



# Advanced Series Scripted Actions

**Wifi SSID: MSFTGUEST**

**Password: msevent061724**

# Meet the Team



**Kevin Murray**

*VP, Technical Solutions*



# Meet the Team



**Nick Wagner**

*Sr. Systems Engineer*



# Meet the Team



**Carl Long**

*Director, Customer Engagement*





**WAIT A MINUTE**

**WHO ARE YOU?**

# Scripted Actions – Master Class

1

Setting the Foundation



4

Creating Scripts and RunBooks



2

Operationalizing your Environment



5

Testing and Troubleshooting



3

GitHub and Versioning



6

Updating, Reporting and Managing



# Community Matters



## Partner Forums

**Visit, Validate, Vote**

*\*Continue your conversations around scripts beyond just this room!*

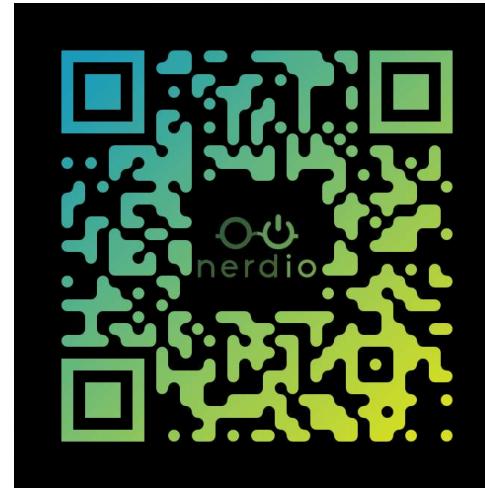


## Feature Release Webinars

**NMM V 5.2; July 10<sup>th</sup>, 2024**



# Resources for Today



- Scan the QR code or go to <http://nerdio.co/altsagithub>

## Repo for this sessions:

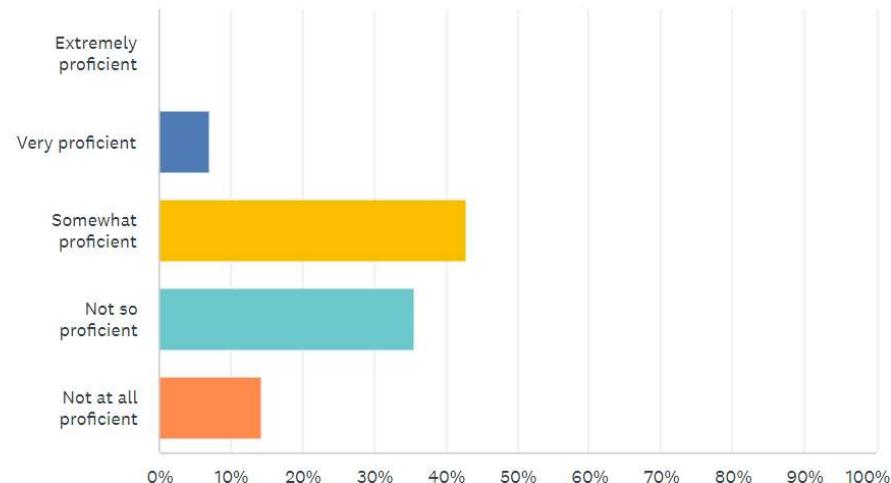
- Lab Guides
- Files(Stub, PS1, Scripts)
- Copy of this deck



# We all start somewhere...

Please rate your proficiency with Scripted Actions and Azure Runbooks

Answered: 14   Skipped: 0



#### ANSWER CHOICES

- ▼ Extremely proficient
- ▼ Very proficient
- ▼ Somewhat proficient
- ▼ Not so proficient
- ▼ Not at all proficient

#### RESPONSES

0.00%	0
7.14%	1
42.86%	6
35.71%	5
14.29%	2
<b>TOTAL</b>	<b>14</b>



## Pre-req check!

---

- ✓ Super Admin access to your NMM instance
- ✓ Login for GitHub
- ✓ Completed the Pre-event survey
  - ✓ <https://nerdio.co/alt-sa-pre>
- ✓ Have a Windows device with...
  - ✓ Visual Studio Code w/ PowerShell Extension

# Lab Assets

01

**Nerdio Lab**

lab.nerdiouniversity.biz

02

**Username**

lab(x)@nerdiouniversity.biz

03

**Password**

N3rdi04u





# Lesson 1

---

*Setting the foundation for Scripted Actions & Runbooks*

## Walkthrough

- ***Operationalizing***
- ***Taking an Inventory of Scripts***
- ***Understanding Repos***

# What Operationalizing your practice means to us

01

**MSP Roles are critical to driving your strategies and focused training will lead to accuracy.**

02

**Build a foundation of scalable assets and repos.**

03

**Script Families will create more precision and agility.**



# **What does an Operationalized MSP look like?**

## **Operationalizing**

*Building and executing processes which are measurable, accessible, repeatable, and iterative to help you grow your business.*

## **The Foundation to Unified Endpoint Management (UEM)**

- Scripts / Runbooks
- Images
- Applications
- Policies

# Start small and grow large

*Get started with the Nerdio Repository*



*Build your own repo and control everything*



*Build out further automation within your repo*

# Let Gravity Do the Work



## Solutions Looking for Problems

- Stop using scripts for Band-Aids and miracles.
- *Build the foundation and framework and manage it with intent.*



## Problems Looking for Solutions

- Start using scripts as assets and products.
- *There is no microwave for building meaningful products and services.*

# Pillars of a Scripted Action Strategy

## Internal

- Optimizations
- Efficiencies
- Accountability
- Scale
- Reliability

## External

- End User Experience
- End User Performance
- Compliance
- Security
- Availability

# Products & Assets

## *Intellectual Property – “Differentiator”*

### Inventory



Assets and Scale

### Mobility



Absence of barriers and constraints

### Flexibility



Combinations and Permutations

### Sustainability



Version control and future dev

# Taking Inventory with Script Families



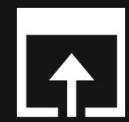
**Security**



**Patching / Updates**



**User Experience**



**App Delivery**



**Log / Data Capture**



**Azure Management**



**M365 Management**



**Third-Party Agents**

# Operationalizing Scripted Actions

## *Getting Started*

### Development Environment

- What do you have?
- What do you need?

