POWERSHELL

>_ Scripting

Intro to Advanced

FULL COURSE • FREE



LONG VIDEO • 25 MIN

POWERSHELL 7 • FULL COURSE

Windows PowerShell Scripting: Intro to Advanced

Commands, the pipeline, scripts, automation and WSL

FREE FULL COURSE • MyWebUniversity.com



Wahidullah Lutfy
Founder, MyWebUniversity • CS Teacher

COURSE ROADMAP

What You Will Learn

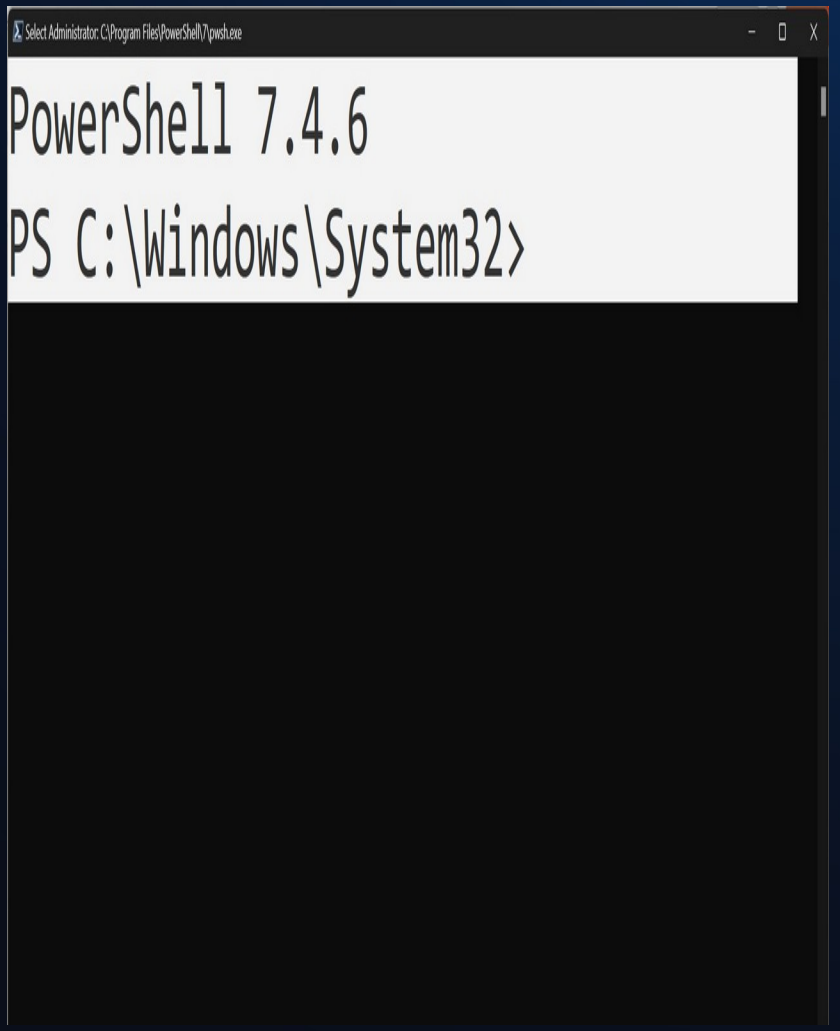
1 · Getting started
What PowerShell is, versions, installing PowerShell 7 and Windows Terminal.

2 · Core skills
Cmdlets, help, objects, the pipeline, files and filtering.

3 · Scripting
Variables, operators, conditions, loops, scripts and functions.

4 · Advanced
Error handling, modules, remoting, jobs, scheduling and REST APIs.

5 · Linux on Windows
Install WSL with PowerShell and run Ubuntu.




