



# The Experts Conference

SPONSORED BY  Quest



**September 29-Oct 1, 2025**  
Minneapolis, MN

Where Identity Security, Microsoft 365, and Copilot experts gather, learn, and innovate



**PAULA JANUSZKIEWICZ**  
Founder and CEO, CQURE



**ALEX SIMONS**  
CVP, Microsoft



**ADAM BANKS**  
Former CIO, Maersk

The logo for The Experts Conference (TEC) features the letters 'TEC' in a bold, sans-serif font. The 'T' is orange, the 'E' is grey, and the 'C' is blue. A vertical black line is positioned to the right of the 'C'.

**The Experts  
Conference**

SPONSORED BY  **Quest**

**Eric Weintraub**

IT Architect

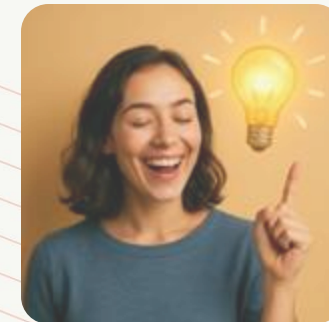
Manager of Identity Automation

Quest Software

# PowerShell Demystified: From Zero to Script Off Challenge Contender

#TEC2025

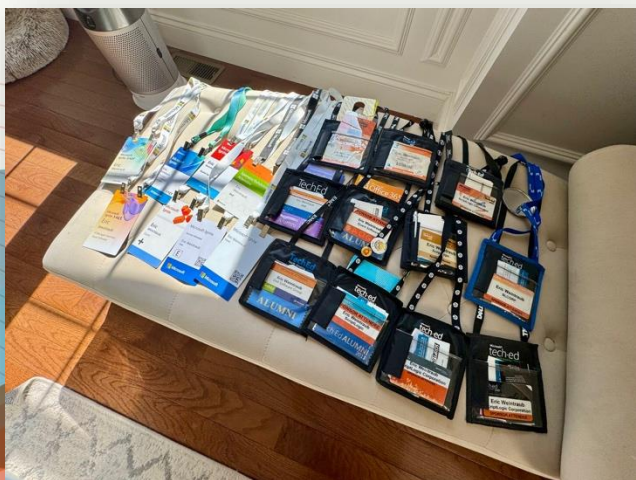
# Agenda



# TEC

## The Experts Conference

SPONSORED BY  Quest



## from the Trenches



IT Architect and 20-year veteran of Microsoft technologies. From NT4 to Azure infrastructure and everything in between. Focused on modernizing systems management for a cloud-first world. Originally from ScriptLogic.



4K  
Users



15K  
Devices



Global /  
Domain  
Admin



M365 /  
Identity /  
Automat  
ion /  
DevOps



I LOVE  
PWSH



# Don't be afraid!

## Fun

- It's like building with Lego, but it's business critical

## AI / Vibe Scripting

- Ai gives everyone the ability to create

## Cross-Platform & Modern

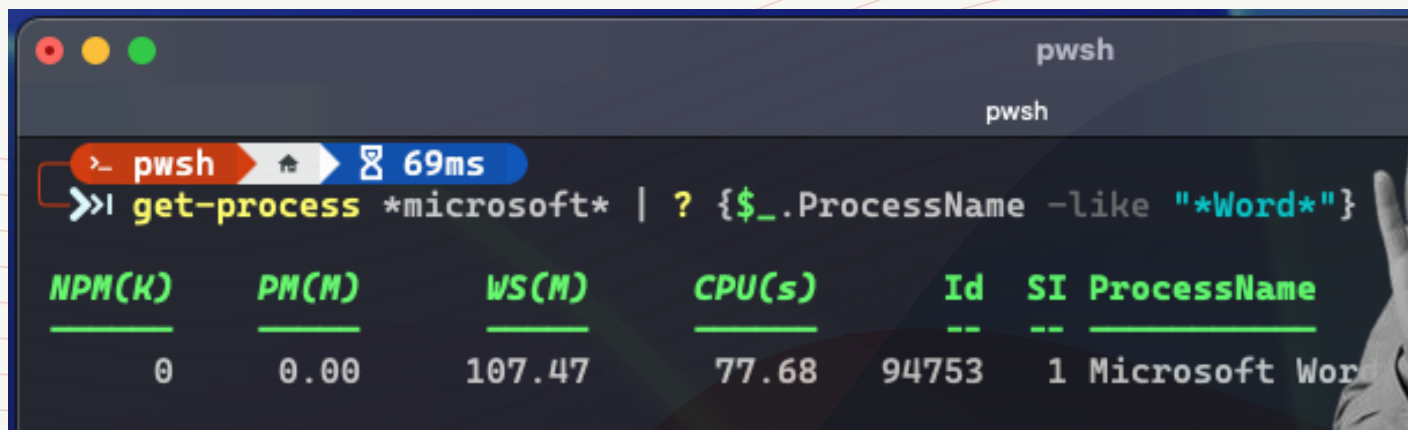
- PowerShell 7+ runs on Windows, macOS, and Linux