### Repository

- Where are you storing automations?

### Linking to NMM

- Do you have the Azure resources in place?



# GitHub Software Version Control

- **KISS**
  - avoid the complexity and configuration paralysis
- **Repositories**
  - Single repository – multiple branches
  - At least two branches – three if you want to overachieve
    - *Test*
    - *Pilot (optional)*
    - *Main*
- **Set up GitHub at the beginning**
  - Starting early will pay dividends later
- **Readme.md**
  - don't leave home without it



# Lesson 1

---

*Setting the foundation for Scripted Actions & Runbooks*

## Lab

- ***Operationalizing Scripted Actions & Runbooks in your Environment***
- ***Getting started with GitHub***

# Centralizing Repositories

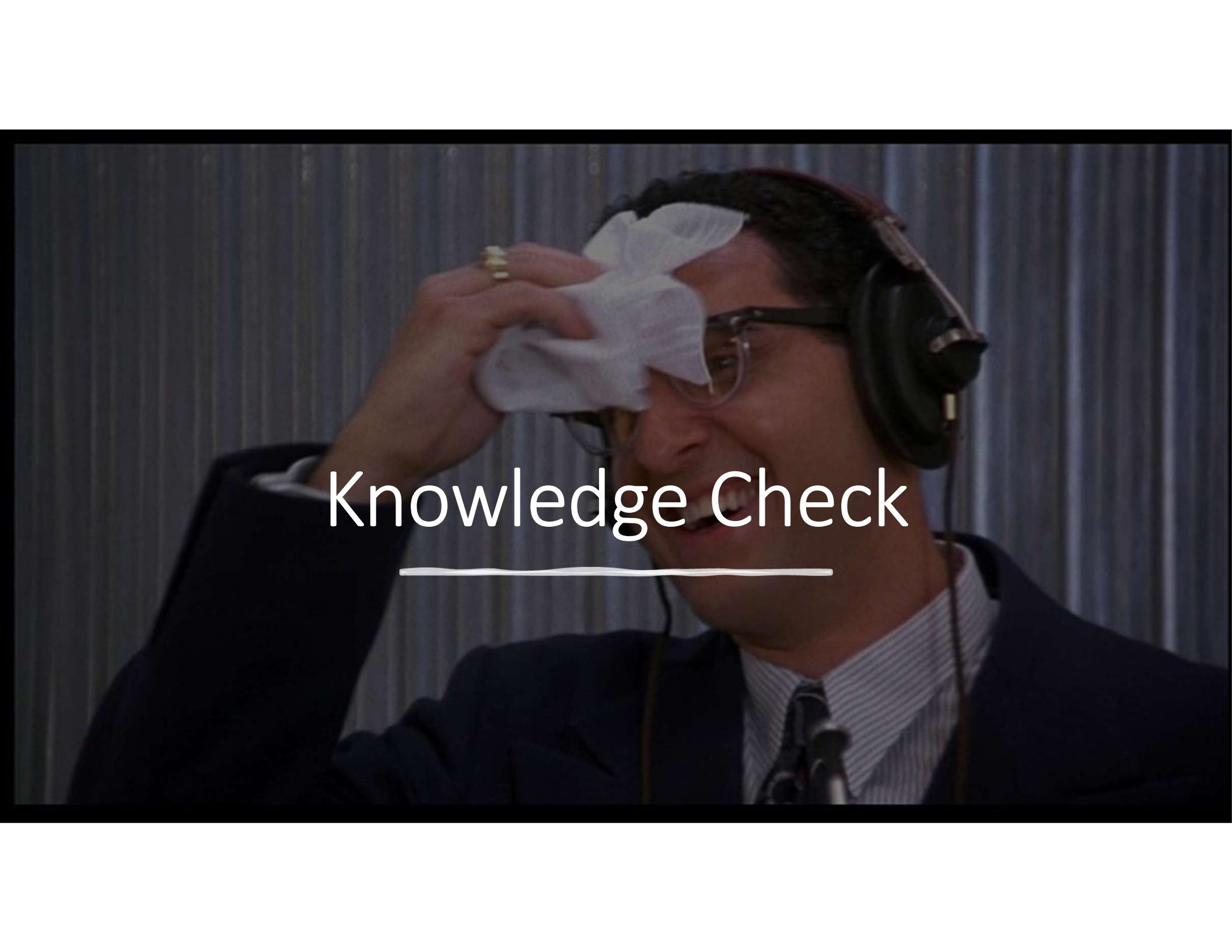
## What we've done so far

- ✓ IDE – VS Code
- ✓ GitHub

## What's up next?

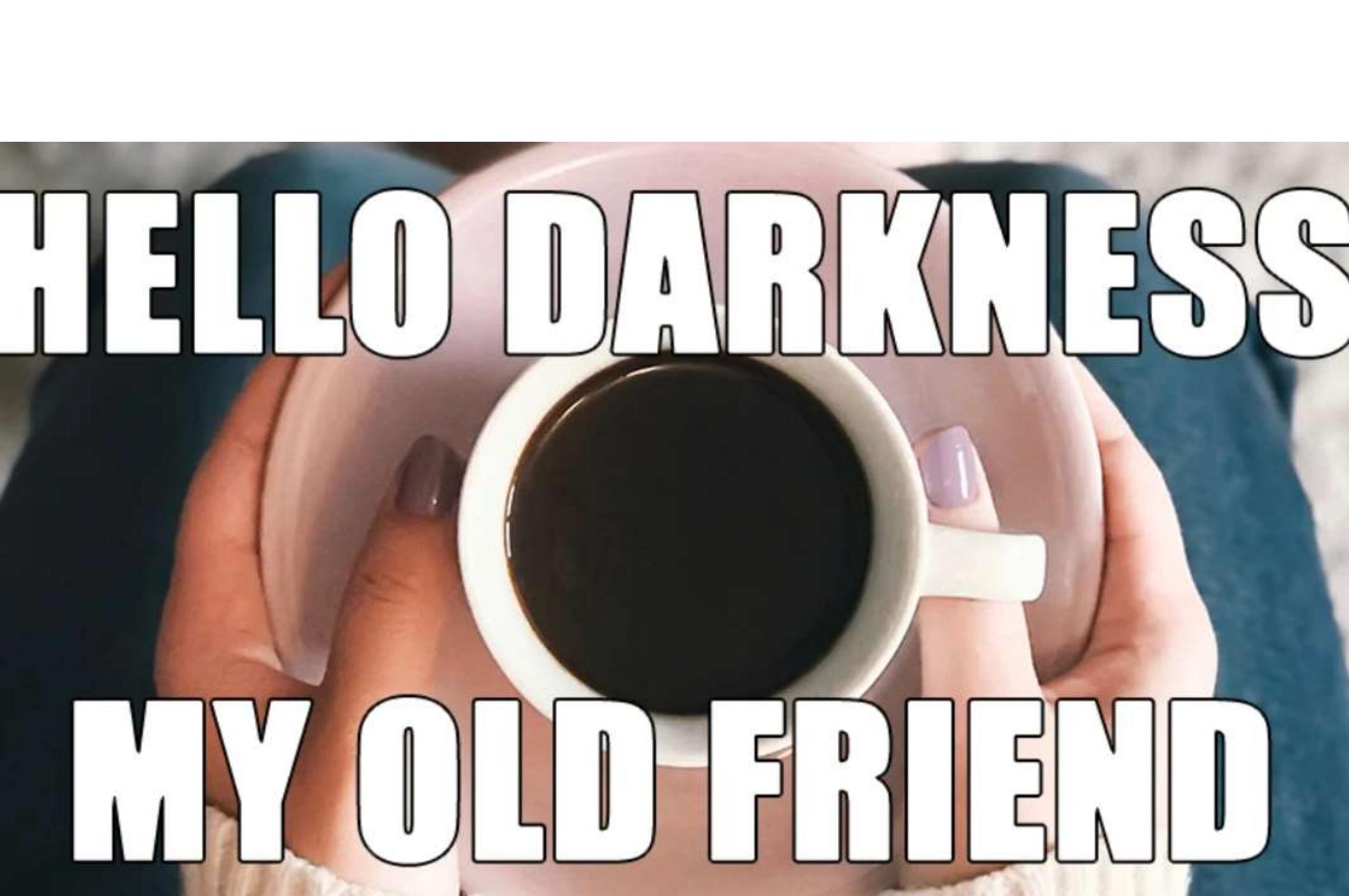
- Integrating GitHub with NMM
  - Nerdio Provided
  - Custom
- Permissions and Tokens
- Confirm the Link



A man in a dark suit, white shirt, and striped tie is shown from the chest up. He wears round-rimmed glasses and a black communication headset. He is sweating profusely, with a white cloth held against his forehead. His expression is one of intense stress or exertion.

# Knowledge Check

---



**HELLO DARKNESS**

**MY OLD FRIEND**



# Lesson 2

---

## Lab

- ***NMM Setup and Config***
- ***Synchronize and Visualize***
- ***Putting it all together***

# NMM Assignment and Flexibility

**Making the pieces work for your operations with three distinct and purpose driven accounts.**

1. Customer – production only
2. Pilot Customer – account with tolerance
3. Production IUL – resources for operations
4. Sandbox – if you blow it up...let it burn

Microsoft  
Partner  
Benefits  