Windows PowerShell Scripting • Full Course
PART 1 · GETTING STARTED

GETTING STARTED

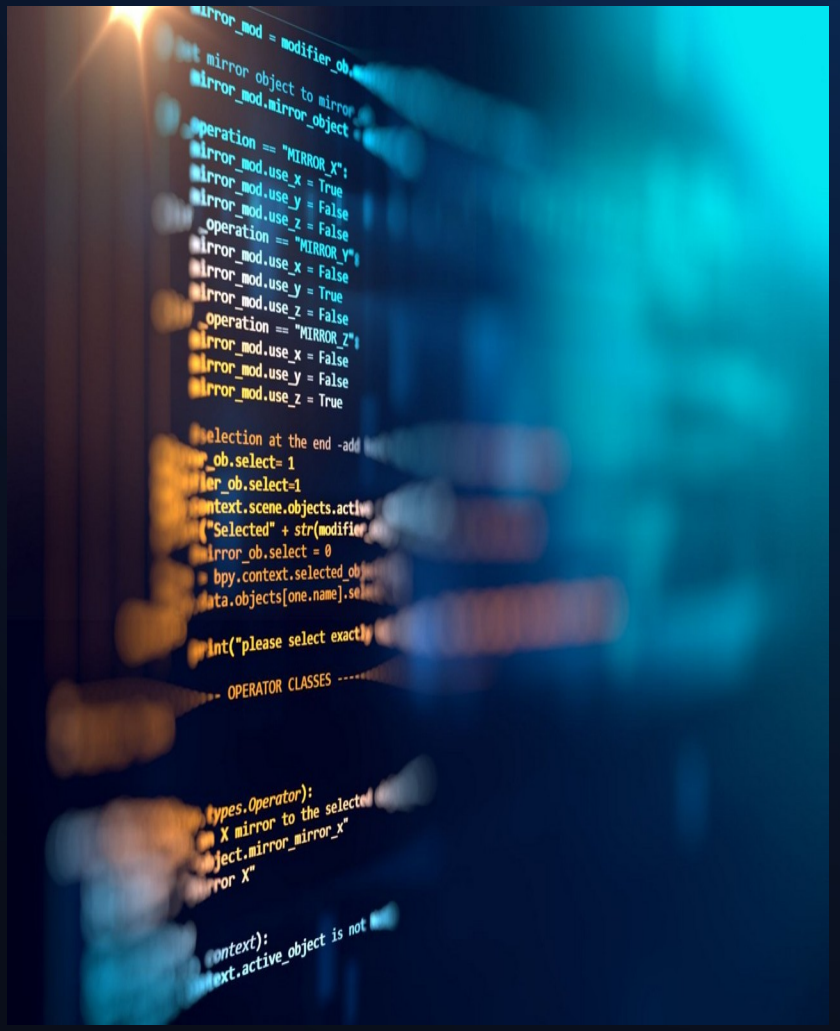
What Is PowerShell?

A shell
Type commands and see results right away, like Command Prompt, but far more powerful.

A scripting language
Save commands in .ps1 files and run them again and again.

Built on .NET
It works with objects, not plain text. That is its superpower.

Cross-platform
PowerShell 7 runs on Windows, Linux and macOS.



GETTING STARTED

Versions and Installation

Windows PowerShell

```
PS C:\Users\wahid> $PSVersionTable.PSVersion
Major  Minor  Patch
5      1      26100

PS C:\Users\wahid> winget install --id Microsoft.PowerShell --source
winget
Successfully installed

PS C:\Users\wahid> pwsh
PowerShell 7.4.6
PS C:\Users\wahid> $PSVersionTable.PSVersion.Major
7
```

Windows PowerShell 5.1

Built into Windows. Stable, but no longer gets new features.

PowerShell 7

The modern, open-source version. Installs side by side; start it with pwsh.

Windows Terminal

The best place to run it, with tabs for PowerShell, Command Prompt and Linux.