## Microsoft

- Microsoft doesn't like to make nice Admin tools for us, they give us PowerShell modules

## Career

- Be the one automating not the one Ai replaces with automation



A terminal window titled 'pwsh' showing a PowerShell command and its output. The command is `>>I get-process *microsoft* | ? {$_.ProcessName -like "*Word*"}`. The output is a table with columns: `NPM(K)`, `PM(M)`, `WS(M)`, `CPU(s)`, `Id`, `SI`, and `ProcessName`. The first row of data shows: `0`, `0.00`, `107.47`, `77.68`, `94753`, `1`, and `Microsoft Word`.

<code>NPM(K)</code>	<code>PM(M)</code>	<code>WS(M)</code>	<code>CPU(s)</code>	<code>Id</code>	<code>SI</code>	<code>ProcessName</code>
0	0.00	107.47	77.68	94753	1	Microsoft Word



# PowerShell Desktop vs Windows

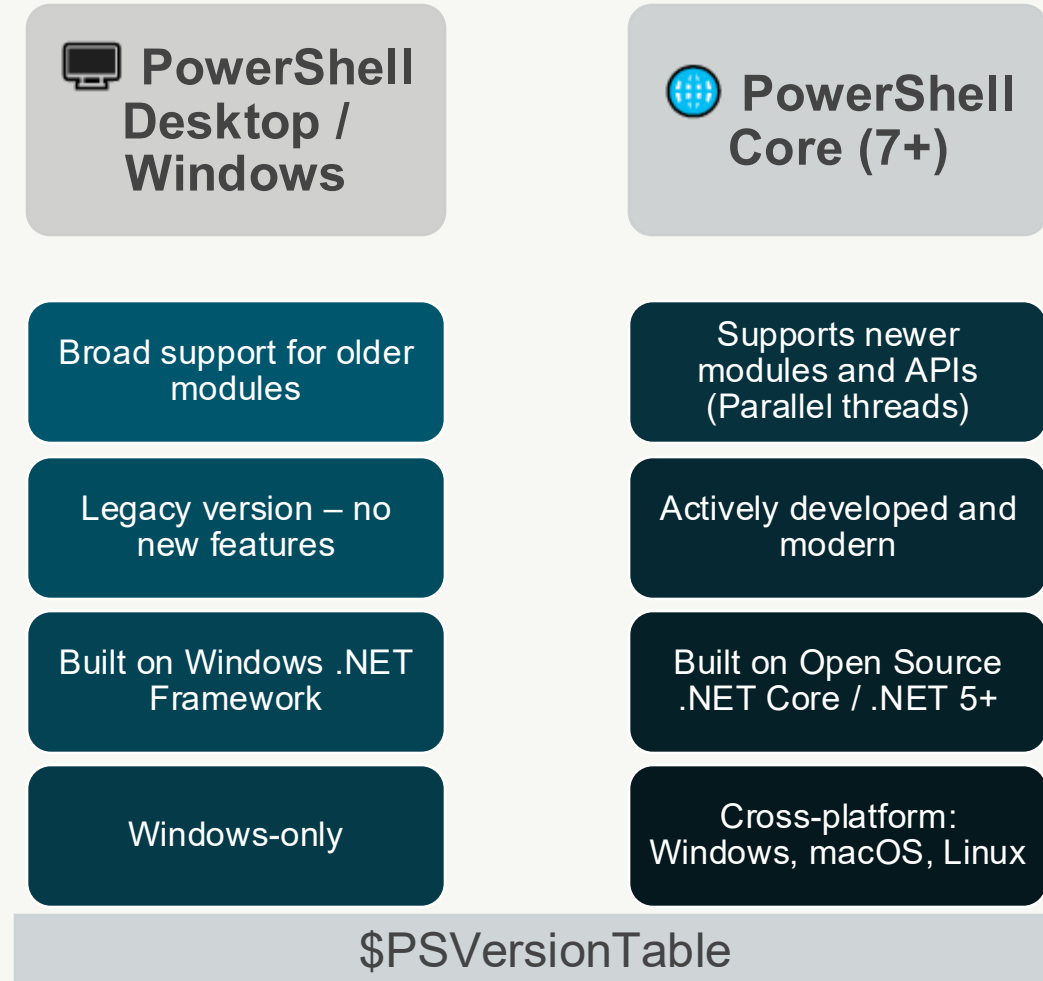
## Interview Questions:

- “Tell me any difference between Windows PowerShell and PowerShell Core”
- “Tell me what was the last version of Windows PowerShell or what the current version of Core is”

## Simple Questions but no one ever answers

```
> pwsh 74ms
>> $PSVersionTable
```

Name	Value
PSVersion	7.5.3
PSEdition	Core
GitCommitId	7.5.3
OS	Darwin 24.6.0 Darwin
Platform	Unix
PSCompatibleVersions	{1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion	2.3
SerializationVersion	1.1.0.1
WSManStackVersion	3.0



# Better IDE: Visual Studio Code

<https://code.visualstudio.com/download>

---

## Integrated Terminal

- Run PowerShell directly inside the editor

## PowerShell Extension

- Syntax help, IntelliSense, and error highlighting

## Copilot Support

- AI-assisted scripting with GitHub Copilot

## Cross-Platform

- Works on Windows, macOS, and Linux

## Version Control Built-In

- Git integration for tracking and collaboration

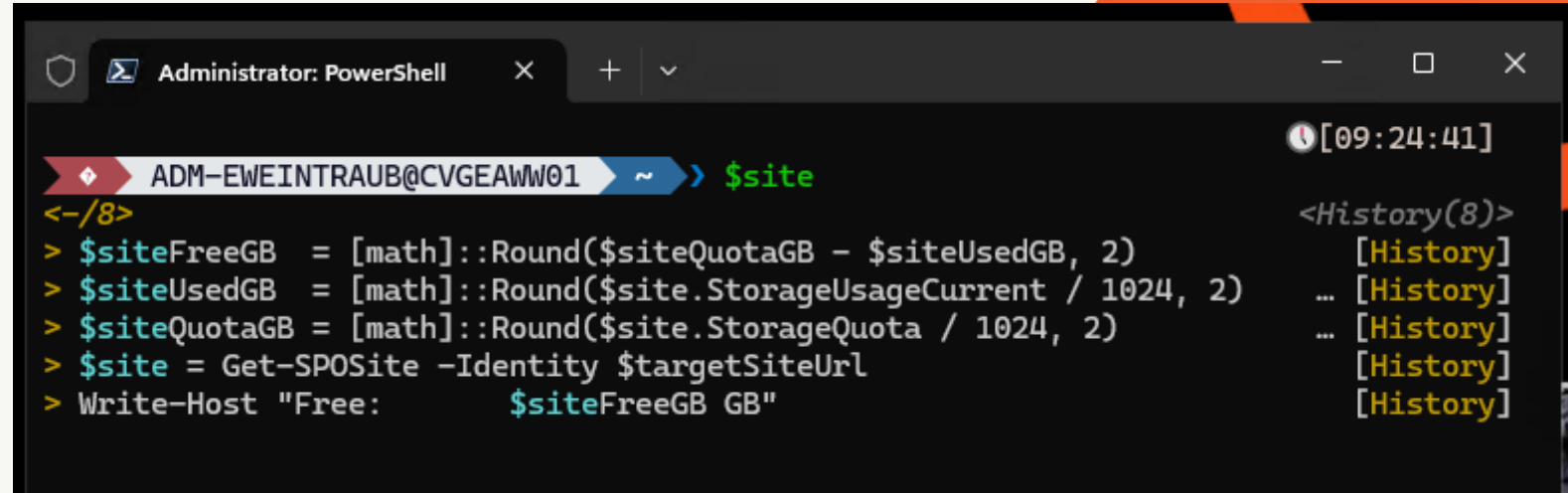
## SSH Remoting

- Create, Modify, and Execute code on a remote host using local IDE