**PSCB**

- <https://partner.microsoft.com/en-rs/partnership/compare-programs>
- <https://learn.microsoft.com/en-us/partner-center/partner-success-core-benefits>

# Nerdio Manager for MSP Production IUL

*Benefits and Ideas to justify the investment*

## **Microsoft Action Pack**

- Minimize the cost

## **AVD – maximize control**

- Moving staff to desktops has its benefits – and can be a hard sale
- Offering up RemoteApp for critical business items doesn't require any selling

*We recommend standing up a dev environment  
in a single-host pool, multi-user or personal.*

# NMM SA Assignment

Value adds – the differentiators

- All assignment – the universal actions that are for you

Line items – products you can sell

- By account – the actions that a customer purchases

*Scripted Actions must become part of your processes.*

*Monitoring those processes is part of your business.*

# Invoking Python Scripts in NMM

## Use Case Example

- Python developed (by a highschooler) specific to a vertical you are serving.
- There are 5 scripts of which 1 is a persistent script to run the other 4
- Each of the 4 scripts has a very specific task and is verbose to Azure Storage Container Table
- Log data is stored in tabular format with key-value pairs making it easy to report on

```
> invokepython.ps1 > ...
1  # path that is established in GI or Image
2  $pythonPath = "C:\MSPName\defaults\python_script.py"
3  # create a loop for multiple files and use JSON to list them - scale
4  |
5  # use the environmental variables on the image to determine this path
6  $pythonExecutable = "C:\Program Files\Python311\python.exe"
7
8  # Build the command to run the Python script
9  $command = "$pythonExecutable $pythonPath"
10
11 # Invoke the Python script
12 Invoke-Expression -Command $command
13
```



