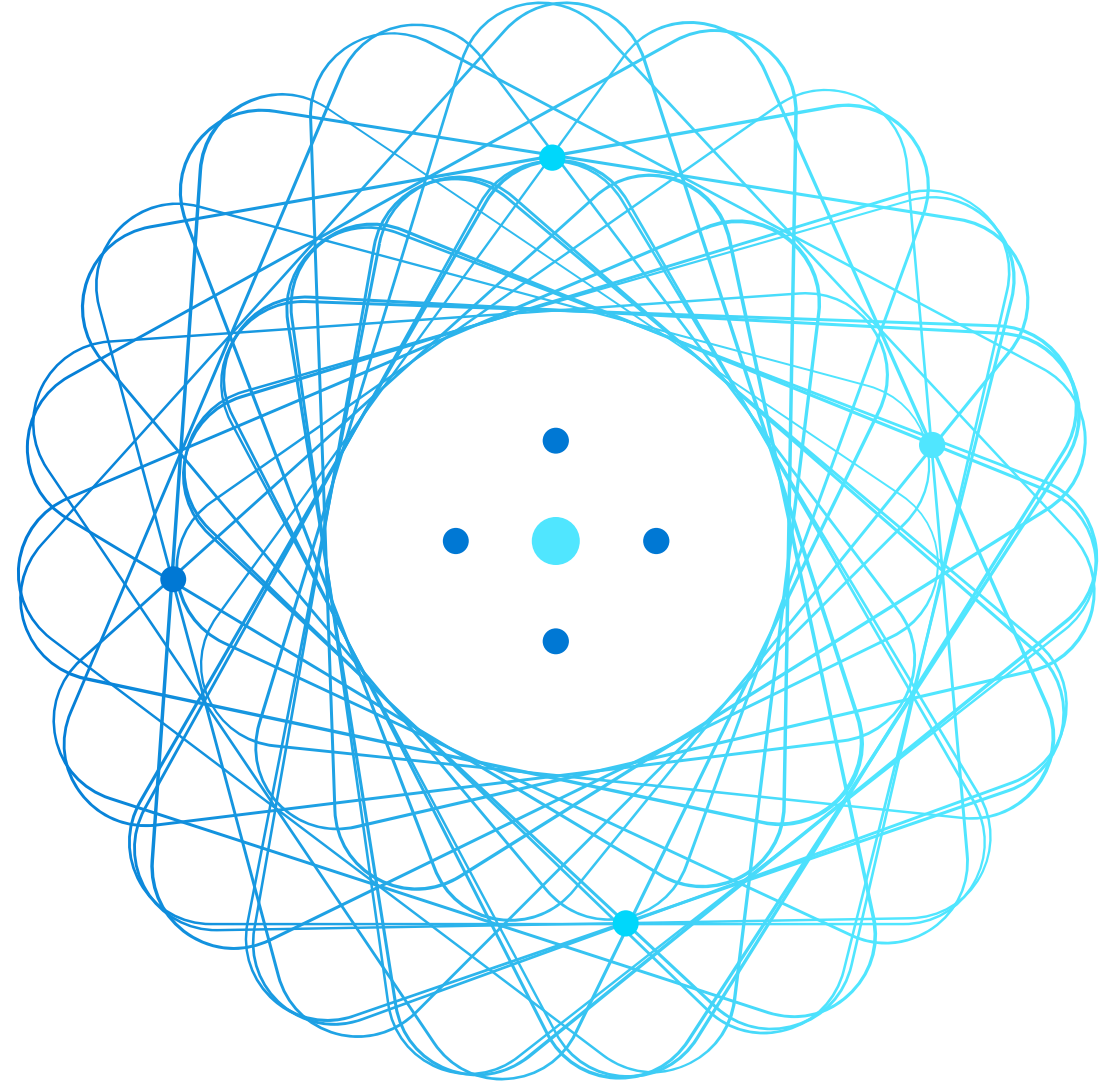


AZ-040 Automating Administration with PowerShell



Learning Path 2: Maintain system administration tasks in Windows PowerShell

Overview

Install-WindowsFeature -Name
Web-Server

In this Learning Path, you'll learn about the cmdlets that you'll commonly use for system administration tasks related to Active Directory, network configuration, server administration, and Windows 10 device administration.