# Better Terminals

## Terminals (for Windows)

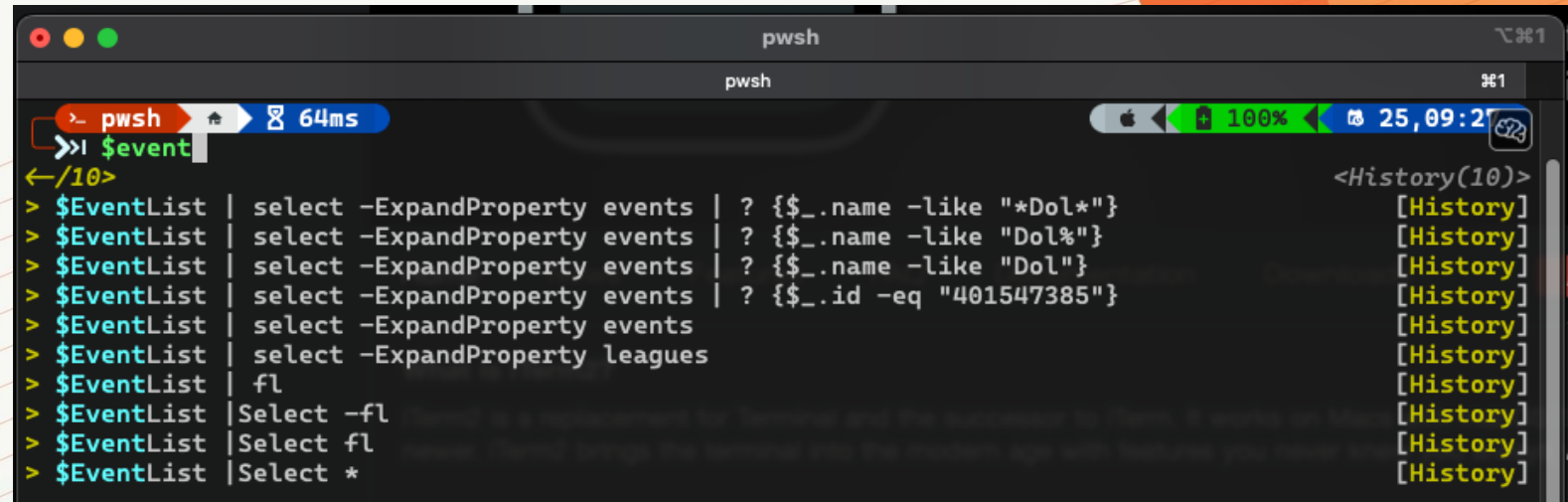
- Included in
  - Windows 11 22H2+
  - Server 2025+

A screenshot of a Windows PowerShell terminal window titled "Administrator: PowerShell". The window shows a command prompt with the user "ADM-EWEINTRAUB@CVGEAWW01" and a tilde symbol. The command "\$site" is entered, and the output shows a list of commands and their results, including calculations for free and used storage space. The history of commands is visible on the right side of the window.

```
Administrator: PowerShell
[09:24:41]
ADM-EWEINTRAUB@CVGEAWW01 ~ > $site
<--/8>
> $siteFreeGB = [math]::Round($siteQuotaGB - $siteUsedGB, 2)
> $siteUsedGB = [math]::Round($site.StorageUsageCurrent / 1024, 2)
> $siteQuotaGB = [math]::Round($site.StorageQuota / 1024, 2)
> $site = Get-SPOSite -Identity $targetSiteUrl
> Write-Host "Free:      $siteFreeGB GB"
<History(8)>
[History]
... [History]
... [History]
[History]
[History]
```