# **Scripted Actions – Azure Runbooks**

## **Differentiation and operationalizing your MSP**

**Functional and controlled means to develop scripts**

- VS Code

**Reliable means to organize and store script assets**

- GitHub

**Effective mechanics to operationalize script products**

- NMM

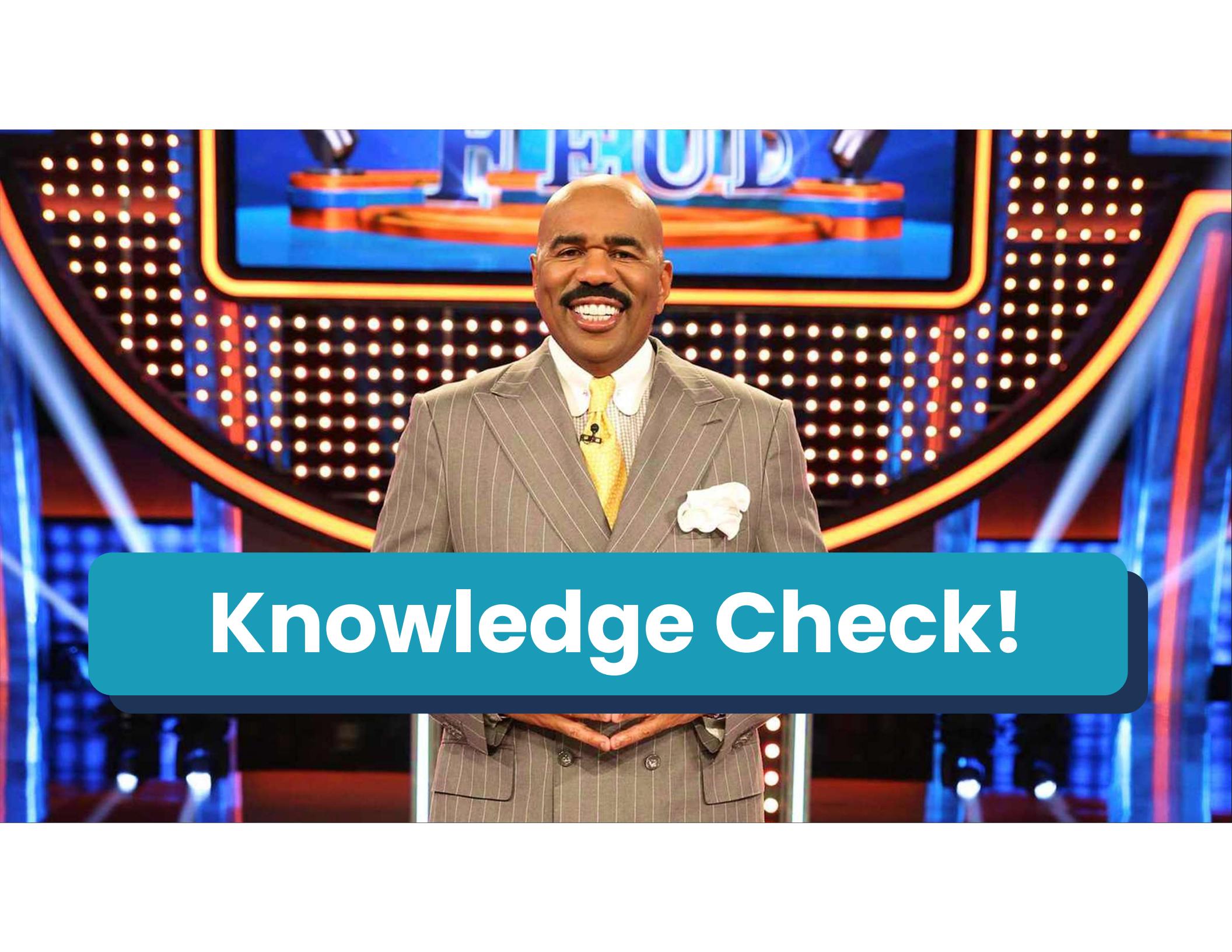
**MSP**



**S.M.A.R.T**



**Customer**

A photograph of Steve Harvey, host of the game show "Family Feud", standing on stage. He is wearing a grey pinstripe suit, a white shirt, a yellow patterned tie, and a white pocket square. He has a wide, joyful smile and is looking towards the camera. The background features the "FAMILY FEUD" logo in large blue letters, surrounded by a vibrant set with red, blue, and yellow lights.

# Knowledge Check!



# Lesson 3

---

## Overview

- ***The Power of Scripted Actions***

# Overview

## *What we'll cover:*



- What are Scripted Actions?
- What can Scripted Actions do for you?
- Examples of useful scripted actions
- Labs:
  - Windows Script
  - Runbook Script

# What are Scripted Actions?



What are

# Scripted Actions?

- PowerShell scripts

- Integrated into Nerdio environment

- Can be run as **Windows scripts** or

**Azure runbooks.**

AZURE RUNBOOKS ⓘ			
SEARCH		FILTER BY EXECUTION MODE	FILTER BY T.
ID ⓘ	NAME ⓘ	DESCRIPTION ⓘ	SOURCE AND DATE ⓘ
272	<b>Move VM to Availability Zone 2</b> Execution mode: Individual Execution environment: Azure runbook	Move a VM to Availability Zone 2	NMW Github: Get-Nerdio Modified: Feb 6, 2023 02:04 PM
90	<b>Use Spot VMs</b> Execution mode: Individual Execution environment: Azure runbook	(PREVIEW) Convert session hosts to spot VM instances for reduced costs in testing environments	NMW Github: Get-Nerdio Modified: Jan 23, 2023 12:39 PM
292	<b>Update AVD Agent</b> Execution mode: Individual Execution environment: Azure runbook	Updates the RDAgent on target VM	NMW Github: Get-Nerdio Modified: Dec 7, 2022 09:42 AM
106	<b>Shrink FSLogix Profiles</b> Execution mode: Individual Execution environment: Azure runbook	(PREVIEW) Creates a temp vm which is used to shrink FSLogix profiles	NMW Github: Get-Nerdio Modified: Dec 5, 2022 03:50 PM
305	<b>Create Hybrid Worker VM</b> Execution mode: Individual Execution environment: Azure runbook	Create a new vm in the specified vnet and enable it as a hybrid worker for the Nerdio automation account	NMW Github: Get-Nerdio Modified: Nov 8, 2022 11:41 AM
314	<b>TS-StartAzureVm-TEST</b> Execution mode: Combined Execution environment: Azure runbook	start specific AZ VM	Local Modified: Oct 20, 2022 07:05 AM
107	<b>Backup NMW App</b> Execution mode: Individual Execution environment: Azure runbook	(PREVIEW) Backup Nerdio App components to a storage account. Read Notes within before use.	NMW Github: Get-Nerdio Modified: Oct 13, 2022 02:32 PM
312	<b>Swap Image for all Host Pools</b> Execution mode: Individual Execution environment: Azure runbook	<no description>	Local Modified: Sep 30, 2022 03:35 PM
97	<b>Include data disk on Set as Image</b> Execution mode: Individual	For use during Set as Image operation only	Local Modified: Sep 22, 2022 04:07 PM
81	<b>Image</b> Execution mode: Individual	For use during Set as Image operation only	Local Modified: Sep 22, 2022 04:07 PM
315	<b>loop zone host</b> Execution mode: Individual	<no description>	Local Modified: Sep 30, 2022 03:32 PM





## Scripted Actions are **PowerShell**

---

### Welcome to the PowerShell Gallery

The central repository for sharing and acquiring PowerShell code including PowerShell modules, scripts, and DSC resources.

