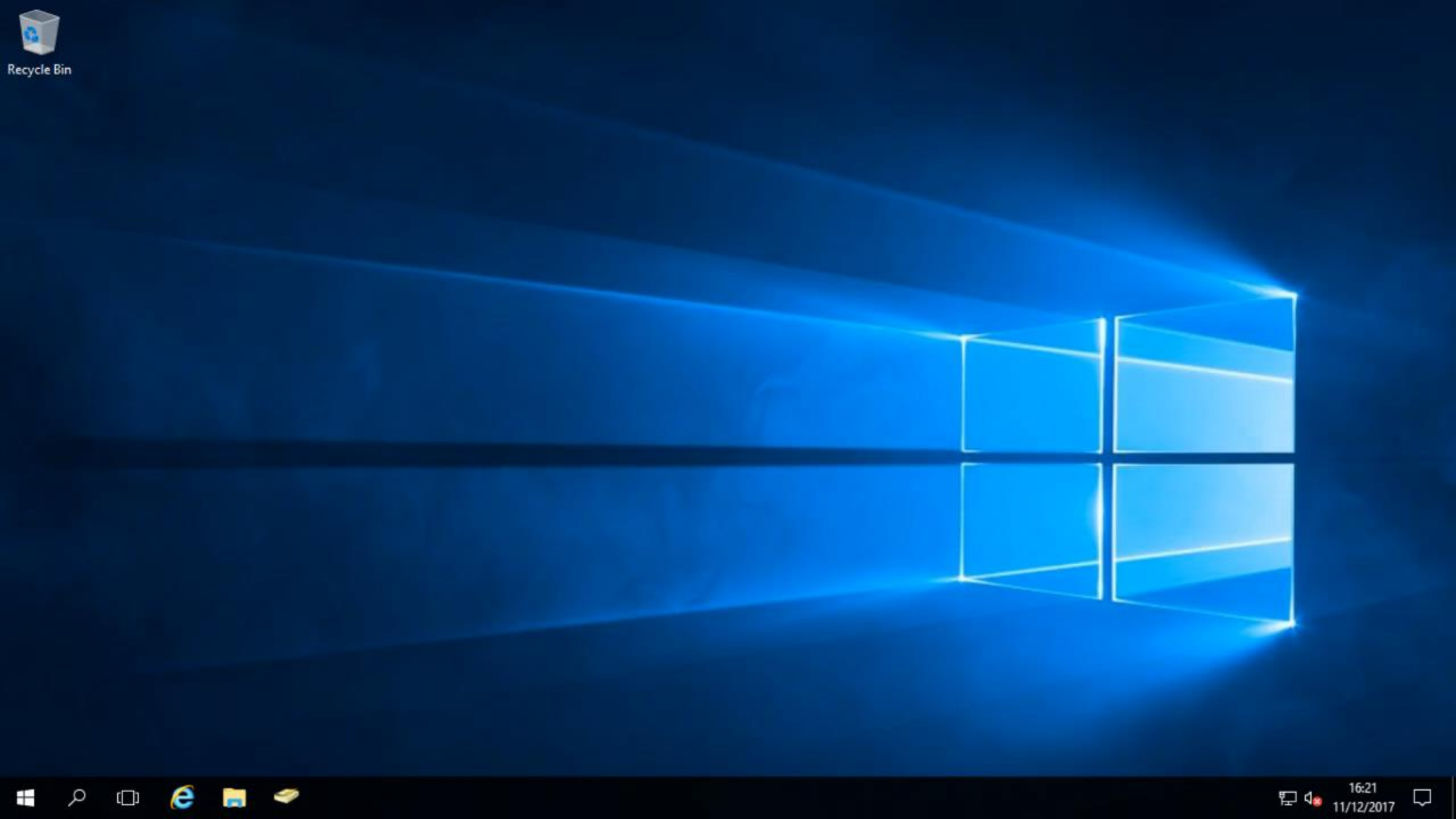
PowerShell

# What does it do?

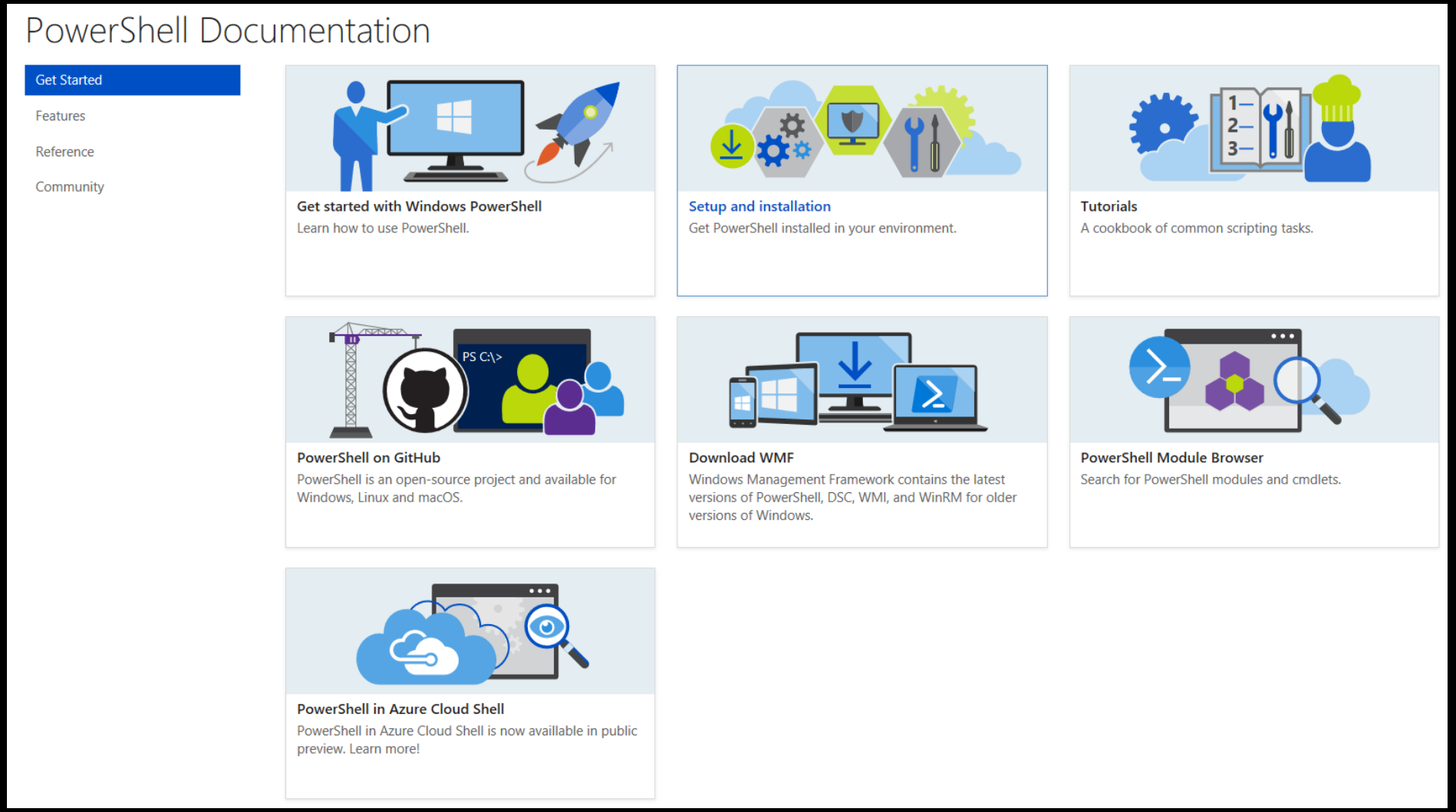
- Pretty much anything that has an API it can talk to
  - So pretty much anything
    - Sending emails
    - Talking direct T-SQL
    - Modifying files
    - Installing roles and features
    - Outputting data as CSV/HTML
    - Managing SYMPA mailing lists
    - Configuring network switches
- Interactive scripts and non interactive