## iTerm2 (for Mac)

- <https://iterm2.com/>

A screenshot of an iTerm2 terminal window titled "pwsh". The window shows a command prompt with the user "pwsh" and a tilde symbol. The command "\$event" is entered, and the output shows a list of commands and their results, including a list of events and their properties. The history of commands is visible on the right side of the window.

```
pwsh
pwsh
pwsh 64ms
> $event
<--/10>
> $EventList | select -ExpandProperty events | ? {$_.name -like "*Dol*"}
> $EventList | select -ExpandProperty events | ? {$_.name -like "Dol%"}
> $EventList | select -ExpandProperty events | ? {$_.name -like "Dol"}
> $EventList | select -ExpandProperty events | ? {$_.id -eq "401547385"}
> $EventList | select -ExpandProperty events
> $EventList | select -ExpandProperty leagues
> $EventList | fl
> $EventList | Select -fl
> $EventList | Select fl
> $EventList | Select *
<History(10)>
[History]
[History]
[History]
[History]
[History]
[History]
[History]
[History]
[History]
[History]
```



# More helpful Command Line (ListView) + Profiles

## Set-PSReadLineOption - PredictionViewStyle ListView

👉 ADD TO YOUR PROFILE

Your Profile? See the files involved with:

**\$Profile | Select \***

Add it with this oneliner:

**"Set-PSReadLineOption - PredictionViewStyle ListView" | out-file \$profile.CurrentUserAllHosts -append**

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ADM-EWEINTRAUB> Set-PSReadLineOption -PredictionViewStyle ListView
Set-PSReadLineOption : A parameter cannot be found that matches parameter name 'PredictionViewStyle'.
At line:1 char:22
+ Set-PSReadLineOption -PredictionViewStyle ListView
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Set-PSReadLineOption], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.SetPSReadLineOption

PS C:\Users\ADM-EWEINTRAUB> get-module psreadline

ModuleType Version      Name                               ExportedCommands
-----
Script      2.0.0         PSReadLine                        {Get-PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PS...

PS C:\Users\ADM-EWEINTRAUB> install-module psreadline -AllowClobber -Force
```

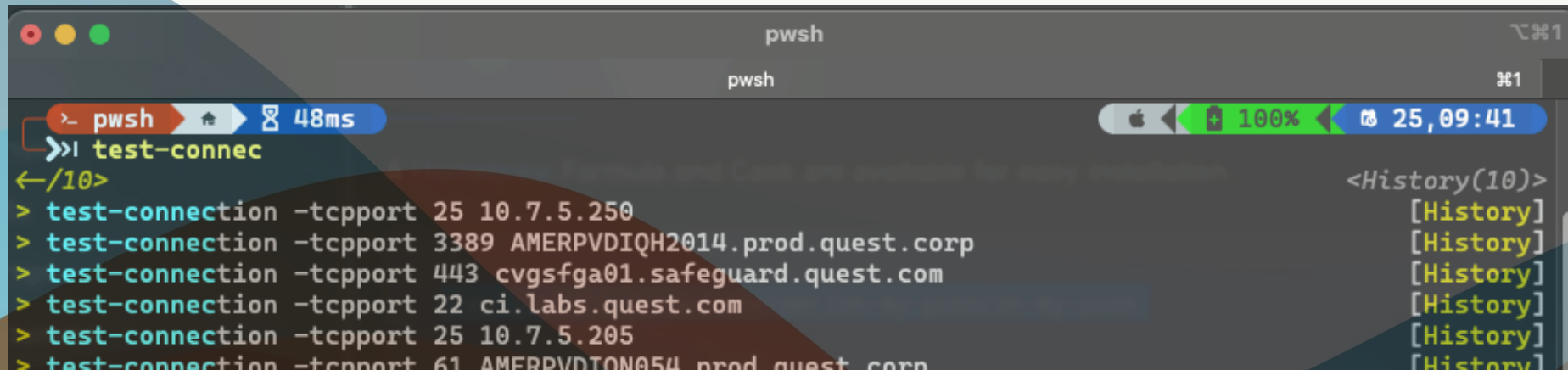
```
Administrator: Windows PowerShell
PS C:\Users\ADM-EWEINTRAUB> $profile
C:\Users\ADM-EWEINTRAUB\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\ADM-EWEINTRAUB> $profile | select *

