



Managing at Scale with PowerShell

Unlocking the Power of the Pipeline





Speakers



Jeff Hicks

PowerShell MVP | Author | Teacher

<https://jdhitsolutions.github.io>



Jordan Benzing

Microsoft MVP, Engineer

<https://jordantheitguy.com/>

Cabana session Wednesday 8:00-10:00AM

Can't see our slides? Can't hear? Need to repeat the question? Call us out!



Sequential vs Scale

- GUIs don't scale
- RDP management doesn't scale
- Legacy remoting doesn't scale
- Management via PowerShell has the *potential* to scale
- Do it for 1, 10, or 1000.
- PowerShell isn't always the answer



Where is Your Code?

- ◆ What command is running?
- ◆ Where is it running?
 - ◆ Locally?
 - ◆ Remotely?
 - ◆ Hybrid?
 - ◆ Application/Platform?
- ◆ Where is the output?
- ◆ What are you doing with the output?
 - ◆ Immediate use in the pipeline?
 - ◆ Where will it be consumed?

Filtering is Critical

- ◆ Early filtering
 - ◆ Look for filtering parameters
 - ◆ -Filter
 - ◆ -Name
 - ◆ -ID
 - ◆ Make this a habit
- ◆ Late filtering
 - ◆ Where-Object
 - ◆ Sometimes you have no other options
 - ◆ Or you are filtering the aggregate

Scripting at Scale Techniques

◆ Variables

- ◆ Don't re-run commands
- ◆ `$users = Get-ADUser -Filter *`
- ◆ Make your variable names meaningful
- ◆ `Help About_Variables`

◆ Arrays

- ◆ Saved cmdlet output is almost always an array
- ◆ `Help About_Arrays`

Scripting at Scale Techniques

- ◆ Generic collections
 - ◆ More efficient than arrays
 - ◆ Better search options
 - ◆ **.NET scripting**
- ◆ OutVariable
 - ◆ Store output for later use
 - ◆ CommonParameter
- ◆ PipelineVariable
 - ◆ CommonParameter
 - ◆ Store output for a later command in the **same** pipeline expression
 - ◆ Typically requires ForEach-Object
 - ◆ Help About_CommonParameters

Leveraging PowerShell Remoting

- ◆ PSSessions
 - ◆ Supports alternate credentials
 - ◆ Re-usable
- ◆ CimSessions
 - ◆ Firewall-friendly access to WMI/CIM
 - ◆ Supports alternate credentials
- ◆ Remoting Challenges
 - ◆ Credentials
 - ◆ 2nd Hop
 - ◆ Error Handling
- ◆ Configure remoting with SSH in PowerShell 7
 - ◆ SSH is the future

Parallel Processing

- ◆ Remoting with Invoke-Command
- ◆ Leveraging background jobs
- ◆ Leveraging thread jobs
 - ◆ Install-Module Threadjob
- ◆ ForEach-Object –parallel (PowerShell 7)
- ◆ Roll your own runspace
 - ◆ Out of scope today

DEMO



PowerShell at Scale

<https://github.com/jdhitsolutions/MMSMiami-2023>



Scripting at Scale

Best Practices

- ✓ Improve output properties
 - ✓ Include computer name
 - ✓ ComputerName
 - ✓ Watch property types
 - ✓ Consider metadata
- ✓ Leverage remoting
 - ✓ Include throttling
- ✓ Separate output from formatting
 - ✓ Don't limit yourself
 - ✓ Avoid having to rewrite code