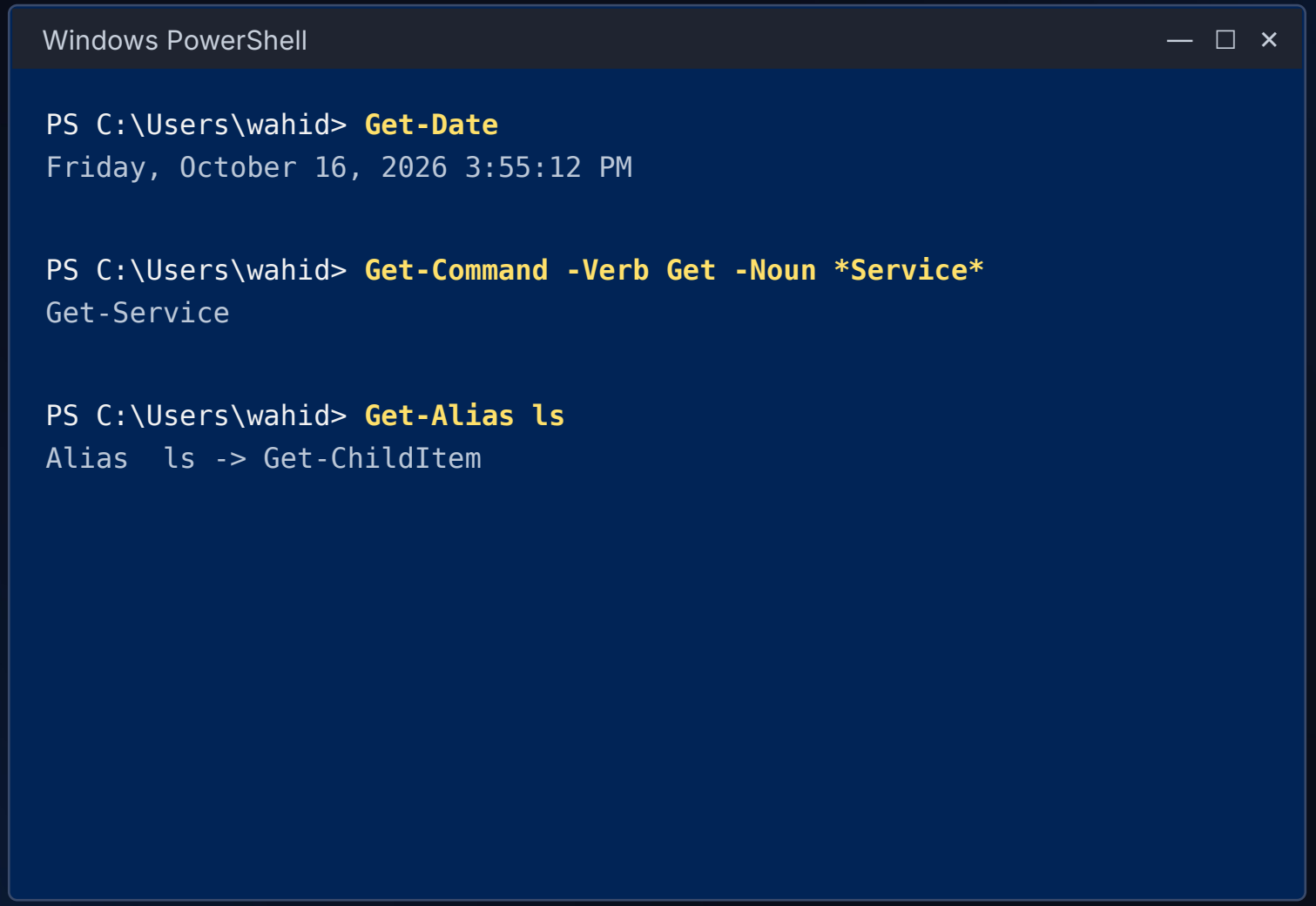
Windows PowerShell Scripting • Full Course



PART 2 · CORE SKILLS

CORE SKILLS

Cmdlets: Verb-Noun Commands



Verb-Noun

Get-, Set-, New-, Remove-, Start-, Stop-... The verb is the action, the noun is the thing.

Discover commands

Get-Command finds commands; Tab completes names and parameters.

Aliases

Short names like ls, dir and cd. Great for typing, but use full names in scripts.

CORE SKILLS

Getting Help

Windows PowerShell

```
PS C:\Windows\System32> Update-Help

PS C:\Users\wahid> Get-Help Get-Process -Examples
Example 1: Get a list of all active processes

PS C:\Users\wahid> Get-Help *firewall*
PS C:\Users\wahid> Get-Process | Get-Member
```

Get-Help

Read the manual for any command. -Examples shows real usage; -Online opens the web page.

