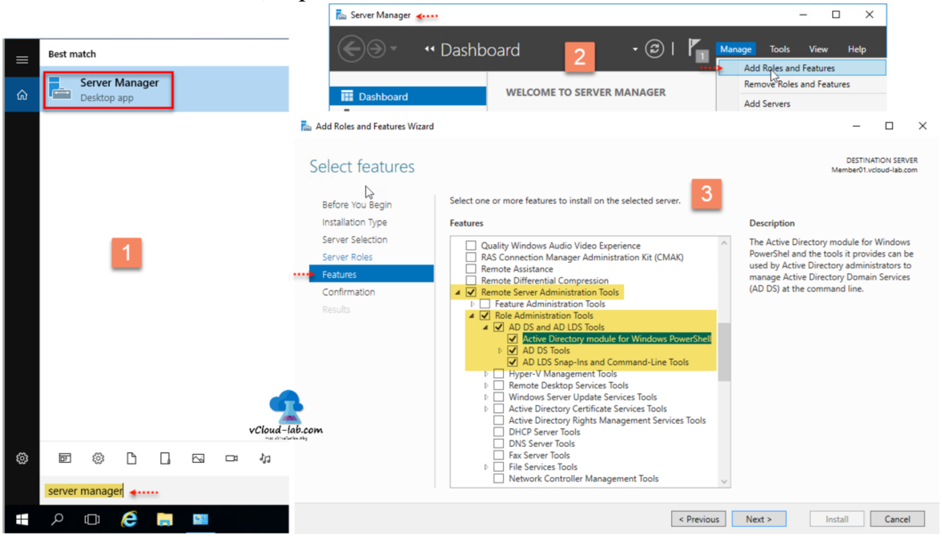
Installing, importing and using any module in powershell

With PowerShell you can achieve great way to automation using additional modules. Modules are additional functionalities or extensions to Powershell. You can also treat them as client softwares, with them you can connect to particular softwares, You will see limitation if you want additional commands and modules does not exist. After installing them you can use related .net objects or adapters. Four types are Powershell modules exist Script, Binary, Manifest, Dynamic. For more detailed information read Microsoft official article.

For more on using script check Different ways to bypass Powershell execution policy :.ps1 cannot be loaded because running scripts is disabled

To demonstrate I will take example of ActiveDirectory module. To install it use, Search and open **Server Manager**, in the **Manage**, click **Add Roles and Features**. From Select **Features** menu, expand **Remote Server Administration Tools**, Under **Role Administration Tools**, Select all **AD DS and AD LDS tools** and Install softwares. Same way you can install other modules and tools from list ie DHCP, Hyper-V and etc, these steps are for Windows 2016 server. If you need it on Windows 10, separate RSAT tools need to be downloaded and installed.



Select one or more features to install on the selected server.

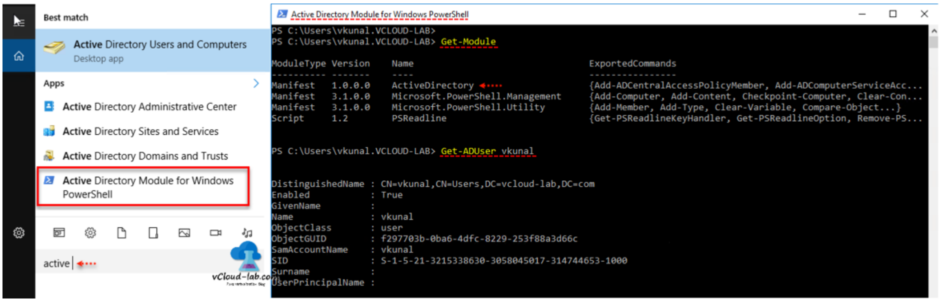
Features

- [ ] Quality Windows Audio Video Experience
- [ ] RAS Connection Manager Administration Kit (CMAK)
- [ ] Remote Assistance
- [ ] Remote Differential Compression
- [✓] Remote Server Administration Tools
  - [ ] Feature Administration Tools
  - [✓] Role Administration Tools
    - [✓] AD DS and AD LDS Tools
      - [✓] Active Directory module for Windows PowerShell
      - [✓] AD DS Tools
      - [✓] AD LDS Snap-Ins and Command-Line Tools
    - [ ] Hyper-V Management Tools
    - [ ] Remote Desktop Services Tools
    - [ ] Windows Server Update Services Tools
    - [ ] Active Directory Certificate Services Tools
    - [ ] Active Directory Rights Management Services Tools
    - [ ] DHCP Server Tools
    - [ ] DNS Server Tools
    - [ ] Fax Server Tools
    - [ ] File Services Tools
    - [ ] Network Controller Management Tools

Description

The Active Directory module for Windows PowerShel and the tools it provides can be used by Active Directory administrators to manage Active Directory Domain Services (AD DS) at the command line.