**Search PowerShell packages:**

Az, etc...



11,251  
Unique Packages

7,422,783,218  
Total package downloads

190,954  
Total packages

- Microsoft technology.
- Widely adopted.
- Modules available for many existing technologies.

\$HostPoolId  
\$HostPoolName  
\$AzureSubscriptionId  
\$AzureSubscriptionName  
\$AzureResourceGroupName  
\$AzureRegionName  
\$AzureVMName  
\$ADUsername  
\$ADPassword  
\$DesktopUser  
\$SATrigger  
\$SATriggerMode

Scripted Actions are

## Integrated with Nerdio

- Context-aware variables.



## RUN AZURE RUNBOOK SCRIPTED ACTION

Are you sure you want to run this Scripted action Shrink FSLogix Profiles?

Azure subscription

Mfs Sponsored Subscription (Nerdio - getnerdio.com)

### PARAMETERS ⓘ

NAME	VALUE
VNetName	avd-vnet-1
SubnetName	avd-subnet-1
FileSharePath	\storageaccount.file.core.windows.net\premiumfslogix01
TempVmSize	Standard_D16s_v4
TempVmResourceGroup	Type value or select existing secure variable...
AdditionalShrinkDiskPara	Type value or select existing secure variable...

parameters to send to the FSLogix-ShrinkDisk.ps1 script. E.g:-  
DeleteOlderThanDays 90 -IgnoreLessThanGB 5

Add

Scripted Actions are

# Integrated with Nerdio

- Context-aware variables.
- Secure Variables and Runtime parameters.



Scripted Actions are

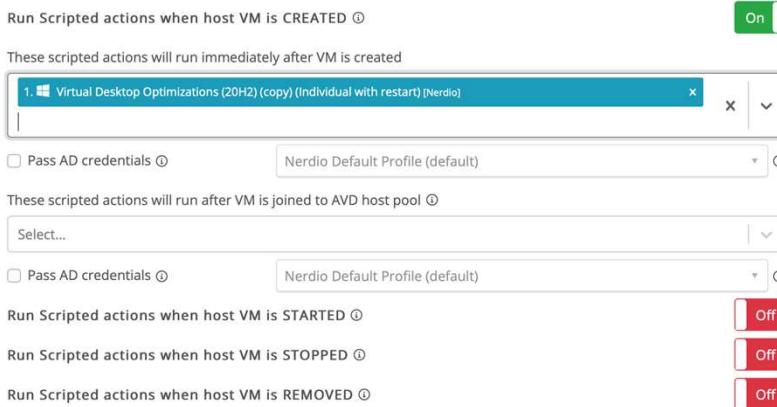
# Integrated with Nerdio

```
41
42 <# Variables:
43 {
44   "VNetName": {
45     "Description": "VNet in which to create the temp VM. Must be able to access the fslogix fileshare.",
46     "IsRequired": false
47   },
48   "SubnetName": {
49     "Description": "Subnet in which to create the temp VM.",
50     "IsRequired": false
51   },
52   "FileSharePath": {
53     "Description": "UNC path e.g. \\\\storageaccount.file.core.windows.net\\\\premiumfslogix01",
54     "IsRequired": false
55   },
56   "TempVmSize": {
57     "Description": "Size of the temporary VM from which the shrink script will be run.",
58     "IsRequired": false,
59     "DefaultValue": "Standard_D16s_v4"
60   },
61   "TempVmResourceGroup": {
62     "Description": "Resource group in which to create the temp vm. If not supplied, resource group of vnet will be used.",
63     "IsRequired": false
64   },
65   "AdditionalShrinkDiskParameters": {
66     "Description": "parameters to send to the FSLogix-ShrinkDisk.ps1 script. E.g: -DeleteOlderThanDays 90 -IgnoreLessThanGB 5",
67     "IsRequired": false
68   }
69 }
70 #>
```



Scripted Actions are

# Integrated with Nerdio



- Context-aware variables.
- Secure Variables and Runtime parameters.
- Can be associated to VM life cycle events.

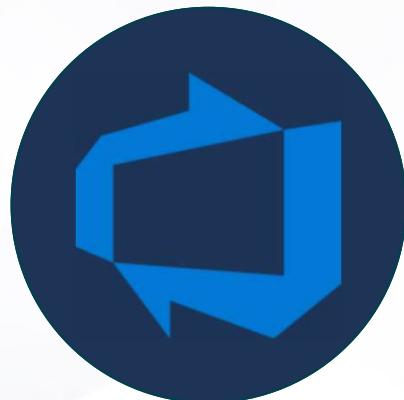


Scripted Actions are

## Integrated with Nerdio

- Context-aware variables.
- Secure Variables and Runtime parameters.
- Can be associated to VM life cycle events.
- Managed Permissions/Authentication.





Scripted Actions are

## Integrated with Nerdio

- Context-aware variables.
- Secure Variables and Runtime parameters.
- Can be associated to VM life cycle events.
- Managed Permissions/Authentication.
- Version Controlled and IaC compatible.

Scripted Actions can be

## Integrated with Nerdio

- Run as **SYSTEM** user.



Scripted Actions can be

## Integrated with Nerdio

- Run as **SYSTEM** user.
- Install software.



Scripted Actions can be

## Integrated with Nerdio

- Run as **SYSTEM** user.
- Install software.
- Modify registry.



Scripted Actions can be

## Integrated with Nerdio

- Run as **SYSTEM** user.
- Install software.
- Modify registry.
- Used on Image VM, AVD Host, or CloudPC





Scripted Actions are

## Azure Runbooks

- Run as Nerdio service principal.