Get-Member

Shows what properties and methods an object has. Your X-ray glasses.

CORE SKILLS

Objects and the Pipeline

Windows PowerShell

```
PS C:\Users\wahid> Get-Process | Sort-Object CPU -Descending | Select-Object -First 3 Name, CPU
Name      CPU
msedge    412.5
Teams     230.1

PS C:\Users\wahid> Get-Service | Where-Object Status -eq Running | Measure-Object
Count : 112

PS C:\Users\wahid> Get-Process | Export-Csv processes.csv -NoTypeInformation
```

Objects, not text

Each process is an object with properties like Name, CPU and Id.

The pipeline |

Passes objects from one command to the next, like an assembly line.

Sort, filter, select

Sort-Object, Where-Object, Select-Object, Measure-Object, Export-Csv.

CORE SKILLS

Working with Files and Folders

Windows PowerShell

```
PS C:\Users\wahid> New-Item -ItemType Directory -Path C:\Lab
PS C:\Users\wahid> Set-Content C:\Lab\hello.txt "Hello, PowerShell!"
PS C:\Users\wahid> Get-Content C:\Lab\hello.txt


PS C:\Users\wahid> Get-ChildItem C:\Lab -Recurse -Filter *.txt
PS C:\Users\wahid> Copy-Item C:\Lab\hello.txt C:\Lab\backup.txt

PS C:\Users\wahid> Remove-Item C:\Lab\*.tmp -WhatIf
What if: Performing "Remove File" on target "C:\Lab\old.tmp".
```

Item cmdlets

New-Item, Get-ChildItem, Copy-Item, Move-Item, Rename-Item, Remove-Item.

Content cmdlets

Get-Content reads; Set-Content writes; Add-Content appends.

-WhatIf and -Confirm

Preview a dangerous command before running it for real.

Windows PowerShell Scripting • Full Course



PART 3 · SCRIPTING

SCRIPTING

Variables, Arrays and Hash Tables

Windows PowerShell

```
PS C:\Users\wahid> $name = "Wahid"
PS C:\Users\wahid> $count = 5
PS C:\Users\wahid> "Hello $name, you have $count new messages"
Hello Wahid, you have 5 new messages

PS C:\Users\wahid> $fruits = @("apple", "banana", "cherry")
PS C:\Users\wahid> $fruits[0]
apple

PS C:\Users\wahid> $user = @{ Name = "Ali"; Age = 30 }
PS C:\Users\wahid> $user.Name
Ali
```

Variables

Start with $. PowerShell figures out the type: text, number, date, object.

Arrays @( )

An ordered list. Count starts at 0.

Hash tables @{ }

Key and value pairs, like a small dictionary.

SCRIPTING

Operators and Decisions

Windows PowerShell

```
$disk = Get-PSDrive C
$freeGB = [math]::Round($disk.Free / 1GB, 1)

if ($freeGB -lt 10) {
   Write-Warning "Low disk space: $freeGB GB left"
} elseif ($freeGB -lt 50) {
   Write-Host "Disk is OK: $freeGB GB free"
} else {
   Write-Host "Plenty of space" -ForegroundColor Green
}
```

Comparison operators

-eq equal, -ne not equal, -gt greater, -lt less, -ge, -le.

Text matching

-like with wildcards (*), -match with regular expressions.

Logic

-and, -or, -not combine conditions.

SCRIPTING

Loops: Repeat the Work

Windows PowerShell

```
foreach ($svc in "Spooler", "W32Time") {
   $s = Get-Service $svc
   "$($s.Name) is $($s.Status)"
}

PS C:\Users\wahid> 1..5 | ForEach-Object { $_ * 10 }
10 20 30 40 50

$tries = 0
while ($tries -lt 3) { $tries++; "Attempt $tries" }
```

foreach

Repeat for every item in a list.

ForEach-Object and $_

Inside the pipeline, $_ means the current item.

while / do-until

Repeat while a condition is true, e.g. retry a connection.