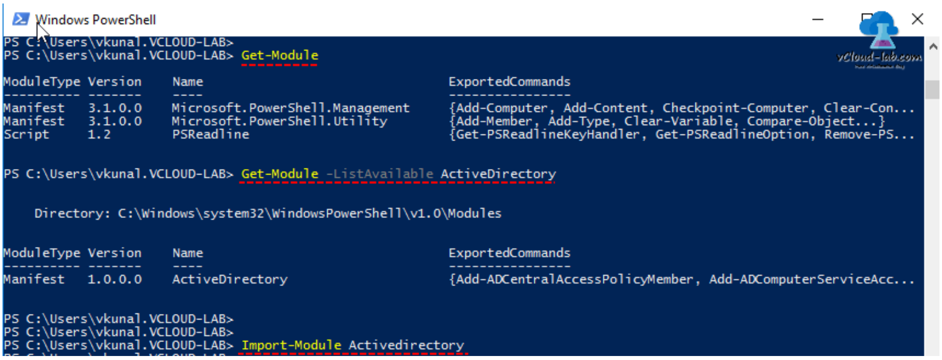
Once **Active Directory modules for windows PowerShell** installed successfully. Same can be searched in find and open it. Module is already imported on the console. To verify the list of loaded modules run command **Get-Module**, ActiveDirectory module is in the list. To view what commands are associated to AD use command **Get-Command -Module ActiveDirectory**. Again to verify everything is OK, I am running one of the AD command **Get-ADUser vkunal**.


Best match
Active Directory Users and Computers
Desktop app
Apps
Active Directory Administrative Center
Active Directory Sites and Services
Active Directory Domains and Trusts
Active Directory Module for Windows PowerShell
active
vCloud-lab.com
Active Directory Module for Windows PowerShell

```
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0.0.0    ActiveDirectory                     {Add-ADCentralAccessPolicyMember, Add-ADComputerServiceAcc...
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Con...
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Script     1.2        PSReadline                          {Get-PSReadlineKeyHandler, Get-PSReadlineOption, Remove-PS...


PS C:\Users\vkunal.VCLOUD-LAB> Get-ADUser vkunal


DistinguishedName : CN=vkunal,CN=Users,DC=vcloud-lab,DC=com
Enabled           : True
GivenName         :
Name              : vkunal
ObjectClass       : user
ObjectGUID        : f297703b-0ba6-4dfc-8229-253f88a3d66c
SamAccountName    : vkunal
SID               : S-1-5-21-3215338630-3058045017-314744653-1000
Surname           :
UserPrincipalName :
```

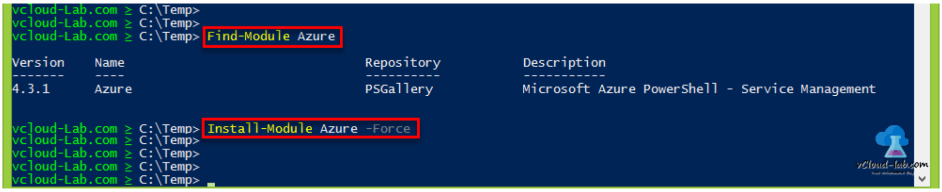
Next step is helpful when you want to run active directory command from normal PowerShell console. By default these steps are not required for newer versions of PowerShell above v4. As whenever you run related commands ie Get-AdUser, module will automatically loaded it exists on machine. But if you want to manually import it run **Import-Module ActiveDirectory**. It is a best practice while writing script import module first instead of depend on auto loading of module.

If you want to list of all the installed modules you can run command **Get-Module -ListAvailable**.



```
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Con...
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Script     1.2        PSReadline                          {Get-PSReadlineKeyHandler, Get-PSReadlineOption, Remove-PS...

PS C:\Users\vkunal.VCLOUD-LAB> Get-Module -ListAvailable ActiveDirectory


    Directory: C:\Windows\system32\WindowsPowerShell\v1.0\Modules


ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0.0.0    ActiveDirectory                     {Add-ADCentralAccessPolicyMember, Add-ADComputerServiceAcc...

PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> Import-Module Activedirectory
```