Scripted Actions are

## Azure Runbooks

- Run as Nerdio service principal.
- Can make changes in Azure, extending the functionality of Nerdio



Scripted Actions are

## Azure Runbooks

- Run as Nerdio service principal.
- Can make changes in Azure, extending the functionality of Nerdio
- Examples:
  - Shrink OS Disk
  - Use Spot VMs
  - Reconfigure Nerdio using RestAPI

# The Power of Scripted Actions





**Scripted Actions can be used to add features to  
Nerdio without extended development time.**

The Power of Scripted Actions



```
1 $WinstationsKey = 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations'
2 New-ItemProperty -Path $WinstationsKey `
3                     -Name 'ICEControl' `
4                     -ErrorAction:SilentlyContinue `
5                     -PropertyType:dword `
6                     -Value 2 `
7                     -Force
8
9
```

## Windows Script Example

Enable RDP Public Shortpath



**Scripted Actions can be used to extend the  
functionality of Nerdio.**

The Power of Scripted Actions



```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

# Azure Runbook Example

Shrink Windows OS Disk



## 1. Run **Shrink Partition** script on VM.

```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

# Azure Runbook Example

Shrink Windows OS Disk



```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

# Azure Runbook Example

1. Run **Shrink Partition** script on VM.
2. Copy snapshot to new VHD in blob storage.

Shrink Windows OS Disk



```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

# Azure Runbook Example

1. Run **Shrink Partition** script on VM.
2. Copy snapshot to new VHD in blob storage.
3. Create new disk and assign to VM.

Shrink Windows OS Disk



```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

1. Run **Shrink Partition** script on VM.
2. Copy snapshot to new VHD in blob storage.
3. Create new disk and assign to VM.
4. Chop down VHD to size.

# Azure Runbook Example

Shrink Windows OS Disk



```
39 $PartitionScriptBlock | Out-File .\partitionscriptblock.ps1
40
41 Start-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName
42 $Result = Invoke-AzVMRunCommand ` 
43     -ResourceGroupName $AzureResourceGroupName ` 
44     -VMName $AzureVMName -ScriptPath .\partitionscriptblock.ps1 ` 
45     -CommandId runpowershellscript
46 Stop-AzVM -ResourceGroupName $AzureResourceGroupName -Name $AzureVMName -Force
47
48 if ($Result.Value[1].Message -match "Not enough free space") {
49     Write-Output $Result.Value[1].Message
50     Throw "Not enough free space to resize partition"
51 }
52 if ($Result.Value[1].Message -match "The partition is already the requested size") {
53     Write-Output $Result.Value[1].Message
```

# Azure Runbook Example

1. Run **Shrink Partition** script on VM.
2. Copy snapshot to new VHD in blob storage.
3. Create new disk and assign to VM.
4. Chop down VHD to size.
5. Thanks Jack Rudlin!

Shrink Windows OS Disk



**Scripted Actions can customize the VM creation  
process for AVD hosts.**

The Power of Scripted Actions



```
446 # Build JSON and txt Configuration Files - These are built here according to the hash table variables specified above.
447
448 # AppX Packages Json
449 $AppxPackages = ($AppxPackages -split "`n").trim()
450 $AppxPackages = $AppxPackages | ConvertFrom-Csv -Delimiter ',' -Header "PackageName", "HelpURL"
451 $AppxPackagesJson = $AppxPackages | ForEach-Object { [PSCustomObject]@{ 'AppxPackage' = $_.PackageName; 'VDIState' = 'Disabled'; 'Description' = $_.PackageName; 'URL' = $_.HelpURL }
452 $AppxPackagesJson | Out-File C:\wdtemp\Optimize_sa\optimize\Virtual-Desktop-Optimization-Tool-main\$WinVersion\ConfigurationFiles\AppxPackages.json
453
454 #AutoLoggers JSON
455 $AutoLoggers = ($AutoLoggers -split "`n").Trim() | ForEach-Object {
456     $LogHash = @{}
457     $BaseKey = 'HKLM:\SYSTEM\CurrentControlSet\Control\WMI\Autologger\' 
458     Switch ($_)
459     {
460         AppModel
461         {
462             $Description = "Used by Packaging, deployment, and query of Windows Store apps. Especially on non-persistent VDI, we tightly control what apps are installed and ava:
463             $URL = "https://docs.microsoft.com/en-us/windows/win32/api/appmodel/"
464             $Disable = $True
465         }
466         CloudExperienceHostOOBE
467         {
468             $Description = "'Cloud Experience Host' is an application used while joining the workplace environment or Azure AD for rendering the experience when collecting your
469             $URL = "https://docs.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-how-it-works-technology#cloud-experience-host"
470             $Disable = $True
471         }
472         DiagLog
473         {
474             $Description = "A log generated by the Diagnostic Policy Service, which is documented here. 'The Diagnostic Policy Service enables problem detection, troubleshootin:
475             $URL = "https://docs.microsoft.com/en-us/windows-server/security/windows-services/security-guidelines-for-disabling-system-services-in-windows-server"
476             $Disable = $True
477         }
478         ReadyBoot
479         {
480             $Description = "'ReadyBoot is boot acceleration technology that maintains an in-RAM cache used to service disk reads faster than a slower storage medium such as a d:
481             $URL = "https://docs.microsoft.com/en-us/previous-versions/windows/desktop/xperf/readyboot-analysis"
482             $Disable = $True
483         }
484         WDIContextLog
485         {
486             $Description = 'This is a startup trace that runs all the time, with these loggers: "Microsoft-Windows-Kernel-PnP":0x48000:0x4+"Microsoft-Windows-Kernel-WDI":0x10000
487             $URL = "https://docs.microsoft.com/en-us/windows-hardware/drivers/network/wifi-universal-driver-model"
```

# Windows Script Example

Virtual Desktop Optimizations



## FIELD APPS PROPERTIES

Directory

AVD

**VM Deployment**

Azure Capacity Extender

RDP settings

FSLogix

Azure Monitor

ControlUp

Sepago

Session time limits

Backup

Disaster Recovery

Run Scripted actions when host VM is CREATED ⓘ

On

These scripted actions will run immediately after VM is created

1. Virtual Desktop Optimizations (20H2) (copy) (Individual with restart) [Nerdio]

Pass AD credentials ⓘ

Nerdio Default Profile (default)

These scripted actions will run after VM is joined to AVD host pool ⓘ

Select...

Pass AD credentials ⓘ

Nerdio Default Profile (default)

Run Scripted actions when host VM is STARTED ⓘ

Off

Run Scripted actions when host VM is STOPPED ⓘ

Off

Run Scripted actions when host VM is REMOVED ⓘ

Off

# Windows Script Example

Virtual Desktop Optimizations



# Lesson 3

---

## Lab

- ***Set FSLogix App Settings***
- ***Windows Script***
- ***Modifies registry settings***

# Modify FSLogix Apps Settings

There are registry settings available for FSLogix. This script will set the following registry keys.

1

**CleanupinvalidSessions**

In cases where a user's session terminates abruptly, ensures the profile can load properly.

2

**RoamRecycleBin**

Creates a redirection for the user's specific recycle bin into the VHD(x) container.

3

**VHDCompactDisk**

Compacts the VHD disk during the sign out operation to decrease the *Size On Disk* of the user's container.

# Lab 1 | Modify FSLogix Settings