SCRIPTING

Your First Script and Execution Policy

Windows PowerShell

```
PS C:\Users\wahid> Get-ExecutionPolicy
Restricted

PS C:\Users\wahid> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser


PS C:\Users\wahid> notepad C:\Lab\DiskReport.ps1
PS C:\Users\wahid> C:\Lab\DiskReport.ps1
Plenty of space
```

Scripts are .ps1 files

Write them in VS Code or Notepad, then run them by path.

Execution policy

A safety setting. RemoteSigned runs your own scripts; downloaded ones must be signed.

Run by path

Type the full path, or .\script.ps1 in the current folder.

SCRIPTING

Functions and Parameters

Windows PowerShell

```
function Get-FreeSpace {
   [CmdletBinding()]
   param(
       [string]$Drive = "C",
       [int]$WarnGB = 10
   )
   $gb = [math]::Round((Get-PSDrive $Drive).Free / 1GB, 1)
   [pscustomobject]@{ Drive = $Drive; FreeGB = $gb; Low = $gb -lt $WarnGB
}
}

PS C:\Users\wahid> Get-FreeSpace -Drive C -WarnGB 50
Drive FreeGB  Low
C      212.4 False
```

Functions

Name a block of code and reuse it. Use Verb-Noun names, like real cmdlets.

param( )

Typed parameters with default values make your function flexible.

Return objects

[pscustomobject] output works with the pipeline: sort, filter, export.

Windows PowerShell Scripting • Full Course

PART 4 · ADVANCED

ADVANCED

Error Handling: try, catch, finally

Windows PowerShell

```
try {
   Get-Content C:\Lab\missing.txt -ErrorAction Stop
}
catch {
   Write-Warning "Could not read file: $($_.Exception.Message)"
}
finally {
   "Cleanup done"
}

WARNING: Could not read file: Cannot find path 'C:\Lab\missing.txt'
```

Two kinds of errors

Terminating errors stop the script; non-terminating errors only warn.

-ErrorAction Stop

Turns a warning-style error into one that try/catch can catch.

finally

Always runs, error or not. Perfect for cleanup.

ADVANCED

Modules and the PowerShell Gallery

Windows PowerShell

```
PS C:\Users\wahid> Get-Module -ListAvailable | Select-Object -First 3 Name

PS C:\Users\wahid> Find-PSResource -Name PSScriptAnalyzer
PS C:\Users\wahid> Install-PSResource -Name PSScriptAnalyzer -Scope CurrentUser

PS C:\Users\wahid> Invoke-ScriptAnalyzer C:\Lab\DiskReport.ps1
```

Modules

Packages of cmdlets and functions: ActiveDirectory, Az for Azure, and many more.

PowerShell Gallery

The public library of modules at powershellgallery.com.

Your own module

Save functions in a .psm1 file to share them across scripts.

ADVANCED

Managing Remote Computers

Administrator: PowerShell 7

```
PS C:\Windows\System32> Enable-PSRemoting

PS C:\Users\wahid> Invoke-Command -ComputerName SRV01, SRV02 -ScriptBlock
{ Get-Service Spooler }

PS C:\Users\wahid> Enter-PSSession -ComputerName SRV01
[SRV01]: PS C:\> hostname
SRV01
```

PowerShell remoting

Run commands on other computers over WinRM, or SSH in PowerShell 7.

One-to-many

Invoke-Command runs the same script on many servers at once.

Interactive

Enter-PSSession gives you a prompt on the remote machine.

ADVANCED

Jobs, Scheduling and Web APIs

Windows PowerShell

```
PS C:\Users\wahid> $job = Start-Job { Get-ChildItem C:\ -Recurse -
ErrorAction SilentlyContinue | Measure-Object }
PS C:\Users\wahid> Receive-Job $job -Wait


$t = New-ScheduledTaskTrigger -Daily -At 7am
$a = New-ScheduledTaskAction -Execute pwsh -Argument "-File C:\Lab\
DiskReport.ps1"
Register-ScheduledTask -TaskName DiskReport -Trigger $t -Action $a



PS C:\Users\wahid> Invoke-RestMethod
https://api.github.com/repos/PowerShell/PowerShell | Select-Object
stargazers_count
```

Background jobs

Start-Job runs slow work in the background; Receive-Job collects the result.

Scheduled tasks

Run your script automatically, every day, without you.