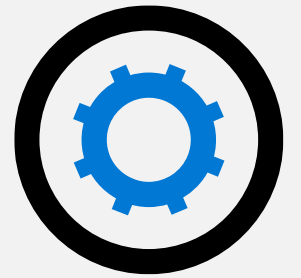
Modules:

- Manage Active Directory Domain Services using PowerShell cmdlets
- Manage network service settings for Windows devices using PowerShell cmdlets
- Manage Windows Server settings using PowerShell cmdlets
- Manage settings for a local Windows machine using PowerShell cmdlets

ping
↳ Test-NetConnection -port 80
-Comment TCP Port HTTP

on Prem AD
adatum.com

Section break 1



Manage Active Directory Domain Services using PowerShell cmdlets

Module overview

In this Module, you'll learn about the cmdlets used for administering AD DS. To find Active Directory cmdlets, search for the prefix "AD," which most Active Directory cmdlets have in the noun part of the cmdlet name.

Units:

- User management cmdlets
- Group management cmdlets
- Demonstration: Managing users and groups
- Cmdlets for managing computer objects
- OU management cmdlets
- Active Directory object cmdlets
- Demonstration: Managing Active Directory objects

User management cmdlets

LDAP —————> PowerShell
Get-ADUser - Filter * - Properties *

Cmdlet	Description
<u>New-ADUser</u>	Creates a user account
<u>Set-ADUser</u>	Modifies properties of a user account
<u>Remove-ADUser</u>	Deletes a user account
Set-ADAccountPassword	Resets the password of a user account
Set-ADAccountExpiration	Modifies the expiration date of a user account
Unlock-ADAccount	Unlocks a user account that's been locked after exceeding the permitted number of incorrect sign-in attempts
Enable-ADAccount	Enables a user account
Disable-ADAccount	Disables a user account

New-ADUser "Ana Bowman" -Department IT

Group management cmdlets

Cmdlet	Description
New-ADGroup	Creates a new group
Set-ADGroup	Modifies properties of a group
Get-ADGroup	Displays properties of a group
Remove-ADGroup	Deletes a group
Add-ADGroupMember	Adds members to a group <i>Member</i>
Get-ADGroupMember	Displays membership of a group
Remove-ADGroupMember	Removes members from a group
Add-ADPrincipalGroupMembership	Adds group membership to an object <i>Member of</i>
Get-ADPrincipalGroupMembership	Displays group membership of an object
Remove-ADPrincipalGroupMembership	Removes group membership from an object

`New-ADGroup -Name "FileServerAdmins" -GroupScope Global`

Demonstration: Managing users and groups

1. Create a new global group in the IT department.
2. Create a new user in the IT department.
3. Add two users from the IT department to the HelpDesk group.
4. Set the address for all HelpDesk group users.
5. Verify the group membership for the new user.
6. Verify the updated user properties.



Cmdlets for managing computer objects

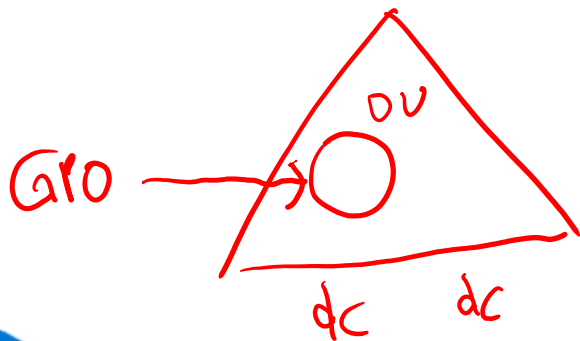
Cmdlet	Description
New-ADComputer	Creates a new computer account
Set-ADComputer	Modifies properties of a computer account
Get-ADComputer	Displays properties of a computer account
Remove-ADComputer	Deletes a computer account
Test-ComputerSecureChannel	Verifies or repairs the trust relationship between a computer and a domain
Reset-ComputerMachinePassword	Resets the password for a computer account

```
New-ADComputer -Name LON-CL10 -Path  
"ou=marketing,dc=adatum,dc=com" -Enabled $true
```

OU management cmdlets

Cmdlet	Description
New-ADOrganizationalUnit	Creates an OU
Set-ADOrganizationalUnit	Modifies properties of an OU
Get-ADOrganizationalUnit	Displays properties of an OU
Remove-ADOrganizationalUnit	Deletes an OU

```
New-ADOrganizationalUnit -Name Sales -Path  
"ou=marketing,dc=adatum,dc=com" -ProtectedFromAccidentalDeletion $true
```



