

Admin Server 2 - Configure a Core Install I

Anthony Adams

What is a core install?

A core install is a minimal Windows Server Installation option that provides a fixed subset of roles but does not include:

- Server Graphical Shell
- Microsoft Management Console
- Desktop UI

It includes a limited UI for managing the server. Most management is done locally at a command console (CMD or PWSH) or remotely using management tools.

Is ideally suited to a specific dedicated server role or a combination of roles (IE DNS server, web server).

Why use this? Because it is light weight. Thus the reduced overhead provides more resources for the dedicated roles.

What are the Roles?

Limited to the following Roles:

Active Directory Certificate Services

Active Directory Domain Services

DHCP (Dynamic Host Configuration Protocol) Server

DNS (Domain Name System) Server

File Services (including File Server Resource Manager)

Active Directory Lightweight Directory Services (AD LDS)

Hyper-V (VM)

Print and Document Services

Streaming Media Services

Web Server (including a subset of ASP.NET)

Windows Server Update Server

Active Directory Rights Management Server (Encryption)

Routing and Remote Access Server

Default Shell

- A Windows Server Core install will launch a cmd.exe shell at login.

- The registry editor GUI utility is one of the few installed by default during a Core install.

(<https://www.techtarget.com/searchenterprisedesktop/definition/Windows-Registry-Editor#:~:text=The%20Windows%20Registry%20Editor%20enables,hardware%20or%20software%20level%20configurations.>)

- The default shell can be customized using the HKLM\Software\Microsoft\Windows NT\CurrentVersion\WinLogon\AlternateShells Registry key

Manage a Core Install with PowerShell

We will be using the following PowerShell cmdlets to manage a core server installation:

Get-WindowsEdition

Display information about the windows

editions (DataCenter, standard, etc)

Example:

Windows Server 2019 Datacenter

Max CPU Cores Unlimited

Max Memory 24TB

VM's Unlimited

Get-WMIObject

PowerShell cmdlet to return instances of Windows Management Instrumentation (WMI) Classes

We will be working with the following WMI classes:

Win32_ComputerSystem

Win32_OperatingSystem

Win32_NetworkAdapterConfiguration

Can be used to monitor all the software and hardware configurations for the system. OS, Hard Drives, how much space is available so on.

Get-Member

Returns the properties and methods of an

object.

Get-NetIPAddress

Displays information about IP address configuration.

New-NetIPAddress

Creates a new IP Address and its configuration properties.

Get-DnsClientServerAddress

Displays DNS server addresses associated with the IP configured interfaces on a computer.

Set-DnsClientServerAddress

Configures the DNS server addresses for IP configured interfaces on a computer.

Add-Computer

Can be used to add the local computer to a domain or workgroup.

Format-List

Formats command output as a list of properties.

Can display the default set of properties, specify a list of properties or display all (*) properties of an object.

Stop-Computer

Can be used to shutdown local and remote computers (gracefully).

Restart-Computer

Can be used to reboot local and remote computers.

Set-Location

Sets the current working location to a specified location.

The same as the “cd” command in the cmd.exe shell.

Has been aliased to “cd” in PowerShell.

Variables used in the lab

`$env:SystemEnvironmentVariableName`

Is used when referring to a Windows environment variable such as `$SystemRoot`, `$SystemDrive`

In the lab

Determine network and system configuration information

Modify network configuration

Join the computer to a domain

Install Roles and Features

Manage services and processes