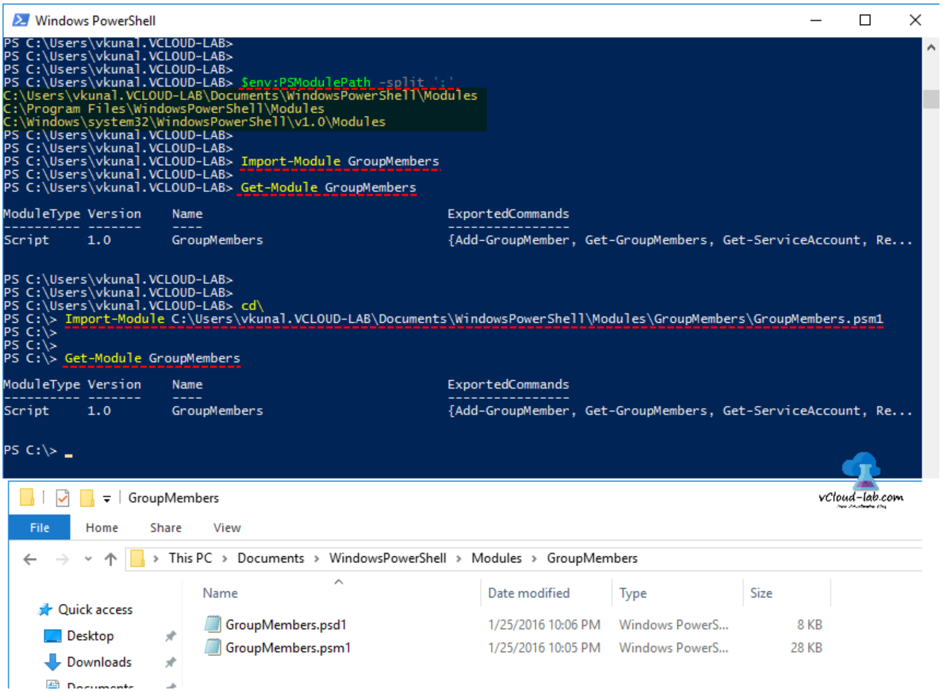
There might be a scenario, your required modules will not exist in RSAT tools and needs to find it online with **Find-Module**. This requires active internet. It will show all community and official modules in the list from online repositories. To download required module use command **Install-Module ModuleName**.



```
vcloud-Lab.com ≥ C:\Temp>
vcloud-Lab.com ≥ C:\Temp>
vcloud-Lab.com ≥ C:\Temp> Find-Module Azure

Version    Name                           Repository           Description
-------    ----                           ----------           -----------
4.3.1      Azure                          PSGallery            Microsoft Azure PowerShell - Service Management

vcloud-Lab.com ≥ C:\Temp> Install-Module Azure -Force
vcloud-Lab.com ≥ C:\Temp>
vcloud-Lab.com ≥ C:\Temp>
vcloud-Lab.com ≥ C:\Temp>
vcloud-Lab.com ≥ C:\Temp> _
```

Module files can be directly copied to modulepath, Path can be viewed using inbuilt command **$env:PSModulePath -split ';'**. There are 3 default paths.
C:\Users\username\Documents\WindowsPowerShell\Modules
C:\Program Files\WindowsPowerShell\Modules
C:\Windows\system32\WindowsPowerShell\v1.0\Modules

I have copy-pasted files to first location **C:\Users\username\Documents\WindowsPowerShell\Modules**, If path does not exist create folders. Note my module folder name and psm1 file has same name and it should be that way to Import it. If in-case you have psm1 module in another location simply provide a file path while importing it. For this step administrator rights are not required.



Windows PowerShell

```
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> $env:PSModulePath -split ';'
C:\Users\vkunal.VCLOUD-LAB\Documents\WindowsPowerShell\Modules
C:\Program Files\WindowsPowerShell\Modules
C:\Windows\system32\WindowsPowerShell\v1.0\Modules
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> Import-Module GroupMembers
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> Get-Module GroupMembers

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     1.0        GroupMembers                        {Add-GroupMember, Get-GroupMembers, Get-ServiceAccount, Re...

PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB>
PS C:\Users\vkunal.VCLOUD-LAB> cd\
PS C:\> Import-Module C:\Users\vkunal.VCLOUD-LAB\Documents\WindowsPowerShell\Modules\GroupMembers\GroupMembers.psm1
PS C:\>
PS C:\>
PS C:\> Get-Module GroupMembers

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     1.0        GroupMembers                        {Add-GroupMember, Get-GroupMembers, Get-ServiceAccount, Re...

PS C:\> _
```

GroupMembers

File Home Share View

This PC › Documents › WindowsPowerShell › Modules › GroupMembers

Name	Date modified	Type	Size
GroupMembers.psd1	1/25/2016 10:06 PM	Windows PowerS...	8 KB
GroupMembers.psm1	1/25/2016 10:05 PM	Windows PowerS...	28 KB

One of the other way to extend Powershell is install 3rd party software check VMWARE VSPHERE POWERCLI INSTALLATION AND CONFIGURATION STEP BY STEP.