Active Directory object cmdlets

Cmdlet	Description
New-ADObject	Creates a new Active Directory object
Set-ADObject	Modified properties of an Active Directory object
Get-ADObject	Displays properties of an Active Directory object
Remove-ADObject	Deletes an Active Directory object
Rename-ADObject	Renames an Active Directory object
Restore-ADObject	Restores a deleted Active Directory object from the Active Directory recycle bin
Move-ADObject	Moves an Active Directory object from one container to another container
Sync-ADObject	Syncs an Active Directory object between two domain controllers

New-ADObject -Name "AnaBowmancontact" -Type contact

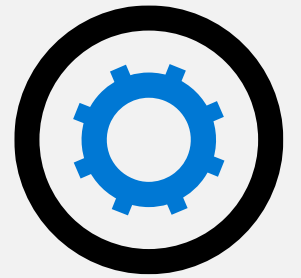
Demonstration: Managing Active Directory objects

In this demonstration, you'll learn how to:

1. Create an Active Directory contact object that has no dedicated cmdlets.
2. Verify the creation of the contact.
3. Manage user properties by using Active Directory object cmdlets.
4. Verify the property changes.
5. Change the name of the HelpDesk group to **SupportTeam**.
6. Verify the HelpDesk group name change.



Section break 2



Manage network service settings for Windows devices using PowerShell cmdlets

ipconfig
nslookup
ping
route print

Get-NetIPConfiguration
Resolve-DNSName
Test-NetConnection
Get-NetRoute

Module overview

In this Module, you'll learn about the PowerShell modules and cmdlets used for configuring network settings for Windows devices.

Units:

- Managing IP addresses
- Managing routing
- Managing DNS clients
- Managing Windows Firewall
- Demonstration: Configuring network settings

Managing IP addresses

Cmdlet	Description
New-NetIPAddress	Creates a new IP address
Set-NetIPAddress	Sets properties of an IP address
Get-NetIPAddress	Displays properties of an IP address
<u>Remove-NetIPAddress</u>	Deletes an IP address

```
New-NetIPAddress -IPAddress 192.168.1.10 -InterfaceAlias "Ethernet" -PrefixLength 24 -DefaultGateway 192.168.1.1
```

- 1) Get-Command
- 2) Get-Help
- 3) Get-Member

gcm

help man

gm

Managing routing

Cmdlet	Description
New-NetRoute	Creates an IP routing table entry
Set-NetRoute	Sets properties of an IP routing table entry
Get-NetRoute	Displays properties of an IP routing table entry
Remove-NetRoute	Deletes an IP routing table entry
Find-NetRoute	Identifies the best local IP address and route to reach a remote address

New-NetRoute -DestinationPrefix 0.0.0.0/24 -InterfaceAlias "Ethernet" -DefaultGateway 192.168.1.1

Default Gw

Managing DNS clients

Cmdlet	Description
Get-DnsClient	Gets details about a network interface on a computer
Set-DnsClient	Set DNS client configuration settings for a network interface
Get-DnsClientServerAddress	Gets the DNS server address settings for a network interface
Set-DnsClientServerAddress	Sets the DNS server address for a network interface 8.8.8.8
Get-DnsClient	Gets details about a network interface on a computer

```
Set-DnsClient -InterfaceAlias Ethernet -ConnectionSpecificSuffix "adatum.com"
```

Resolve - DNS Name

Managing Windows Firewall

Cmdlet	Description
New-NetFirewallRule	Creates a new firewall rule
Set-NetFirewallRule	Sets properties for firewall rules
Get-NetFirewallRule	Gets properties for firewall rules
Remove-NetFirewallRule	Deletes firewall rules
Rename-NetFirewallRule	Renames firewall rules
Copy-NetFirewallRule	Makes a copy of firewall rules
Enable-NetFirewallRule	Enables firewall rules
Disable-NetFirewallRule	Disables firewall rules
Get-NetFirewallProfile	Gets properties for firewall profiles
Set-NetFirewallProfile	Sets properties for firewall profiles

Get - Net Connection Profile
Set

Demonstration: Configuring network settings

In this demonstration, you'll learn how to:

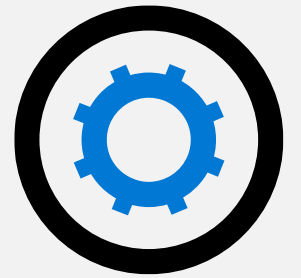
1. Test the network connection to **LON-DC1**.
2. Review the network configuration for **LON-CL1**.
3. Change the client IP address.
4. Change the DNS server for **LON-CL1**.
5. Change the default gateway for **LON-CL1**.
6. Confirm the network configuration changes.
7. Test the effect of the changes.