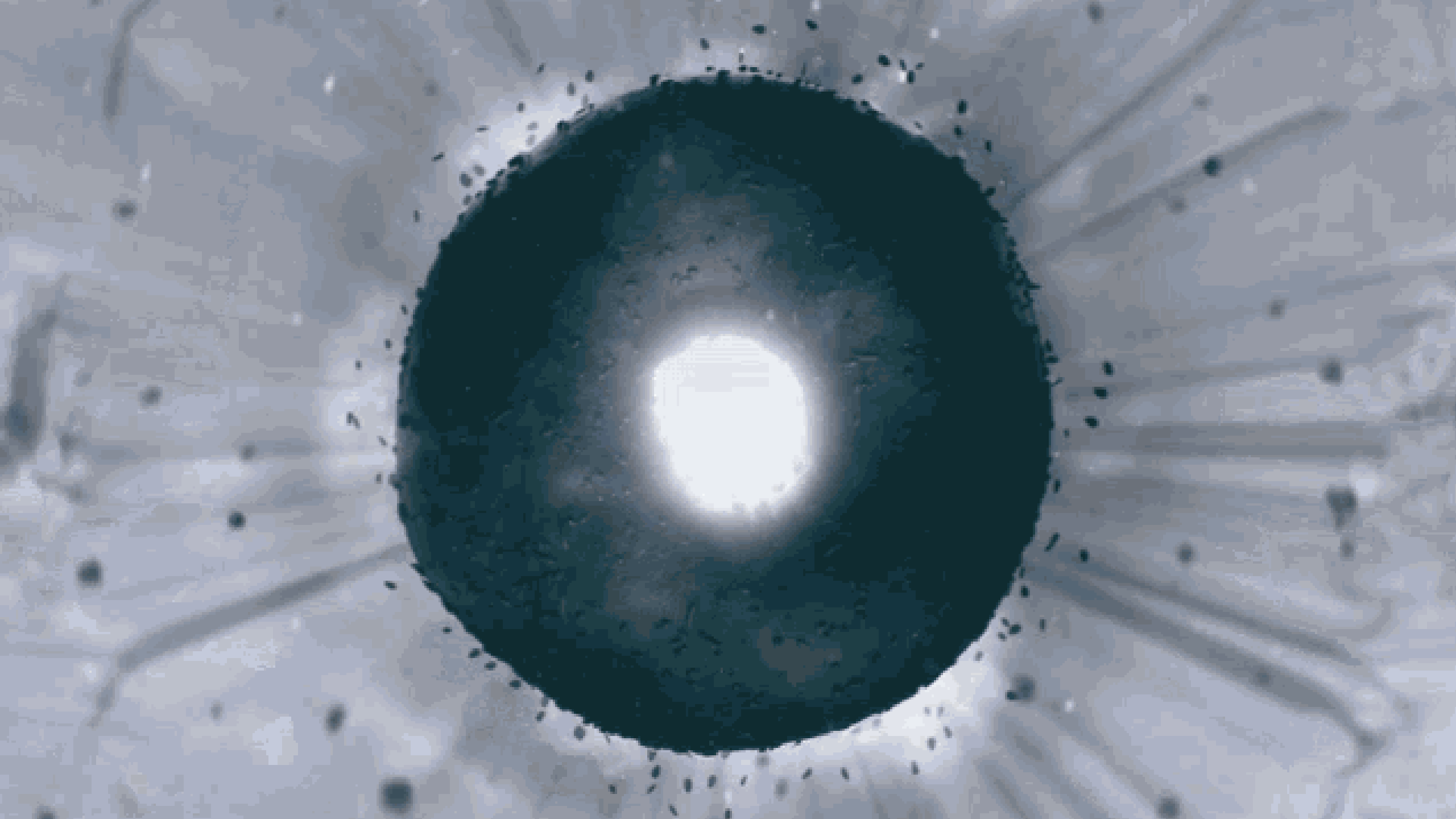


Scripting at Scale

Best Practices

- ✓ Log when necessary
 - ✓ Write-Host is not a logging mechanism
 - ✓ Use detailed verbose output
 - ✓ Write to the event log
 - ✓ Experiment with transcripts
- ✓ Don't think about doing one thing at a time
 - ✓ If in doubt, measure





Extended Q&A



Save the Dates



May 5-9, 2024



Oct 20-23, 2024