```
> Windows Script stub.ps1 > ...
1  ✓ #description: Helpful script
2    #execution mode: IndividualWithRestart
3    #tags: Preview
4
5  ✓ <# Notes:
6    |   This script will do some things!
7  #>
8
9
10   $ErrorActionPreference = 'Stop'
11
12
```

- ✓ Begin by copying the Windows Script "Stub" file.

## Tip:

Setting the **ErrorActionPreference** will ensure the script stops if there's an error



# Lab 1 | Modify FSLogix Settings

1

**CleanupInvalidSessions**

**Registry Hive:**

HKEY\_LOCAL\_MACHINE

**Registry Path:**

SOFTWARE\FSLogix\Apps

**Value Name:**

CleanupInvalidSessions

**Value Type:** DWORD

**Enabled Value:** 1

2

**RoamRecycleBin**

**Registry Hive:**

HKEY\_LOCAL\_MACHINE

**Registry Path:**

SOFTWARE\FSLogix\Apps

**Value Name:**

RoamRecycleBin

**Value Type:** DWORD

**Enabled Value:** 1

3

**VHDCompactDisk**

**Registry Hive:**

HKEY\_LOCAL\_MACHINE

**Registry Path:**

SOFTWARE\FSLogix\Apps

**Value Name:**

VHDCompactDisk

**Value Type:** DWORD

**Enabled Value:** 1

**Hint:** The script will require three PowerShell commands to set the registry / key values.

**Tip:**

Use **Write-Output** before and after each command to provide feedback on what the script is doing and confirm success.

CONFIRM SUCCESS  
THE SUBSCRIPTIONS ARE UP TO DATE

# Lab 1 | Modify FSLogix Settings

## Tip:

For this script, the registry values can be hard-coded to a value of 1. But for scripts where you may want to use different values in different environments, you can define the values as secure variables at the customer account level.

The same script can then be used in each customer environment, pulling customer-specific values from the secure variables.