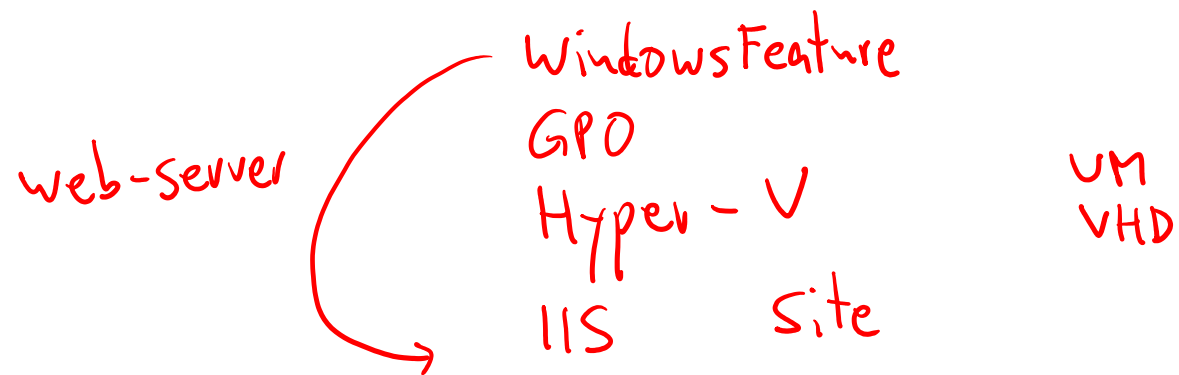
Demonstration: Configuring network settings (Slide 2)



Section break 3



Manage Windows Server settings using PowerShell cmdlets



Module overview

PowerShell is commonly used to perform administration tasks for Windows Server features and services. In this Module, you'll learn about the cmdlets you can use to configure settings related to Group Policy, Server Manager, Hyper-V, and Internet Information Services (IIS).

Units:

- Group Policy management cmdlets
- Server Manager cmdlets
- Hyper-V cmdlets
- IIS administration cmdlets

Group Policy management cmdlets

Cmdlet	Description
New-GPO	Creates a new GPO
Get-GPO	Retrieves a GPO
Set-GPO	Modifies properties of a GPO
Remove-GPO	Deletes a GPO
Rename-GPO	Renames a GPO
Backup-GPO	Creates a backup of a GPO
Copy-GPO	Copies a GPO from one domain to another
Restore-GPO	Restores a GPO from backup files
New-GPLink	Links a GPO to an AD DS container
Import-GPO	Imports GPO settings from a backed-up GPO
Set-GPRegistryValue	Configures one or more registry-based policy settings in a GPO

Server Manager cmdlets

Cmdlet	Description
Get-WindowsFeature	Obtains and displays information about Windows Server roles, services, and features on the local computer
Install-WindowsFeature	Installs roles, services, or features
Uninstall-WindowsFeature	Uninstalls roles, services, or features

Install-WindowsFeature "nlb"

Install-WindowsFeature "web-server"

imperativ

deklarativ



Hyper-V cmdlets

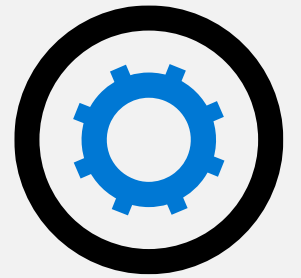
Cmdlet	Description
Get-VM	Gets properties of a VM
Set-VM	Sets properties of a VM
New-VM	Creates a new VM
Start-VM	Starts a VM
Stop-VM	Stops a VM
Restart-VM	Restarts a VM
Suspend-VM	Pauses a VM
Resume-VM	Resumes a paused VM
Import-VM	Imports a VM from a file
Export-VM	Exports a VM to a file
Checkpoint-VM	Creates a checkpoint of a VM

IIS administration cmdlets | ①. Install - Windows Feature Web-server

②.