# Alternatives…

- Python
- JavaScript
- VBScript
- Sooo many others…


Recycle Bin
16:21
11/12/2017

# Get the latest version! (if not on Windows 10)



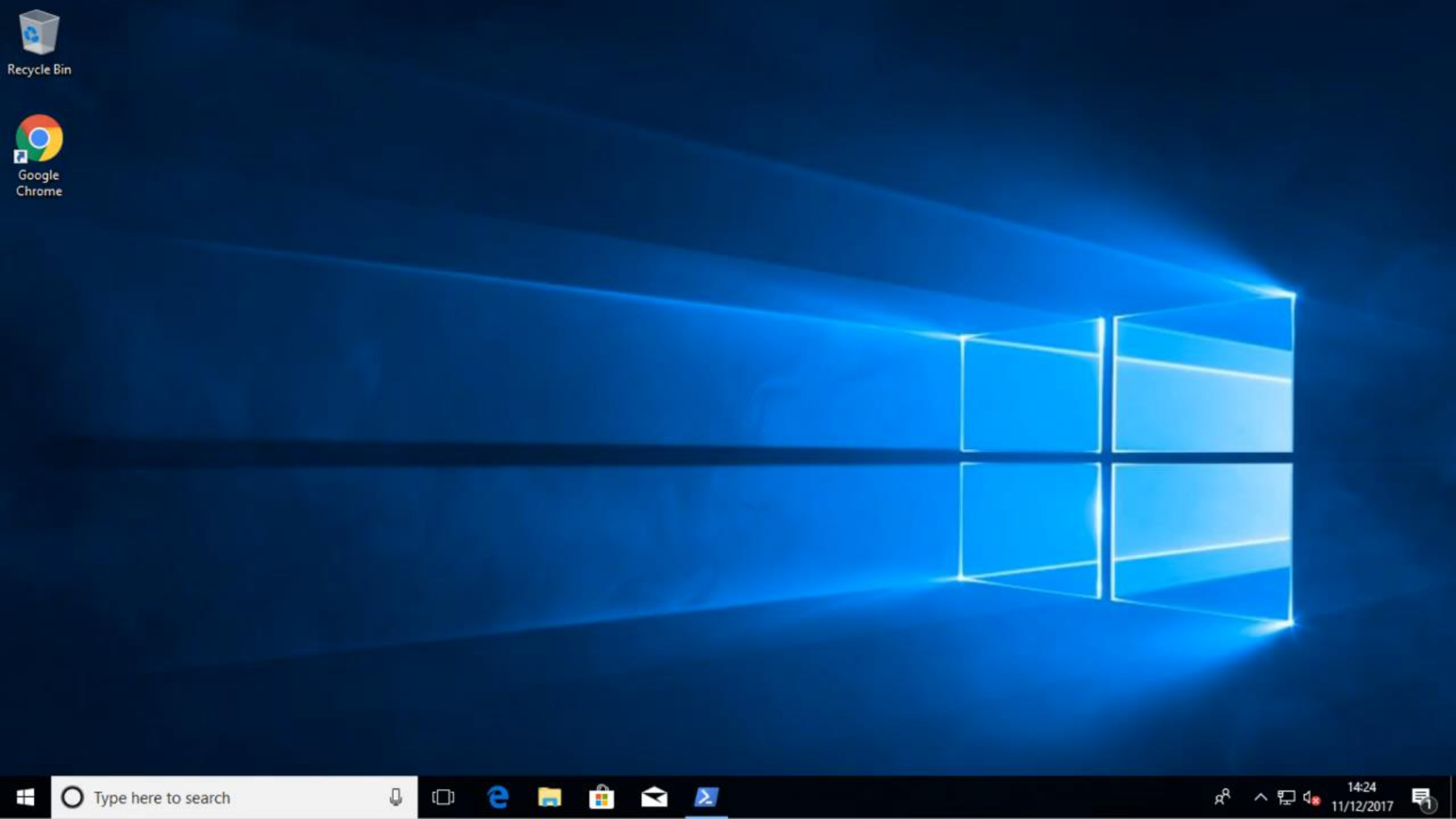

https://docs.microsoft.com/en-us/powershell/

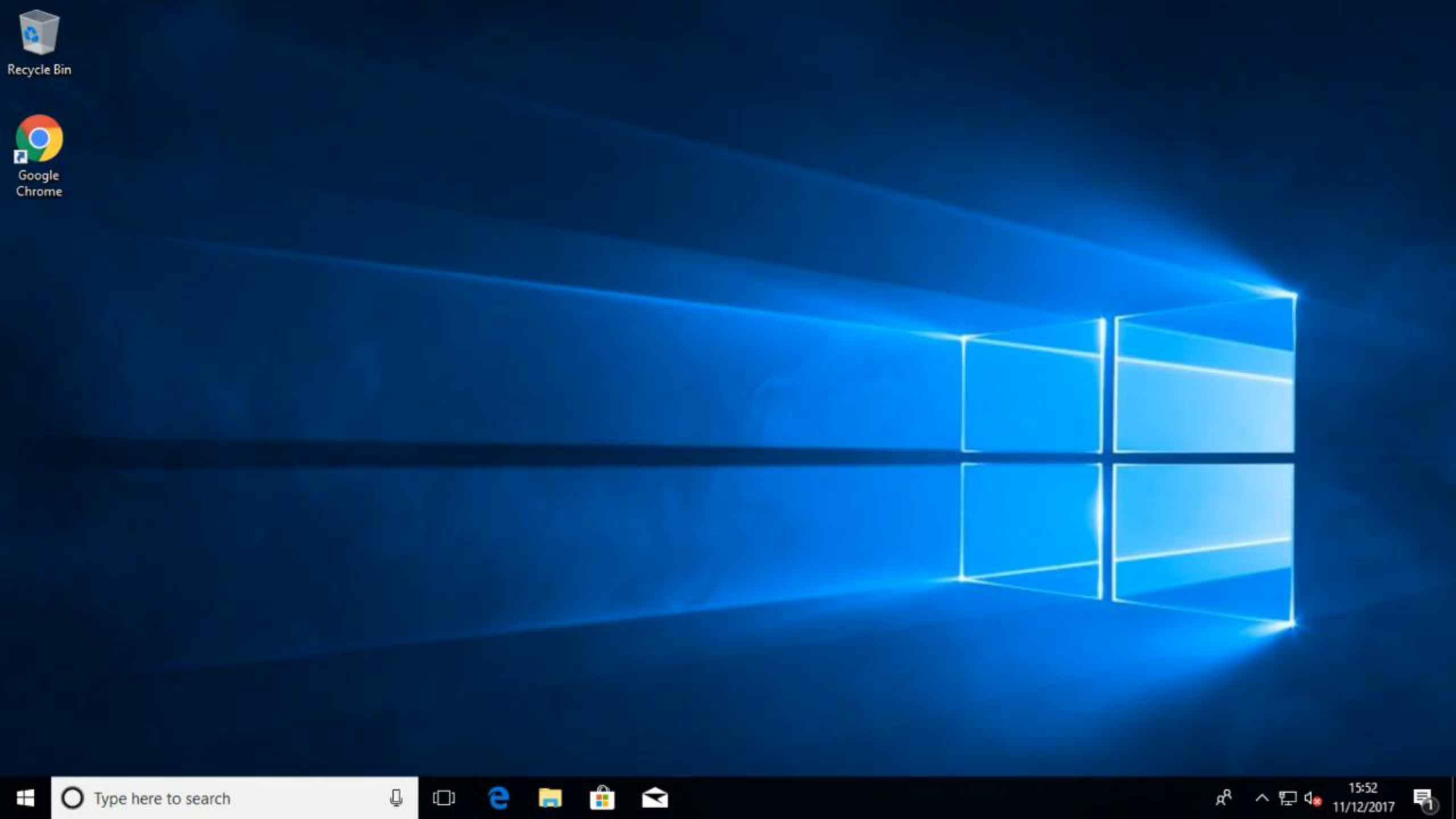
# Windows Management Framework 5.1

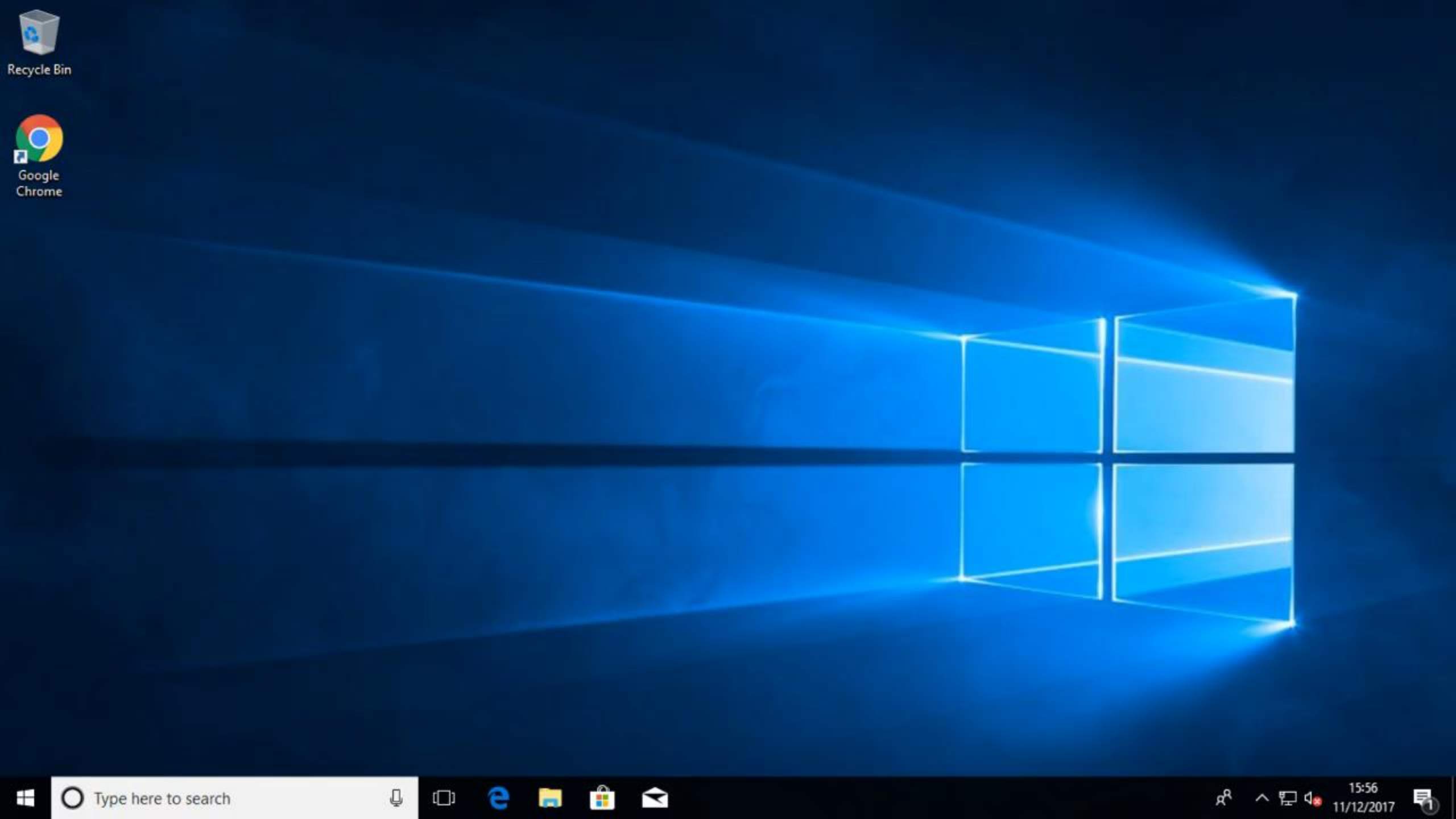
Language: **English** Download

Windows Management Framework 5.1 includes updates to Windows PowerShell, Windows PowerShell Desired State Configuration (DSC), Windows Remote Management (WinRM), Windows Management Instrumentation (WMI). Release notes: https://go.microsoft.com/fwlink/?linkid=839460

- Details
- System Requirements
- Install Instructions


Recycle Bin
Google Chrome
Type here to search
14:24
11/12/2017


Recycle Bin
Google Chrome
Type here to search
15:52
11/12/2017


Recycle Bin
Google Chrome
Type here to search
15:56
11/12/2017

# Remoting

- PowerShell Remoting is enabled by default on Server 2012 onwards for