REST APIs

Invoke-RestMethod turns web data (JSON) straight into objects.

ADVANCED

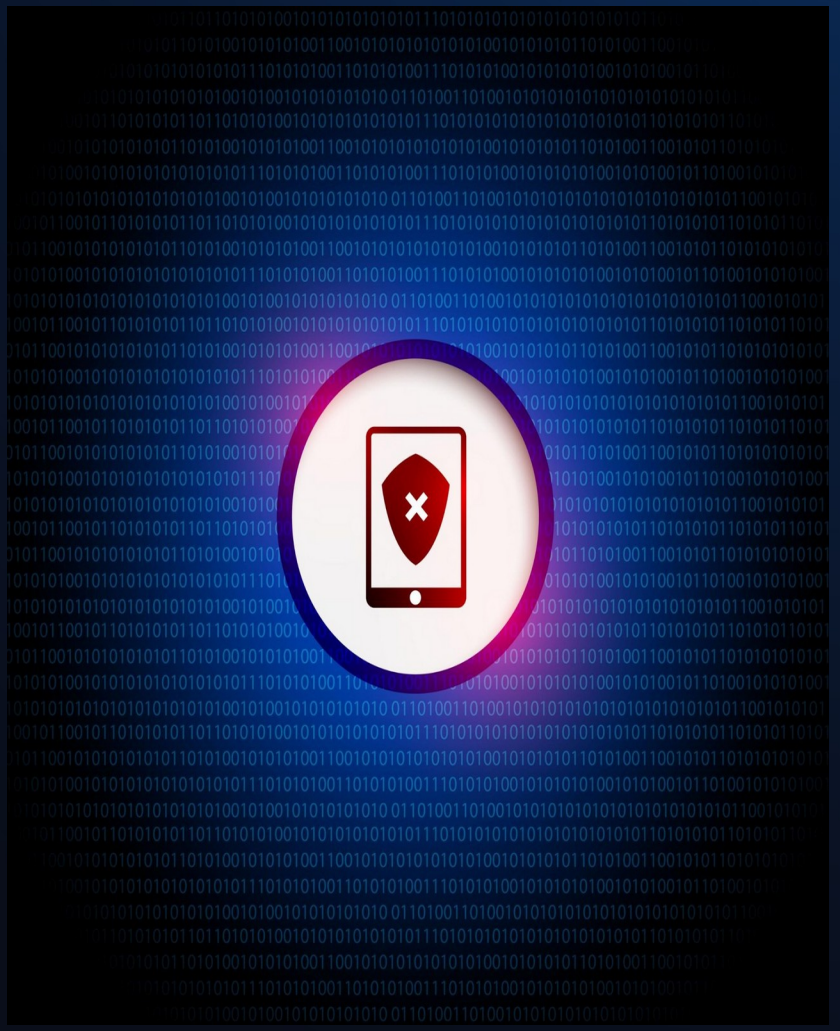
Scripting Best Practices

Readable code
Full cmdlet names, clear variable names, comments and comment-based help.

Safe by default
Test with -WhatIf, run as a standard user, and never store passwords in scripts.