Cmdlet	Description
New-IISite	Creates a new IIS website
Get-IISite	Gets properties and configuration information about an IIS website
Start-IISite	Starts an existing IIS website on the IIS server
Stop-ISSite	Stops an IIS website
New-WebApplication	Creates a new web application
Remove-WebApplication	Deletes a web application
New-WebAppPool	Creates a new web application pool
Restart-WebAppPool	Restarts a web application pool

Section break 4



Manage settings for a local Windows machine using PowerShell cmdlets

Module overview

In addition to network service and configuration settings, PowerShell is commonly used to configure and manage settings on a local Windows machine. You can use PowerShell cmdlets to perform GUI-based operations more quickly or as a basis to run multiple commands within a script.

In this Module, you'll learn about common PowerShell cmdlets that you can use to perform tasks on a Windows 10 computer.

Units:

- Managing Windows 10 using PowerShell
- Managing permissions with PowerShell

Managing Windows 10 using PowerShell

Cmdlet	Description
Get-ComputerInfo	Retrieves all system and operating system properties from the computer
Get-Service	Retrieves a list of all services on the computer
Get-EventLog	Retrieves events and event logs from local and remote computers (Only available in Windows PowerShell 5.1)
Get-Process	Retrieves a list of all active processes on a local or remote computer
Stop-Service	Stops one or more running services
Stop-Process	Stops one or more running processes
Stop-Computer	Shuts down local and remote computers
Clear-EventLog	Deletes all of the entries from the specified event logs on the local computer or on remote computers
Clear-RecycleBin	Deletes the content of a computer's recycle bin
Restart-Computer	Restarts the operating system on local and remote computers
Restart-Service	Stops and then starts one or more services

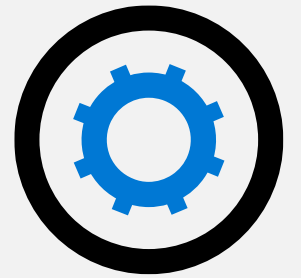
Managing permissions with PowerShell

Cmdlet	Description
Get-Acl	Gets objects that represent the security descriptor of a file or resource. The security descriptor includes the access control lists (ACLs) of the resource. The ACL lists permissions that users and groups have to access the resource.
Set-Acl	Changes the security descriptor of a specified item, such as a file, folder, or a registry key, to match the values in a security descriptor that you supply

To update access permissions:

1. Use **Get-Acl** to retrieve the existing access control list rules for the object.
2. Create a new FileSystemAccessRule to be applied to the object.
3. Add the new rule to the existing ACL permission set.
4. Use **Set-Acl** to apply the new ACL to the existing file or folder.

Section break 5

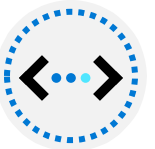


Lab: Performing local system administration with PowerShell

Exercise 1: Creating and managing Active Directory objects



Exercise 2: Configuring network settings on Windows Server



Exercise 3: Creating a website



Sign-in information for the exercises:

Virtual machines:

- **AZ-040T00A-LON-DC1**
- **AZ-040T00A-LON-SVR1**
- **AZ-040T00A-LON-CL1**

Username: **Adatum\Administrator**

Password: **Pa55w.rd**

Lab scenario

You work for Adatum Corporation on the server support team. One of your first assignments is to configure the infrastructure service for a new branch office. You decide to complete the tasks by using Windows PowerShell.

Lab-review questions



In Exercise 1, you created the Active Directory objects first, and then moved them into the OU. Provide an example of how you can create the London Admins group and place them in the OU at the same time.



Which PowerShell cmdlet is used to review the current network configuration on a Windows computer?



In Exercise 3, why did you assign the London site to port 8080?

Lab-review answers



In Exercise 1, you created the Active Directory objects first, and then moved them into the OU. Provide an example of how you can create the London Admins group and place them in the OU at the same time.

New-ADGroup "London Admins" –GroupScope Global –Path "OU=London,DC=Adatum,DC=com"



Which PowerShell cmdlet is used to review the current network configuration on a Windows computer?

Get-NetIPConfiguration



In Exercise 3, why did you assign the London site to port 8080?

After installing the Web Server role, it creates a default web site at port 80. You assigned port 8080 so that both websites are still active. You can confirm this by running the **Get-Website** command and reviewing the output.

References

[PowerShell 7 module compatibility](#)

[Microsoft.PowerShell.Management](#)