  - Domain/Private Networks – any client
  - Public Networks – local subnet only
- By default only members of the Administrators group can use Remoting
- On older clients use Enable-PSRemoting -Force to enable remote management (or configure through Group Policy)
- Test connections with Test-WsMan <clienthostname>

# Get Remote Server Administration Tools

https://support.microsoft.com/en-gb/help/2693643/remote-server-administration-tools-rsat-for-windows-operating-systems

# Remote Server Administration Tools (RSAT) for Windows operating systems

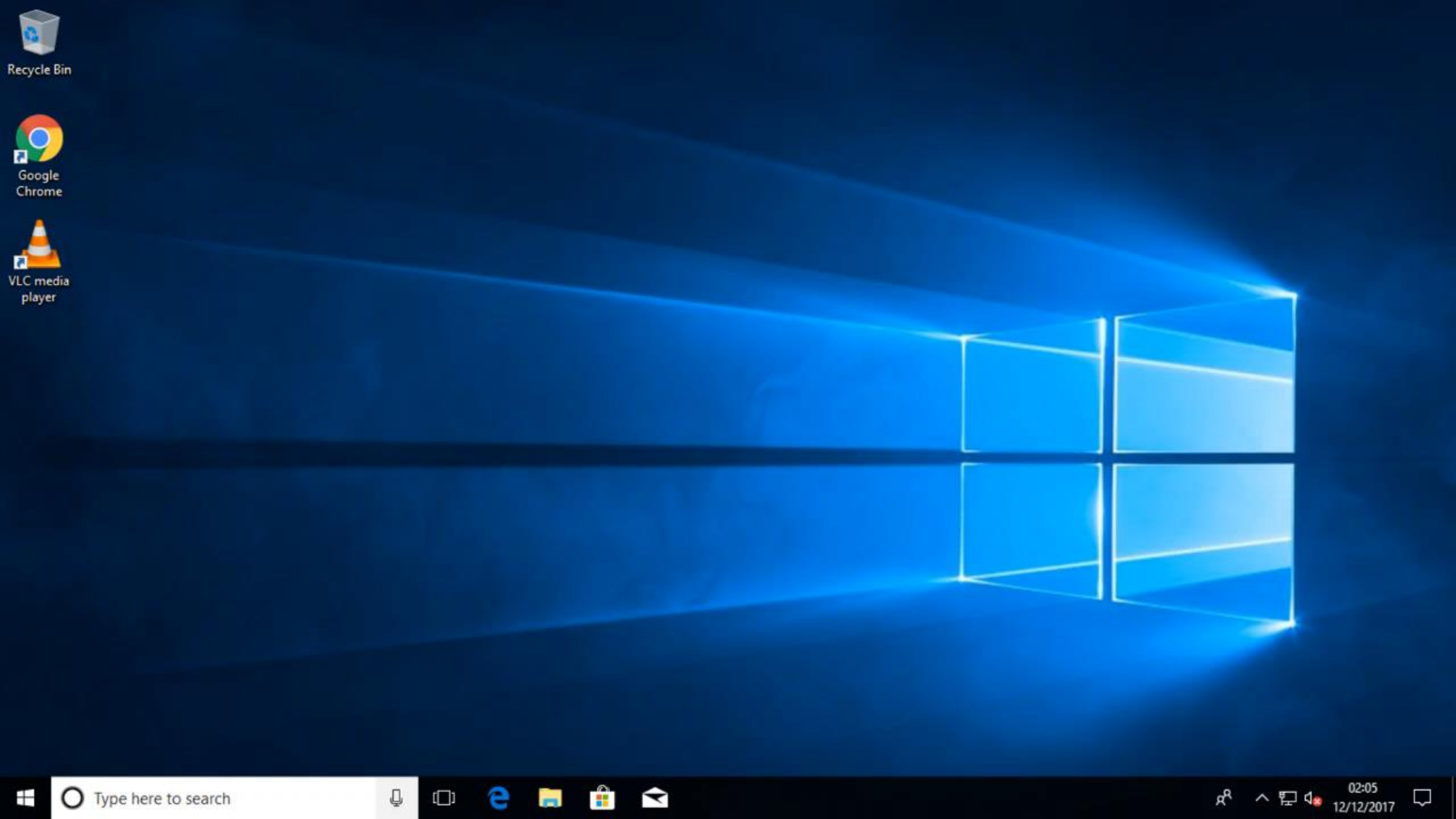


Applies to multiple products

## Introduction

Remote Server Administration Tools (RSAT) enables IT administrators to remotely manage roles and features in Windows Server 2012 R2, Windows Server 2012, Windows Server 2008, and Windows Server 2008 R2 from a computer that is running Windows 10, Windows 8.1, Windows 8, Windows 7, or Windows Vista.