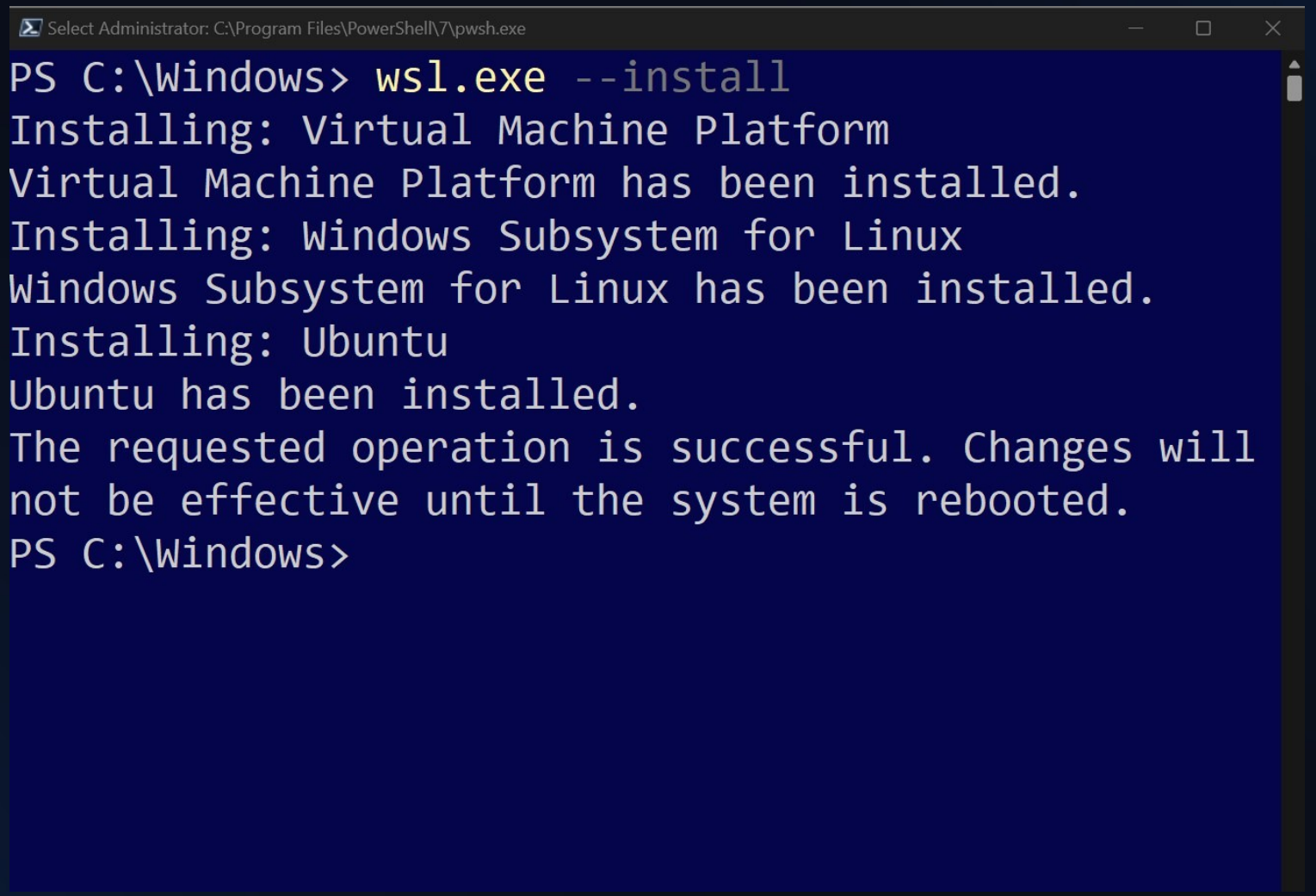
Version control
Keep scripts in Git so you can track and undo changes.

Trust, then run
Read scripts from the internet before running them. Check signatures.