AllUsersAllHosts      : C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
AllUsersCurrentHost   : C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1
CurrentUserAllHosts   : C:\Users\ADM-EWEINTRAUB\Documents\WindowsPowerShell\profile.ps1
CurrentUserCurrentHost : C:\Users\ADM-EWEINTRAUB\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
Length                : 84
```

# Oh-My-Posh?

<https://ohmyposh.dev/>

- Install Win Pre-2025: see <https://ohmyposh.dev/docs/installation/windows>
- Install Win2025: **winget install JanDeDobbeleer.OhMyPosh --source winget --scope user**
- Install Mac: **brew install jandedobbeleer/oh-my-posh/oh-my-posh**
- Install a font : **oh-my-posh font install** (*CascadiaCode (not-MS) /CaskaydiaCove Nerd Font*)
- Set your terminal to use font
- Exit and reopen terminal to ensure new path and font
- Add the following to your profile so it runs:
  - **oh-my-posh init pwsh | Invoke-Expression**



```
pwsh
pwsh
> pwsh 48ms
>> test-connect
<- /10>
> test-connection -tcpport 25 10.7.5.250
> test-connection -tcpport 3389 AMERPVDIQH2014.prod.quest.corp
> test-connection -tcpport 443 cvgsfga01.safeguard.quest.com
> test-connection -tcpport 22 ci.labs.quest.com
> test-connection -tcpport 25 10.7.5.205
> test-connection -tcpport 61 AMERPVDIQH2014.prod.quest.corp
<History(10)>
[History]
[History]
[History]
[History]
[History]
[History]
```

# 101: Language Basics

**Variables &  
Basic Syntax**

**Strings vs  
Numbers**

**Arrays & Data  
Collections**

**PowerShell  
Pipeline**

**Custom  
Objects & Data  
Structures**

**Filtering &  
Querying Data**



**Follow along**

## 102: Loops and Logic

---

If / Then  
Logic

Foreach  
Loops

Do / While  
Loops

**TEC**

**The Experts  
Conference**

SPONSORED BY  Quest



Follow along



## 103: Functions and Libraries

---

OOP

Functions

Libraries

**TEC**

**The Experts  
Conference**

SPONSORED BY  Quest



Follow along

# Real World Use Cases

---

## LAdmin

- LAPS system utilizing PKI, Azure Functions & Azure SQL

## User Affinity

- System for assigning ownership of servers

## Certificate Automation

- Automatic renewal of free SSL Certificates

## Identity Hygiene

- Checking for breached passwords and other account related issues

## Data Syncs to SQL

- AD, Entra MFA, Defender Tags, Entra Apps

## Dynamic DL's

- AllReports
- DirectReports

# Practical 365

## PowerShell Script-off

5:15pm-6:30pm (after welcome reception)

**Location:** Keynote ballroom

**You can still sign-up!**

Attendee with laptop with HDMI output?

Familiar with Microsoft 365 scripting?

AI Coding Assistants allowed 🤖

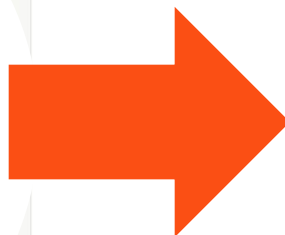
Chance to WIN BIG!



# TEC

The Experts  
Conference

SPONSORED BY 



## Q&A and Resources

---

<https://aka.quest/tec25/pwsh101>