You cannot install RSAT on computers that are running Home or Standard editions of Windows. You can install RSAT only on Professional or Enterprise editions of the Windows client operating system. Unless the download page specifically states that RSAT applies to a beta, preview, or other prerelease version of Windows, you must be running a full (RTM) release of the Windows operating system to install and use


Recycle Bin
Google Chrome
VLC media player
Type here to search
02:05
12/12/2017

Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

Untitled1.ps1

```
1
```

```
PS C:\Users\administrator.KINGS>
```

Commands

Modules: All Refresh

Name:

A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicationPolicy
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BgpCustomRoute
Add-BgpPeer
Add-BgpRouteAggregate
Add-BgpRouter
Add-BgpRoutingPolicy
Add-BgpRoutingPolicyForPeer
Add-BitLockerKeyProtector
Add-BitsFile
Add-CauClusterRole
Add-CertificateEnrollmentPolicyServer
Add-ClusterCheckpoint

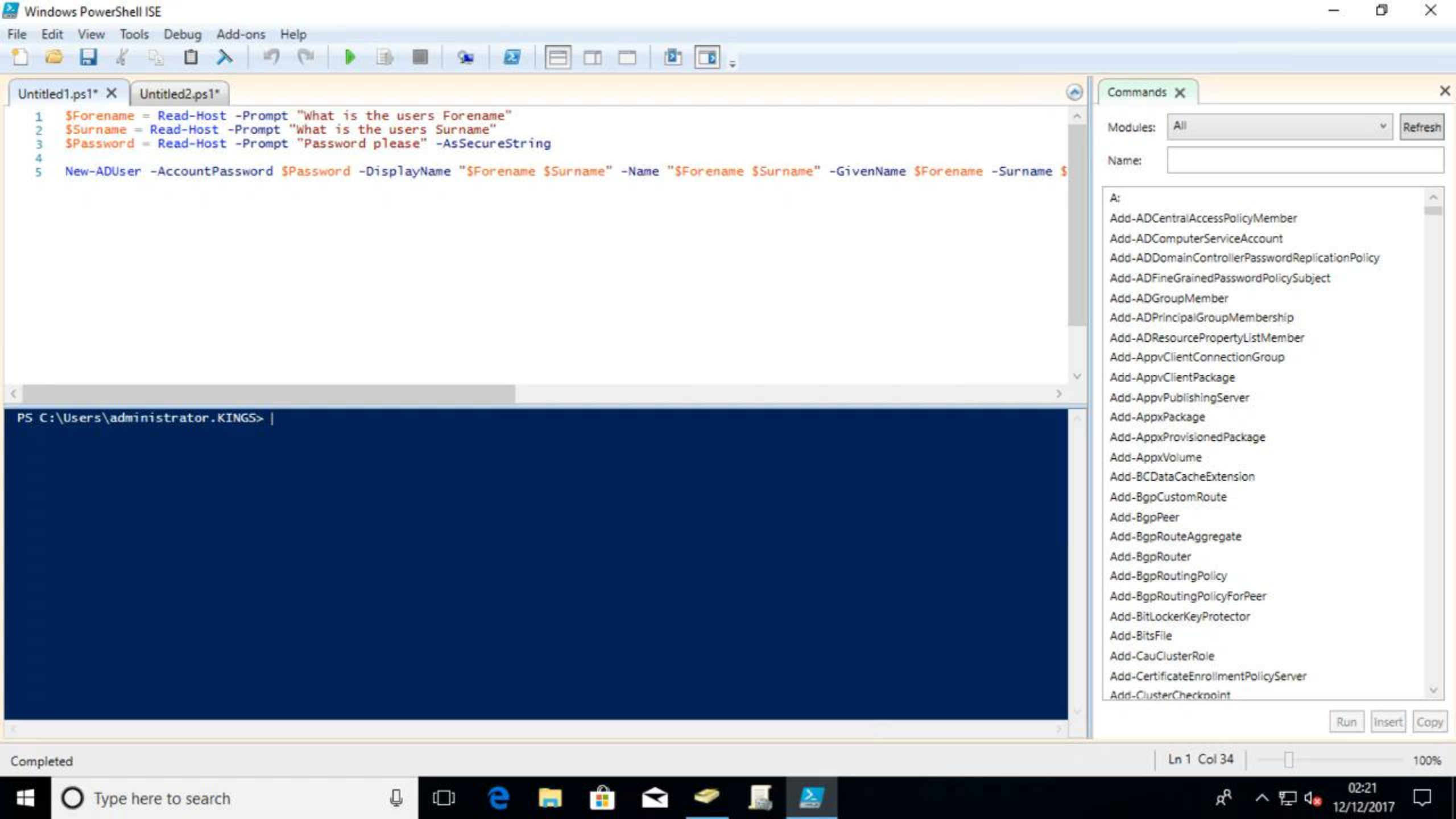
Run Insert Copy

Ln 1 Col 1 100%

Group Policy Management

Type here to search

02:07
12/12/2017


Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1*
Untitled2.ps1*
1 $Forename = Read-Host -Prompt "What is the users Forename"
2 $Surname = Read-Host -Prompt "What is the users Surname"
3 $Password = Read-Host -Prompt "Password please" -AsSecureString
4
5 New-ADUser -AccountPassword $Password -DisplayName "$Forename $Surname" -Name "$Forename $Surname" -GivenName $Forename -Surname $
PS C:\Users\administrator.KINGS>
Commands
Modules: All
Refresh
Name:
A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicationPolicy
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BgpCustomRoute
Add-BgpPeer
Add-BgpRouteAggregate
Add-BgpRouter
Add-BgpRoutingPolicy
Add-BgpRoutingPolicyForPeer
Add-BitLockerKeyProtector
Add-BitsFile
Add-CauClusterRole
Add-CertificateEnrollmentPolicyServer
Add-ClusterCheckpoint
Run
Insert
Copy
Completed
Ln 1 Col 34
100%
Type here to search
02:21
12/12/2017

Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

Untitled1.ps1*

```
1  function KingsNew-AdAccount {
2  param(
3      [Parameter(Mandatory=$true)]
4      [String]$Forename,
5      [Parameter(Mandatory=$true)]
6      [String]$Surname
7  )
8
9  $Password = Read-Host -Prompt "Password please" -AsSecureString
10
11 New-ADUser -AccountPassword $Password -DisplayName "$Forename $Surname" -Name "$Forename $Surname" -GivenName $Forename -Surname $
12 }
```

```
PS C:\Users\administrator.KINGS> KingsNew-AdAccount

PS C:\Users\administrator.KINGS>
```

Commands

Modules: All Refresh

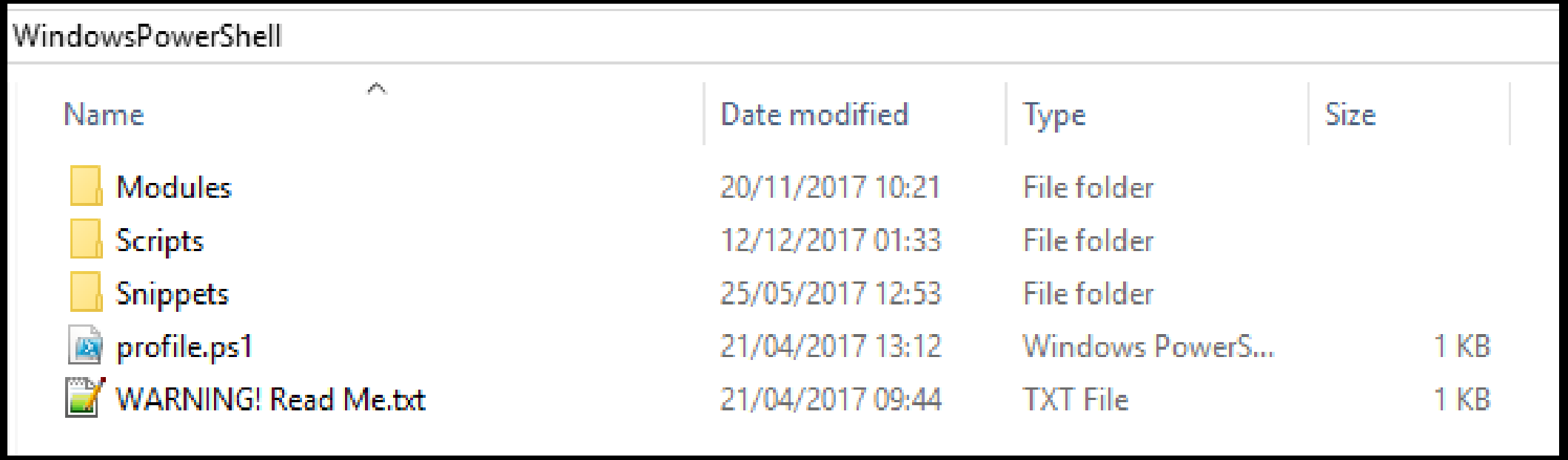
Name:

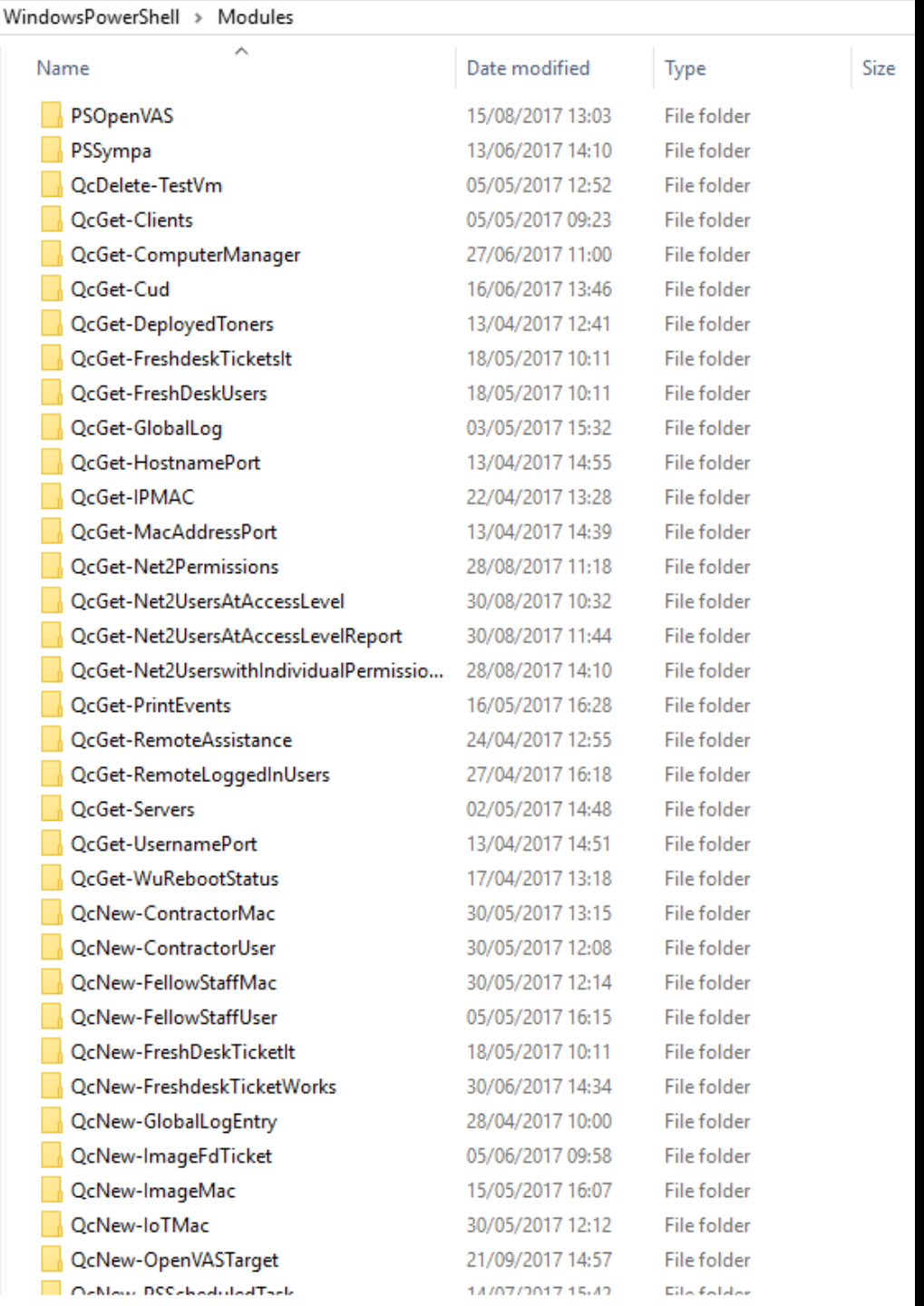
A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicationPolicy
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BgpCustomRoute
Add-BgpPeer
Add-BgpRouteAggregate
Add-BgpRouter
Add-BgpRoutingPolicy
Add-BgpRoutingPolicyForPeer
Add-BitLockerKeyProtector
Add-BitsFile
Add-CauClusterRole
Add-CertificateEnrollmentPolicyServer
Add-ClusterCheckpoint

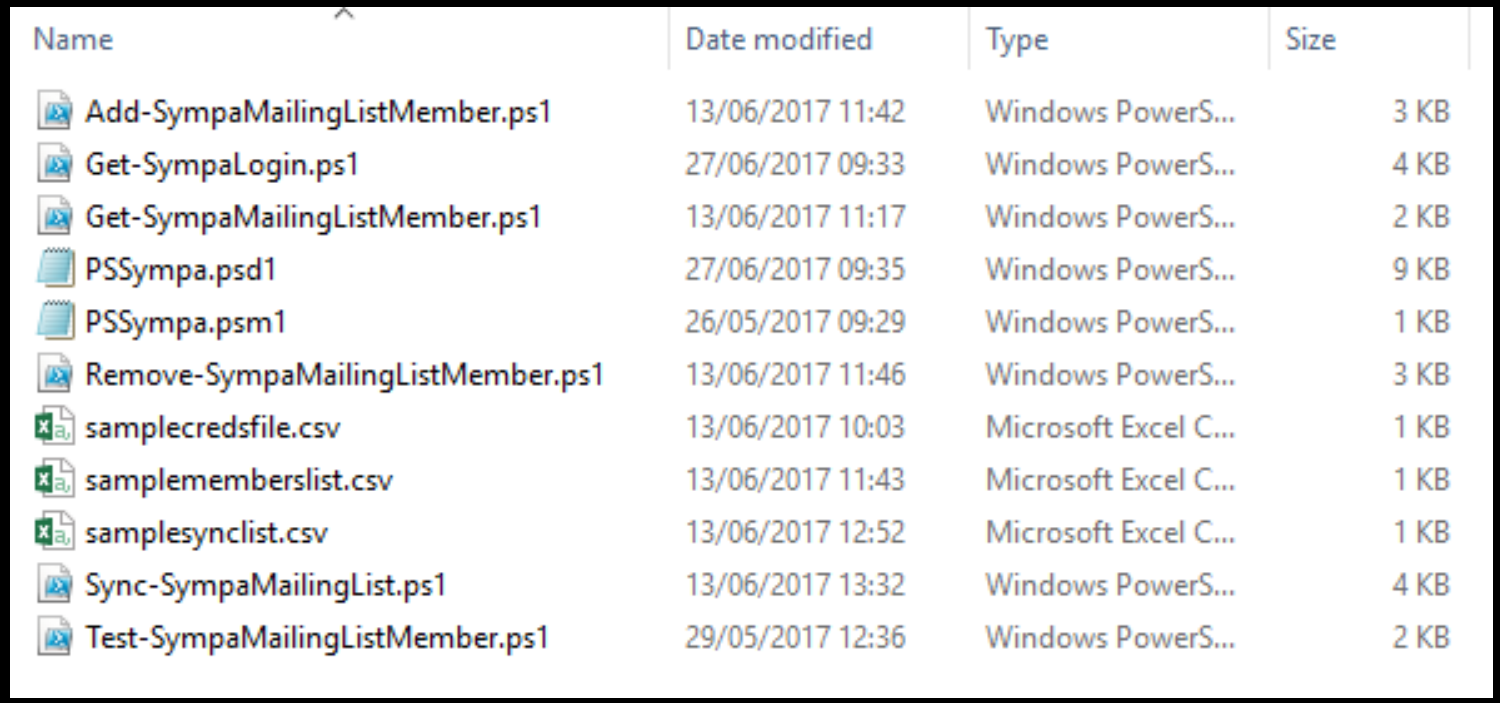
Run Insert Copy

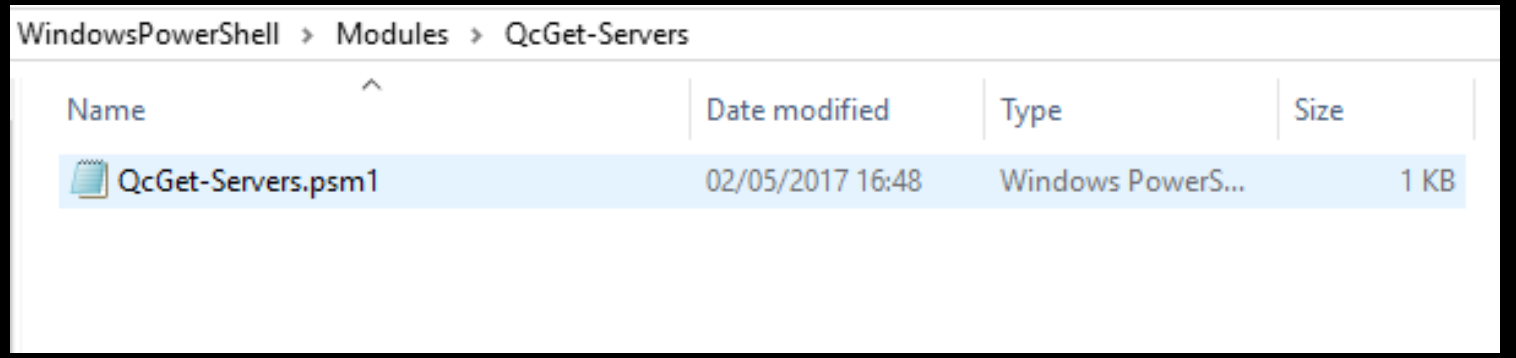
Stopped Ln 6 Col 21 100%

Type here to search 02:29 12/12/2017


WindowsPowerShell
Name
Date modified
Type
Size
Modules
20/11/2017 10:21
File folder
Scripts
12/12/2017 01:33
File folder
Snippets
25/05/2017 12:53
File folder
profile.ps1
21/04/2017 13:12
Windows PowerS...
1 KB
WARNING! Read Me.txt
21/04/2017 09:44
TXT File
1 KB