Windows PowerShell Scripting • Full Course
PART 5 · LINUX ON WINDOWS WITH WSL

Install WSL with One Command



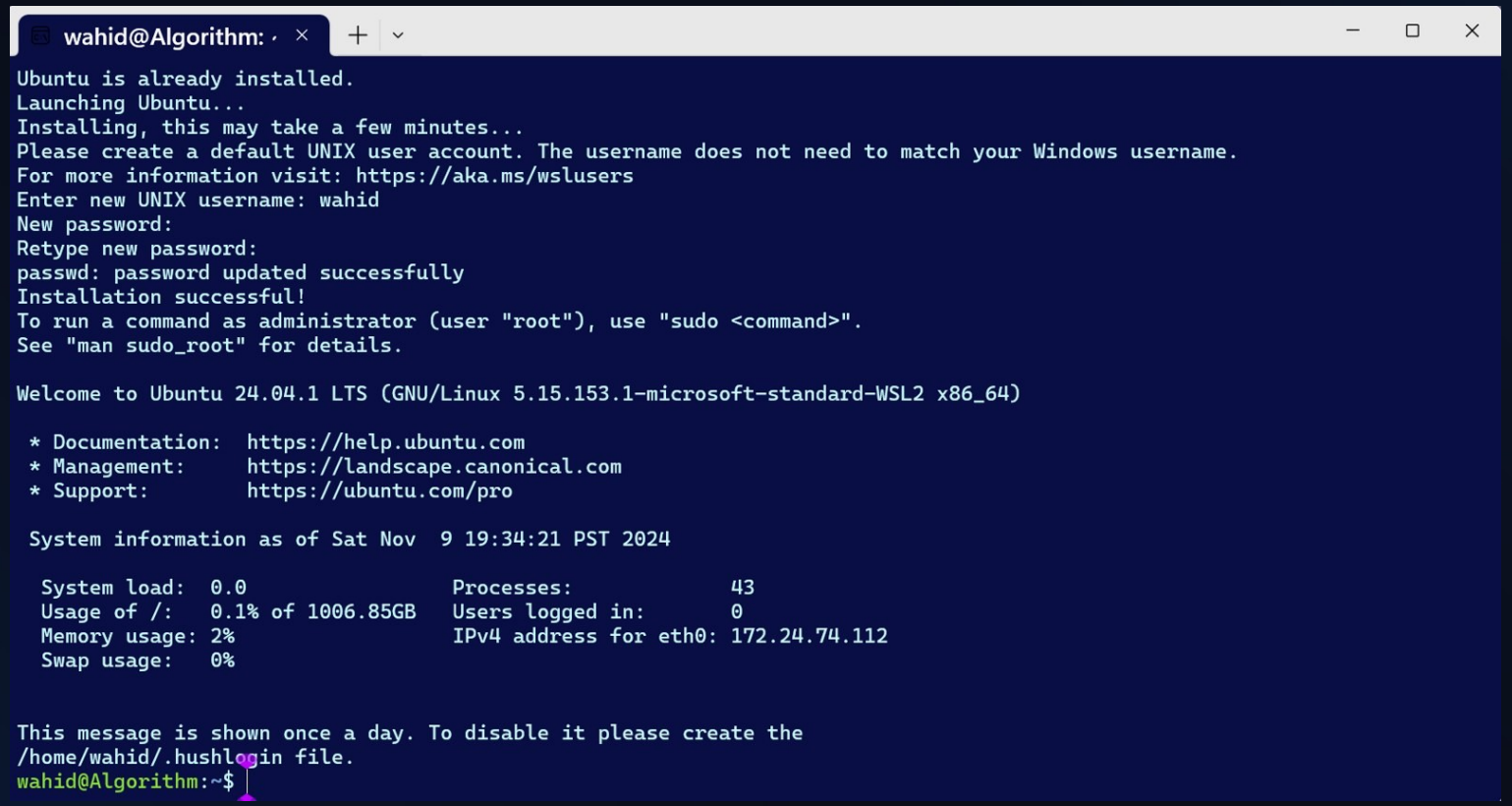
Run as Administrator

Open PowerShell as admin and run: wsl.exe --install

Reboot

It enables Virtual Machine Platform and WSL, installs Ubuntu, then asks you to restart.

First Start: Create Your Linux User



Your Linux account

Choose a Linux username and password. They do not need to match Windows.

Welcome to Ubuntu

You now have a full Ubuntu system, with sudo, apt and the Linux file system.

WSL

Everyday WSL Commands

Windows PowerShell

```
PS C:\Users\wahid> wsl -l -v
  NAME      STATE      VERSION
* Ubuntu    Running    2

wahid@Algorithm:~$ touch ~/.hushlogin      # hide the daily welcome
message
wahid@Algorithm:~$ hostname && id -un
Algorithm
wahid

wahid@Algorithm:~$ sudo su -
root@Algorithm:~# poweroff
PS C:\Users\wahid> wsl --shutdown
```

Manage from PowerShell

wsl -l -v lists distros; wsl --shutdown stops them all; wsl --update updates WSL.

Shutting down

In WSL, systemd poweroff may need root. From Windows, wsl --shutdown always works.

WRAP-UP

Your Next Steps

Automate one task

Pick one boring task you do every week, and script it.

Read and reuse

Use Get-Help -Examples, and learn from well-written modules.

Keep learning

Windows, Linux and programming courses, free at MyWebUniversity.com.

