Introduction to Powershell

@iLabAfrica, Strathmore University
John Ombagi

What is PowerShell?

- PowerShell is a cross-platform task automation solution made up of a
  - command-line shell,
  - a scripting language, and
  - a configuration management framework.
- PowerShell runs on Windows, Linux, and macOS.
- PowerShell is also abreviated as PS.

Source: https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.1

PowerShell – CMD Line Shell

- PowerShell is a modern command shell that includes the best features of other popular shells.
  - Unlike most shells that only accept and return text, PowerShell accepts and returns .NET objects.
- The shell includes the following features:
  - Robust command-line history,
  - Tab completion and command prediction (about_PSReadLine),
  - Supports command and parameter aliases,
  - Pipeline for chaining commands,
  - In-console help system, similar to Unix man pages.

PowerShell - Scripting Language

- As a scripting language, PowerShell is commonly used for automating the management of systems.
- It is also used to build, test, and deploy solutions, often in CI/CD environments.
- PowerShell is built on the .NET Common Language Runtime (CLR). All inputs and outputs are .NET objects.
  - No need to parse text output to extract information from output.
  - The PowerShell scripting language includes the following features:

Scripting Language Features

- ***Scripting language includes the following features:***
- Extensible through functions, classes, scripts, and modules.
- Extensible formatting system for easy output.
- Extensible type system for creating dynamic types.
- Built-in support for common data formats like CSV, JSON, and XML.

PowerShell - Configuration Management

- PowerShell Desired State Configuration (DSC) is a management framework in PowerShell that enables you to manage your enterprise infrastructure with configuration as code. With DSC, you can:
  - Create declarative configurations and custom scripts for repeatable deployments.
  - Enforce configuration settings and report on configuration drift.
  - Deploy configuration using push or pull models.

PowerShell command (cmdlet)

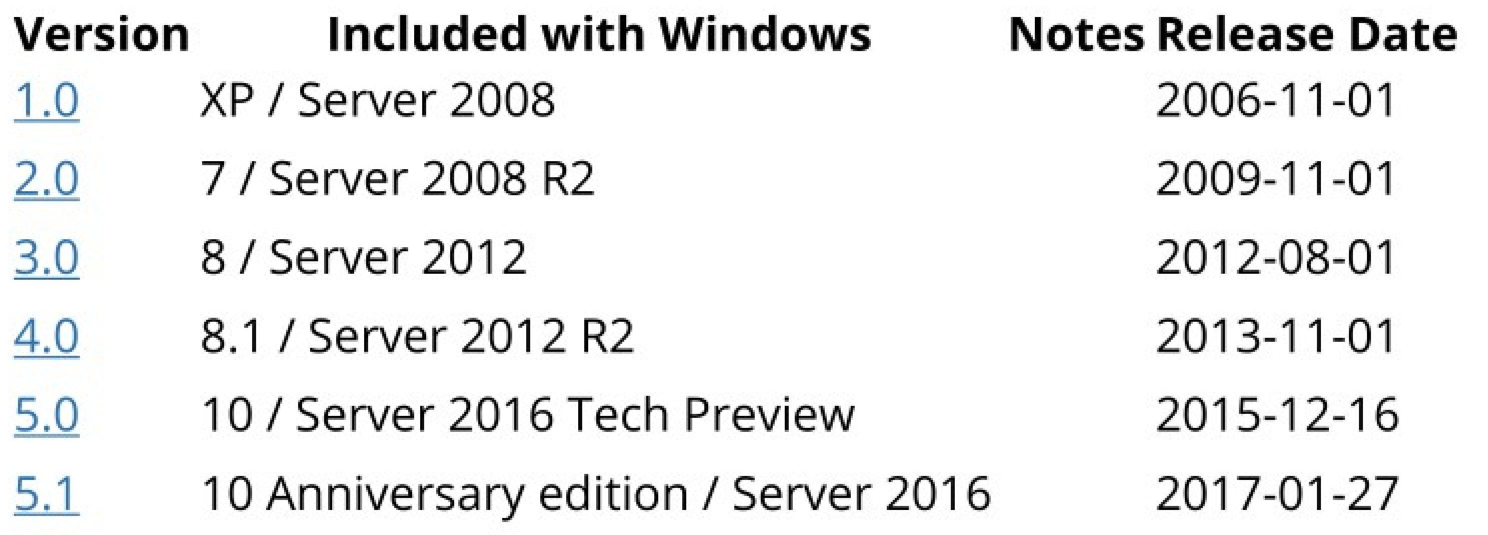
- Commands for PowerShell are known as cmdlets (pronounced command-lets).
  - In addition to cmdlets, PowerShell allows you to run any command available on your system.
- Cmdlets are native PowerShell commands, not stand-alone executables.
- Cmdlets are collected into PowerShell modules that can be loaded on demand.
- Cmdlets can be written in any compiled .NET language or in the PowerShell scripting language itself.