WindowsPowerShell › Modules
Name
Date modified
Type
Size
PSOpenVAS 15/08/2017 13:03 File folder
PSSympa 13/06/2017 14:10 File folder
QcDelete-TestVm 05/05/2017 12:52 File folder
QcGet-Clients 05/05/2017 09:23 File folder
QcGet-ComputerManager 27/06/2017 11:00 File folder
QcGet-Cud 16/06/2017 13:46 File folder
QcGet-DeployedToners 13/04/2017 12:41 File folder
QcGet-FreshdeskTicketsIt 18/05/2017 10:11 File folder
QcGet-FreshDeskUsers 18/05/2017 10:11 File folder
QcGet-GlobalLog 03/05/2017 15:32 File folder
QcGet-HostnamePort 13/04/2017 14:55 File folder
QcGet-IPMAC 22/04/2017 13:28 File folder
QcGet-MacAddressPort 13/04/2017 14:39 File folder
QcGet-Net2Permissions 28/08/2017 11:18 File folder
QcGet-Net2UsersAtAccessLevel 30/08/2017 10:32 File folder
QcGet-Net2UsersAtAccessLevelReport 30/08/2017 11:44 File folder
QcGet-Net2UserswithIndividualPermissio... 28/08/2017 14:10 File folder
QcGet-PrintEvents 16/05/2017 16:28 File folder
QcGet-RemoteAssistance 24/04/2017 12:55 File folder
QcGet-RemoteLoggedInUsers 27/04/2017 16:18 File folder
QcGet-Servers 02/05/2017 14:48 File folder
QcGet-UsernamePort 13/04/2017 14:51 File folder
QcGet-WuRebootStatus 17/04/2017 13:18 File folder
QcNew-ContractorMac 30/05/2017 13:15 File folder
QcNew-ContractorUser 30/05/2017 12:08 File folder
QcNew-FellowStaffMac 30/05/2017 12:14 File folder
QcNew-FellowStaffUser 05/05/2017 16:15 File folder
QcNew-FreshDeskTicketIt 18/05/2017 10:11 File folder
QcNew-FreshdeskTicketWorks 30/06/2017 14:34 File folder
QcNew-GlobalLogEntry 28/04/2017 10:00 File folder
QcNew-ImageFdTicket 05/06/2017 09:58 File folder
QcNew-ImageMac 15/05/2017 16:07 File folder
QcNew-IoTMac 30/05/2017 12:12 File folder
QcNew-OpenVASTarget 21/09/2017 14:57 File folder


Name
Date modified
Type
Size
Add-SympaMailingListMember.ps1 13/06/2017 11:42 Windows PowerS... 3 KB
Get-SympaLogin.ps1 27/06/2017 09:33 Windows PowerS... 4 KB
Get-SympaMailingListMember.ps1 13/06/2017 11:17 Windows PowerS... 2 KB
PSSympa.psd1 27/06/2017 09:35 Windows PowerS... 9 KB
PSSympa.psm1 26/05/2017 09:29 Windows PowerS... 1 KB
Remove-SympaMailingListMember.ps1 13/06/2017 11:46 Windows PowerS... 3 KB
samplecredsfile.csv 13/06/2017 10:03 Microsoft Excel C... 1 KB
samplememberslist.csv 13/06/2017 11:43 Microsoft Excel C... 1 KB
samplesynclist.csv 13/06/2017 12:52 Microsoft Excel C... 1 KB
Sync-SympaMailingList.ps1 13/06/2017 13:32 Windows PowerS... 4 KB
Test-SympaMailingListMember.ps1 29/05/2017 12:36 Windows PowerS... 2 KB


WindowsPowerShell › Modules › QcGet-Servers
Name
Date modified
Type
Size
QcGet-Servers.psm1 02/05/2017 16:48 Windows PowerS... 1 KB

```powershell
#Try to connect to the central store, if you can then copy down the latest files, if not then report that you cannot connect to the central store
if($(Get-Item -Path \\queens.ox.ac.uk\softwaredeployment\PowerShell\For-AdminPCs -ErrorAction SilentlyContinue).Exists -eq $True){
    #Tell the user what is going on
    Write-Host "Copying down any changed files from the central PowerShell store to the local store" -ForegroundColor Green

    #Copy modules and scripts from the central store, do not display the list of folders copied or the headers (summary only)
    Robocopy.exe "\\queens.ox.ac.uk\softwaredeployment\PowerShell\For-AdminPCs\" "$($env:USERPROFILE)\Documents\WindowsPowerShell\" /mir /njh /ndl

    #Report all done
    Write-Host "If new files were copied please close and reopen PowerShell to use the new/improved functionality" -ForegroundColor Green
}
else{
    Write-Host "Cannot connect to central PowerShell store" -ForegroundColor Red
}
```

```powershell
function QcNew-ContractorMac
{
    <#
    .Synopsis
       This script can be used to grant MAC authenticated 802.1x access to the network for contractors devices (e.g. a laptop) on a daily basis.
    .DESCRIPTION
       In a world where contractors come and go this script ensures that they only have access to the network for the time that they are on site.
    .EXAMPLE
       This example will create an account for the MAC address aabbccddeehh for James Preston which is valid for 7 days. The account will be placed into VLAN4.

       QcNew-ContractorMac -MACAddress aabbccddeehh -Forename James -Surname Preston -Company 'The Queens College' -DeviceDescription Laptop -Email james.preston@queens.ox.ac.uk -ValidDays
    #>
param(
    [Parameter(Mandatory=$true)]
    [string]$MACAddress,
    [Parameter(Mandatory=$true)]
    [String]$Forename,
    [Parameter(Mandatory=$true)]
    [String]$Surname,
    [Parameter(Mandatory=$true)]
    [String]$Company,
    [Parameter(Mandatory=$true,HelpMessage="e.g. Laptop")]
    [String]$DeviceDescription,
    [Parameter(Mandatory=$true)]
    [String]$Email,
    [Parameter(Mandatory=$true)]
    [Int]$ValidDays,
    [Parameter(Mandatory=$true,HelpMessage="Set the network that the Mac address will belong to")]
    [ValidateSet("Internet Access (VLAN11)", "Internal Network (VLAN4)", "BMS Network (VLAN12)")]
    [String]$NetworkAccess

    )

    Import-Module ActiveDirectory

    switch ($NetworkAccess)
    {
        "Internet Access (VLAN11)" {$NetworkId = "14612"}
        "Internal Network (VLAN4)" {$NetworkId = "515"}
        "BMS Network (VLAN12)" {$NetworkId = "14719"}
    }

    switch ($NetworkAccess)
    {
        "Internet Access (VLAN11)" {$NetworkGroup = "MAC Authenticated IoT"}
        "Internal Network (VLAN4)" {$NetworkGroup = "Domain Computers"}
        "BMS Network (VLAN12)" {$NetworkGroup = "MAC Authenticated Building Management System VLAN12"}
    }

    #Check the MAC Address is 12 letters long, if not quit the script
    if($MACAddress.Length -ne 12){
        Write-Host "MAC Address length must be precisely 12 letters long (no delimiters), double check the address and try again" -ForegroundColor Red
        exit
    }

    #Build the accounts properties
```

```
78      Write-Host "Sleeping for 10 seconds to allow AD replication to catch up"
79      Start-Sleep -Seconds 10
80
81      #Add the user to the right security group for network access
82      Add-ADGroupMember -Identity $NetworkGroup -Members $sAMAccountName
83
84      Write-Host "Sleeping for 10 seconds to allow AD replication to catch up"
85      Start-Sleep -Seconds 10
86
87      #Set the users primary group to the network access group
88      Get-ADUser $sAMAccountName | Set-ADObject -Replace @{primaryGroupID=$NetworkId} -Confirm:$false
89
90      Write-Host "Sleeping for 10 seconds to allow AD replication to catch up"
91      Start-Sleep -Seconds 10
92
93      #Remove the user from the Domain Users security group
94      Remove-ADGroupMember -Identity "Domain Users" -Members $sAMAccountName -Confirm:$false
95
96  }
```

```powershell
function QcGet-MacAddressPort
{
param(
    [Parameter(Mandatory=$true,HelpMessage="Enter the MAC Address that you would like to lookup in the format aabbccddeeff")]
    [string]$MacAddress,
    [Parameter(Mandatory=$true,HelpMessage="Enter the number of results to return")]
    [int]$ResultsToReturn
    )

$GetData = @"
SELECT TOP $ResultsToReturn [timestamp]
      ,[NAS_Identifier]
      ,[NAS_Port]
      ,[NP_Policy_Name]
      ,[Proxy_Policy_Name]
FROM [NPSODBC].[dbo].[accounting_data]
where [User_Name] = '$($MacAddress.ToLower())'
order by [timestamp] DESC
"@
    Write-Host "Query execution time is typically 15 seconds" -ForegroundColor Yellow
    Invoke-Sqlcmd -ServerInstance QC-vSQL02 -Database "NPSODBC" -Query $GetData -OutputAs DataRows
}
```

```powershell
function QcGet-PrintEvents
{
    #The event logs we want to get
    $LogToGet = "Microsoft-Windows-PrintService/Operational"

    Get-WinEvent -ComputerName QC-vPRINT01 -LogName $LogToGet -MaxEvents 50
}
```

```powershell
function QcNew-TestVm
{
param(
    [Parameter(Mandatory=$true,HelpMessage="Enter a name for your VM")]
    [String]$VmName
    )
    #Tell the user what is going on
    Write-Host "I'm going to make a VM with name Test-$VmName on QC-HYPERV01 with 4GB RAM, 4 vCPUs, 127GB HDD connected to VLAN4"

    #Instruct the Hyper-V test host to create the VM
    Invoke-Command -ComputerName QC-HYPERV01 -ScriptBlock {
        $VmName = "Test-$($args[0])"
        New-VM -Name $VmName -MemoryStartupBytes 4GB  -Generation 2 -SwitchName 'QC Network' -BootDevice NetworkAdapter -NewVHDPath "D:\Hyper-V\Virtual Hard Disks\$VmName.vhdx" -NewVHDSize
        Set-VM -Name $VmName -ProcessorCount 4
        Set-VMNetworkAdapterVlan -VMName $VmName -Access -VlanId 4
    } -ArgumentList $VmName
}
```

Untitled1.ps1*

```
1  QcNew-TestVm Cheese
```

```
Copying down any changed files from the central PowerShell store to the local store

    *EXTRA Dir        -1    C:\Users\quee3211\Documents\WindowsPowerShell\Scripts\InstalledScriptInfos\

------------------------------------------------------------------------------

               Total    Copied   Skipped  Mismatch    FAILED    Extras
    Dir :        124         0       124         0         0         1
   File :        172         0       172         0         0         0
   Bytes :   206.9 k         0   206.9 k         0         0         0
   Times :   0:00:14   0:00:00                       0:00:00   0:00:14
   Ended : 12 December 2017 02:39:05

If new files were copied please close and reopen PowerShell to use the new/improved functionality
PS C:\Users\quee3211>
```

Commands

Modules: All Refresh

Name:

A:
Add-ADCentralAccessPolicyMember
Add-ADComputerServiceAccount
Add-ADDomainControllerPasswordReplicationPolicy
Add-ADFineGrainedPasswordPolicySubject
Add-ADGroupMember
Add-ADPrincipalGroupMembership
Add-ADResourcePropertyListMember
Add-AppvClientConnectionGroup
Add-AppvClientPackage
Add-AppvPublishingServer
Add-AppxPackage
Add-AppxProvisionedPackage
Add-AppxVolume
Add-BCDataCacheExtension
Add-BgpCustomRoute
Add-BgpPeer
Add-BgpRouteAggregate
Add-BgpRouter
Add-BgpRoutingPolicy
Add-BgpRoutingPolicyForPeer
Add-BitLockerKeyProtector
Add-BitsFile
Add-CauClusterRole
Add-CertificateEnrollmentPolicyServer
Add-ClusterCheckpoint
Add-ClusterDisk
Add-ClusterFileServerRole
Add-ClusterGenericApplicationRole
Add-ClusterGenericScriptRole
Add-ClusterGenericServiceRole
Add-ClusterGroup

Run Insert Copy

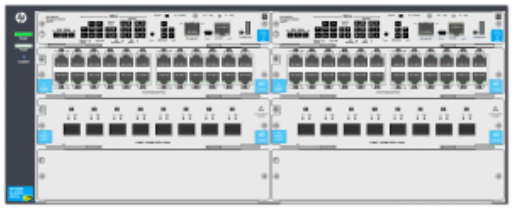
```powershell
function QcTest-GlobalDns
{
param(
    [Parameter(Mandatory=$true,HelpMessage="Enter the hostname that you want to test")]
    [String]$TestHostName
    )
    $DnsServers = "qc-vdc01.queens.ox.ac.uk","qc-vdc02.queens.ox.ac.uk","resolver0.dns.ox.ac.uk","resolver1.dns.ox.ac.uk","google-public-dns-a.google.com","google-public-dns-b.google.com"
    Write-Host "Testing $($TestHostName.ToLower())" -ForegroundColor Green

    foreach($DnsHost in $DnsServers){
    Resolve-DnsName -Name $TestHostName -Server $DnsHost | Add-Member -MemberType NoteProperty -Name DnsHost -Value $DnsHost -PassThru | Select-Object -Property DnsHost, Address, Type
    }
}
```

http://myworldofit.net/?p=9127



# Automated backup for your network switches with WinSCP and PowerShell

Published 2 November, 2015 | By James Preston

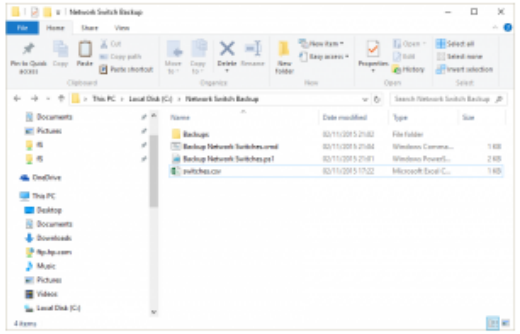


Although it may not be the most glamorous side of IT every sysadmin will appreciate the value of a rock solid backup system. All too often though these systems do not extend down to the 'embedded' systems like network switches and firewalls.

However with a little WinSCP (and its fantastic .NET assembly automation package) and PowerShell combined its pretty easy to cook up something that is 100% less of the cost of any management solution.

This guide shows how to **setup the backup of a HP ProCurve switch** (I've tested it with the ProCurve 8200 series, 5400 series the 2920s, a 2626 and a 2530 all of which were running the most recent firmware) although it should be a simple matter of changing the relevant paths to make it work with other manufacturers kit (e.g. Cisco).

## Download Source Files

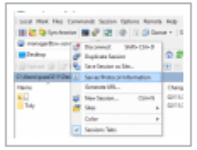


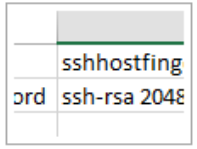
First up grab the source files from the link below and extract the contents to C:\Network Switch Backup (you can use any other path but will just need to update the paths inside the PowerShell) you should then have a folder which contains a .cmd file, a .ps1, a sample .csv and a sub folder called Backups.