Cmdlet Names

- PowerShell uses a Verb-Noun name pair to name cmdlets.
  - For example, the Get-Command cmdlet included in PowerShell is used to get all the cmdlets that are registered in the command shell.
- The *verb identifies the action* that the cmdlet performs, and the *noun identifies the resource* on which the cmdlet performs its action.

Installing PowerShell on Windows

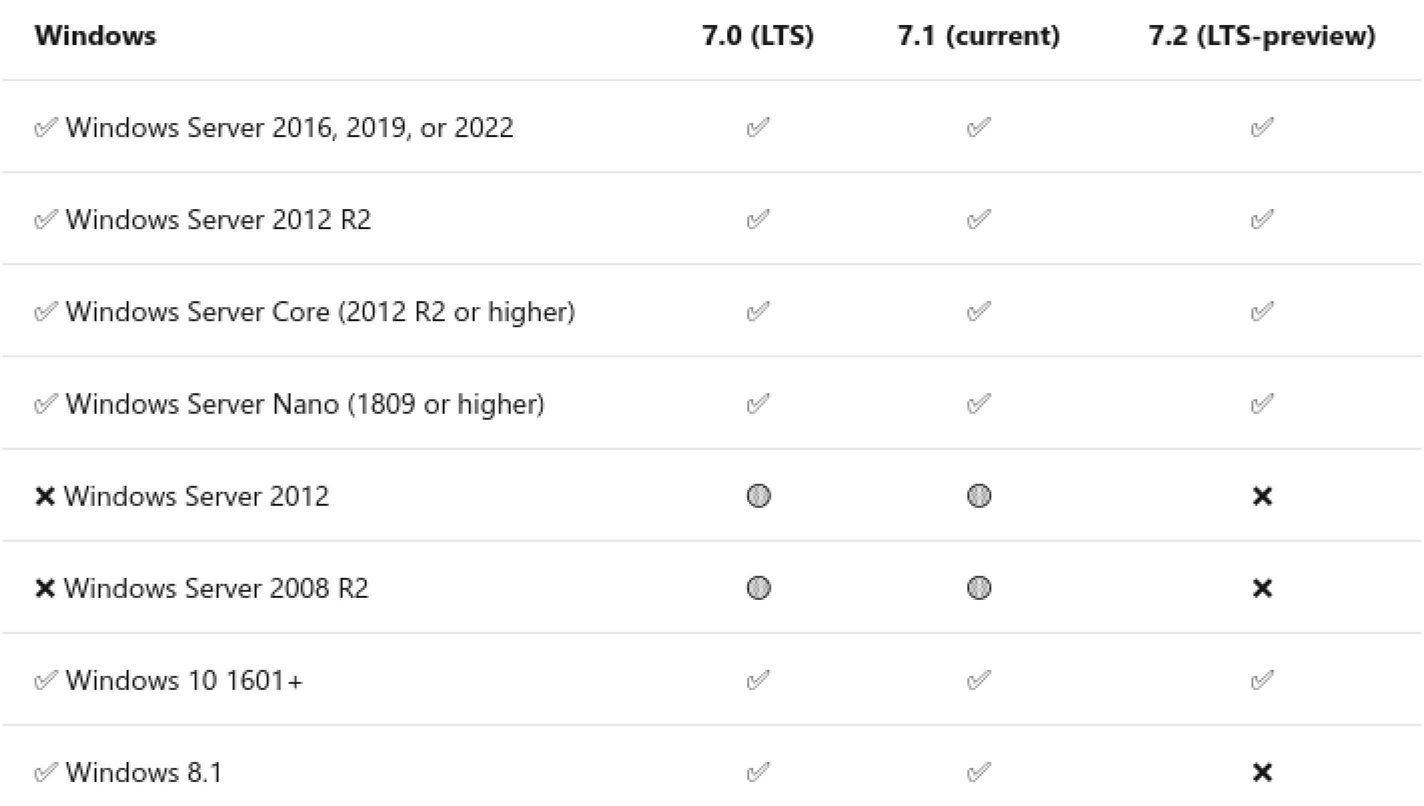
- Previous versions of Windows and compatible versions:



Version	Included with Windows	Notes	Release Date
1.0	XP / Server 2008		2006-11-01
2.0	7 / Server 2008 R2		2009-11-01
3.0	8 / Server 2012		2012-08-01
4.0	8.1 / Server 2012 R2		2013-11-01
5.0	10 / Server 2016 Tech Preview		2015-12-16
5.1	10 Anniversary edition / Server 2016		2017-01-27

Installing PowerShell on Windows

- Latest version of PS is 7.x
- Supported versions of Windows are shown in the table.
- PowerShell has not been tested on Windows 11.



Windows	7.0 (LTS)	7.1 (current)	7.2 (LTS-preview)
✓ Windows Server 2016, 2019, or 2022	✓	✓	✓
✓ Windows Server 2012 R2	✓	✓	✓
✓ Windows Server Core (2012 R2 or higher)	✓	✓	✓
✓ Windows Server Nano (1809 or higher)	✓	✓	✓
✕ Windows Server 2012	●	●	✕
✕ Windows Server 2008 R2	●	●	✕
✓ Windows 10 1601+	✓	✓	✓
✓ Windows 8.1	✓	✓	✕

Installing the MSI package

- To install PowerShell on Windows, download the install package from GitHub.
- Scroll down to the Assets section of the Release page. The Assets section may be collapsed, so you may need to click to expand it.
- The MSI file looks like PowerShell-<version>-win-<os-arch>.msi. For example:
  - PowerShell-7.1.4-win-x64.msi
  - PowerShell-7.1.4-win-x86.msi
-

Note for PowerShell 7.x

- PowerShell 7.1 installs to a new directory and runs side-by-side with Windows PowerShell 5.1.
- PowerShell 7.1 is an in-place upgrade that replaces PowerShell 7.0 and lower.
- If you need to run PowerShell 7.1 side-by-side with other versions, use the ZIP install method to install the other version to a different folder.
-

Installing the ZIP package

- PowerShell binary ZIP archives are provided to enable advanced deployment scenarios. Download one of the following ZIP archives from the current release page.
- Depending on how you download the file you may need to unblock the file using the Unblock-File cmdlet.
- Unzip the contents to the location of your choice and run pwsh.exe from there.
- Unlike installing the MSI packages, installing the ZIP archive doesn't check for prerequisites.

Installation on Debian 10

- Download the Microsoft repository GPG keys
  - wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb
- Register the Microsoft repository GPG keys
  - sudo dpkg -i packages-microsoft-prod.deb
- Update the list of products
  - sudo apt-get update
- Install PowerShell
  - sudo apt-get install -y powershell
- Start PowerShell
  - pwsh

iCAD PowerShell Labs

- Windows 10 Anniversary Edition and Windows Server 2016 both have PowerShell version 5.1 installed by default.
- These two platforms are suitable for the classes.
- Will shall install specific version or make a switch whever needed.
- To check the version of PowerShell you are running:
  - PS C:\> $PSVersionTable
    - If this command returns nothing at all, then you have PowerShell version 1.0 and you'll need to update the version of PowerShell that's installed on your computer.

EOF.

jnyabuti@strathmore.edu