**Network Switch Backup** (1.7 KiB, 1,304 hits)

## Getting your Switch ready and filling out the CSV

Each switch will now need ip ssh and ip ssh filetransfer running on it through the CLI (if its not already setup); be sure to set a manager password (if you haven't done so already!) as well. In addition you will need to find the Server host key fingerprint for each switch; the screen shots below show one way of doing this.





```powershell
#Created by: James Preston of The Queen's College, Oxford
#Version: 1.0 on 02/11/2015 17:05
#Website: myworldofit.net

#Load the .NET assembly for WinSCP
Add-Type -Path "C:\Program Files (x86)\WinSCP Automation\WinSCPnet.dll"

#Import the CSV containing switch details and store it in a variable
$switches = Import-Csv -Path "C:\Network Switch Backup\switches.csv"

#Get the current system date in the format year/month/date which will be used to name the backup files
$date = Get-Date -Format yyyy-M-d

#Loop over the lines in the CSV
Foreach ($line in $switches) {

#Define the folder to store the output in and create it if it does not exist (if the folder exists already this will gener
$outputfolder = "C:\Network Switch Backup\Backups\" + $line.hostname + "\"
New-Item $outputfolder -ItemType Directory

#Define the path to store the result of the download
$outputpath = $outputfolder + $date

#Store the session details
$sessionOptions = New-Object WinSCP.SessionOptions
$sessionOptions.Protocol = [WinSCP.Protocol]::Sftp
$sessionOptions.HostName = $line.hostname
$sessionOptions.UserName = $line.username
$sessionOptions.Password = $line.password
$sessionOptions.SshHostKeyFingerprint = $line.sshhostfingerprint
$session = New-Object WinSCP.Session

#Connect to the host
$session.Open($sessionOptions)

#Define the transfer options
$transferOptions = New-Object WinSCP.TransferOptions
$transferOptions.TransferMode = [WinSCP.TransferMode]::Binary

#Download the startup-config (the result of the last 'write memory' from the switches CLI) and save it to the outputpath
$transferResult = $session.GetFiles("/cfg/startup-config", $outputpath, $False, $transferOptions)

#Disconnect from the server
$session.Dispose()

}
```

| Name | Status | Triggers | Next Run Time | Last Run Time |
|---|---|---|---|---|
| Asset Upload | Ready | Multiple triggers defined | 12/12/2017 08:31:13 | 11/12/2017 11:32:16 |
| Automation - AD Account Creation | Ready | At 07:00 every day | 12/12/2017 07:00:00 | 11/12/2017 07:00:00 |
| Automation - Backup Reports | Ready | At 07:00 every day | 12/12/2017 07:00:00 | 11/12/2017 07:00:00 |
| Automation - Export Student Special Diets | Ready | At 07:30 every day | 12/12/2017 07:30:00 | 11/12/2017 07:30:01 |
| Automation - Freshdesk IT Daily Tasks | Ready | At 07:30 every day | 12/12/2017 07:30:00 | 11/12/2017 07:30:17 |
| Automation - Freshdesk OLDS Daily Tasks | Ready | At 08:00 every day | 12/12/2017 08:00:00 | 11/12/2017 08:00:00 |
| Automation - Get Privileged Groups Changes | Ready | At 00:00 every day - After triggered, repeat every 1 hour for a duration of 1 day. | 12/12/2017 03:00:00 | 12/12/2017 02:00:00 |
| Automation - IT Office Daily Report | Ready | At 07:30 every day | 12/12/2017 07:30:00 | 11/12/2017 07:30:01 |
| Automation - RSS Reader | Ready | At 08:00 every day - After triggered, repeat every 1 hour for a duration of 08:00:00. | 12/12/2017 08:00:00 | 11/12/2017 16:00:00 |
| Automation - Update Computer Description in AD | Ready | At 07:30 every day | 12/12/2017 07:30:00 | 11/12/2017 07:30:01 |
| Automation - Write Network Switch Changes to GlobalLog | Ready | At 17:00 every day | 12/12/2017 17:00:00 | 11/12/2017 17:00:00 |
| Backup Alacer | Ready | At 00:30 every day | 13/12/2017 00:30:00 | 12/12/2017 00:30:00 |
| Backup Network Switches | Ready | At 16:00 every day | 12/12/2017 16:00:00 | 11/12/2017 16:00:00 |
| Backup Uniflow | Ready | At 23:30 every day | 12/12/2017 23:30:00 | 11/12/2017 23:30:00 |
| Get Windows Update Reboot Status | Ready | | | 30/05/2017 12:30:40 |
| Optimize Start Menu Cache Files-S-1-5-21-1343024091-78... | Disabled | When computer is idle | | 04/11/2017 08:24:19 |
| Run Windows Update | Ready | | | 15/11/2017 13:06:38 |
| Shutdown | Ready | At 09:06 on 22/06/2017 | | 22/06/2017 09:06:00 |
| Sophos Cloud Scheduled Scan | Ready | At 00:00 every Saturday of every week, starting 09/12/2017 | 16/12/2017 00:00:00 | 09/12/2017 00:00:00 |

# PowerShell Gallery

https://www.powershellgallery.com

Register | Sign in

Home | Items | Publish | Statistics | Documentation | Support

Search Items

## Welcome to the PowerShell Gallery

The PowerShell Gallery is the central repository for PowerShell content. You can find new PowerShell commands or Desired State Configuration (DSC) resources in the Gallery.

## Getting Started with the Gallery

**Installing items from the Gallery requires the latest version of the PowerShellGet module.**

Get Latest PowerShellGet

For PowerShell 5.0 and up.

To see all options for installing PowerShellGet, see our documentation or the PowerShellGet Github repository.

With the latest PowerShellGet module, you can:

- Search through items in the Gallery with Find-Module and Find-Script
- Save items to your system from the Gallery with Save-Module and Save-Script
- Install items from the Gallery with Install-Module and Install-Script
- Upload items to the Gallery with Publish-Module and Publish-Script
- Add your own custom repository with Register-PSRepository

Check out our documentation for more information on how to use PowerShellGet commands with the Gallery. You can also run *Update-Help -Module PowerShellGet* to install local help for these commands.

**Unique Items**
**2,595**

**Total Item Downloads**
**68,824,817**

**Total Items**
**14,001**

## Got a question? Have feedback?

More information about the PowerShell Gallery and PowerShellGet can be found in our documentation. Please provide feedback and report issues using UserVoice.

# Search for *mysql* returned 7 items

Displaying results 1 - 7.

Sort By Relevance

Module

## xMySql By: PowerShellTeam

*Latest Version: 2.1.0.0*

Module for installing an instance of mySQL

**15,690 downloads** Tags DesiredStateConfiguration DSC DSCResourceKit DSCResource

Functions Invoke-MySqlCommand Get-MySqlInstallerConsole Get-MySqlExe Get-MySqlVersionInstalled Get-MySqlAllInstalled Get-ShortVersion Read-ErrorFile Get-MySqlPort Get-ArchitectureName DSC Resources xMySqlDatabase xMySqlGrant xMySqlProvision xMySqlServer xMySqlUser

Module

## MySQLCmdlets By: CData

*Latest Version: 17.0.6428.0*

CData Cmdlets for MySQL

**263 downloads** Tags CData Cmdlets MySQL Cmdlets Connect-MySQL Disconnect-MySQL Invoke-MySQL Select-MySQL Add-MySQL Update-MySQL Remove-MySQL Sync-MySQL Get-License

Script

## Invoke-MySqlQuery By: ramblingcookiemonster

*Latest Version: 1.0.0*

Runs a SQL script against a MySQL instance. Requires the ADO.NET driver for MySQL - http://dev.mysql.com/downloads/connector/net/

**149 downloads** Tags MySql Query Sql Functions Invoke-MySQLQuery

# Search for *vmware* returned 44 items

Displaying results 1 - 20. Sort By Relevance

## VMware.VimAutomation.Core By: VMware

Module

*Latest Version: 6.5.2.6234650*

This Windows PowerShell module contains Windows PowerShell cmdlets for managing vSphere.

**107,888 downloads** Cmdlets Add-PassthroughDevice Add-VirtualSwitchPhysicalNetworkAdapter Add-VMHost Add-VMHostNtpServer Connect-VIServer Copy-DatastoreItem Copy-HardDisk Copy-VMGuestFile Disconnect-VIServer Dismount-Tools Export-VApp ... Functions HookGetViewAutoCompleter

## VMware.VimAutomation.Common By: VMware

Module

*Latest Version: 6.5.4.6979861*

This Windows PowerShell module contains functionality required by multiple PowerCLI modules.

**99,262 downloads**

## VMware.VimAutomation.Sdk By: VMware

Module

*Latest Version: 1.0.0.5334677*

This Windows PowerShell module contains PowerCLI Sdk.

**99,208 downloads** Functions Get-PSVersion Get-InstallPath

## VMware.VimAutomation.Cis.Core By: VMware

Module

*Latest Version: 6.5.4.6983166*

This Windows PowerShell module contains PowerCLI Cloud Infrastructure Suite cmdlets.

**100,345 downloads** Cmdlets Connect-CisServer Disconnect-CisServer Get-CisService

# Further reading

- Lynda.com https://www.lynda.com/search?q=powershell
- PowerShell Gallery https://www.powershellgallery.com
- myworldofit.net http://myworldofit.net/?s=powershell