```
11 $VHDCompactDisk = $SecureVars.VHDCompactDisk  
12  
13 Set-ItemProperty -Path HKLM:\Software\FSLogix\Apps -Type Dword -Name VHDCompactDisk -Value $VHDCompactDisk  
14  
15
```



## Lab 1: Windows Script

# Modify FSLogix Apps Settings

### Testing locally:

It's possible to test Windows scripts by running them locally if you have local admin access to your environment. However, note that scripted actions will ultimately run as the SYSTEM account, which can yield different results than running as a local admin.

Lab 1: Windows Script

# Modify FSLogix Apps Settings

## Testing in NMM:

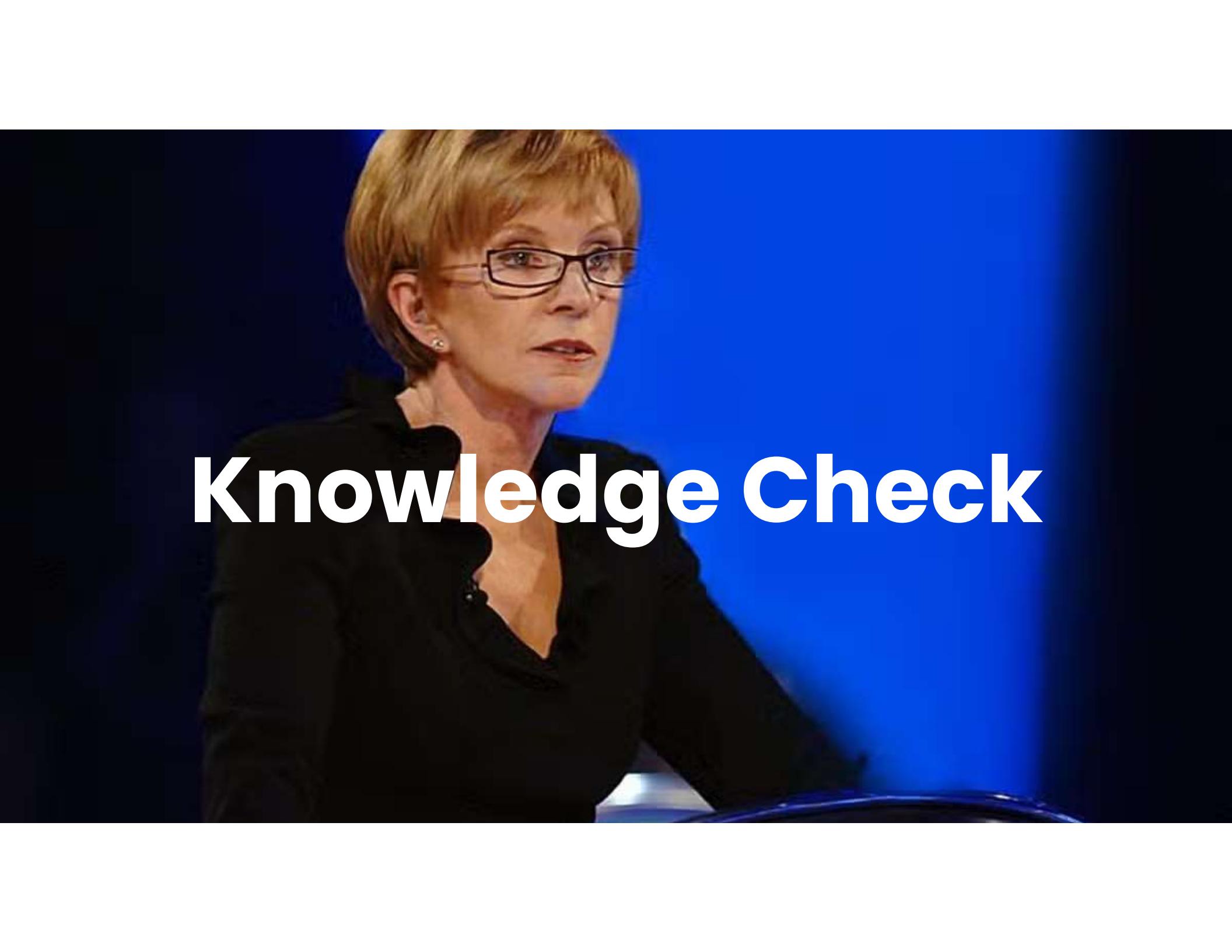
When ready to test your script in NMM, commit your changes to your repository and sync to GitHub. Then initiate a sync in NMM, and your latest changes will be available in the Scripted Actions section.



# Start your scripting!

(and check your lab guides)



A woman with short blonde hair and glasses, wearing a black top, is speaking against a blue background.

# Knowledge Check

# Pre-Lunch Check-in

- Inventory
- Repository
- Accountability
- Validation
- Versioning
- Automation

## Setting Your Foundation

What is Important?

- GitHub
- VSCode
- PowerShell
- Nerdio

## Key Technologies

Core items for your toolbox.

- Context
- Variables
- Integrations
- Controls
- Access
- FSLogix
- Shrink OS Disk
- Disk OS Optimizations

## Scripts & Runbooks

Differences and use cases.





# LUNCH BREAK!



**We are stronger  
together!**

This is advanced training...but we are all at different levels and abilities. Bring your flavor to the forums!

# Community Engagement



## Confirm login to Help Center

Forums & Knowledge Base



## Find your profile & badges

Progress and activity made personal.



## Review the Scripted Actions forums

Check out posts and tips from other MSPs and the Nerdio team!



## Post, Like and Subscribe

Build a better resource by contributing questions, suggestions and more!



# Lesson 4

---

## Lab

***Power on VM for X hours***

- ***Uses runtime parameters***
- ***Uses Az module to modify***

***Azure***





Sometimes it is necessary to have VMs powered on for an extended period, for example when running a centralized patching service once a week.

This script will ensure all hosts in a host pool are powered on for the specified number of hours, regardless of auto-scale settings.

## Power on VMs for X hours

Lab | Azure Runbooks Script



Begin by copying the  
**Azure Runbook Stub** file.

**Tip:**

The “Variables” section is JSON-formatted text that defines the parameters you can provide when running the script. NMM parses this section and provides a UI for entering these values at run time.

```
Windows Script stub.ps1 ● Azure Runbook stub.ps1 ×
Azure Runbook stub.ps1 > ...
1 #description: A runbook script
2 #tags: Preview
3
4 ▼ <# Notes:
5   This script will do things in Azure!
6 #>
7
8 ▼ <# Variables:
9 {
10   "Param1": {
11     "Description": "The first parameter",
12     "IsRequired": true
13   },
14   "Param2": {
15     "Description": "An optional parameter",
16     "IsRequired": false
17 }
18 }
19 #>
20
21 $ErrorActionPreference = 'Stop'
22
23 # Your script here
```

Power on VMs for X hours

Lab | Azure Runbooks Script



# This script should accept the following parameters:

**HostPoolName** – The host pool containing the VMs.

**HostPoolResourceGroup** – The resource group containing the host pool .

**Hours** – An integer indicating how long the hosts should remain on.

**Hint:** The stub file illustrates how to define the above parameters

## Power on VMs for X hours

Lab | Azure Runbooks Script



## This script will...

1. Set a tag on all VMs to exclude hosts from scale-in activity for X hours
2. Power on all hosts

**Hint:** Both of these tasks can be accomplished using [AZ Module](#) commands.

Power on VMs for X hours

Lab | Azure Runbooks Script



## Tip:

When developing an Azure runbook script, you can connect to Azure using your own credentials if your account has permissions to perform the actions required by the script. Do not add the authentication line to the script itself, as Nerdio will take care of authentication for the runbook.

```
PS C:\Users\NWagner> Connect-AzAccount -SubscriptionId xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx
```

# Power on VMs for X hours

Lab | Azure Runbooks Script

## Tip:

If you defined **HostPoolName**, **HostPoolResourceGroup**, and **Hours** in the variables JSON section of the script, you will have access to the variables **\$HostPoolName**, **\$HostPoolResourceGroup**, and **\$Hours** in your script.

```
PS C:\Users\NWagner> $HostPoolName = 'MyHostPool'  
PS C:\Users\NWagner> $HostPoolResourceGroup = 'MyRG'  
PS C:\Users\NWagner> $Hours = 8
```

For testing and development, you can define these variables manually in the console.

# Power on VMs for X hours

Lab | Azure Runbooks Script



## Retrieving the VMs:

You can find all hosts in the host pool using:

- `Get-AzWvdSessionHost`
- `-HostPoolName`
- `$HostPoolName`
- `-ResourceGroupName`
- `$HostPoolResourceGroup`

However, the session hosts name and VM name are not the same. You'll need to derive a list of VM names from the session host names, then retrieve the VMs from Azure using `Get-AzVM`.

Power on VMs for X hours

Lab | Azure Runbooks Script



## Setting a VM tag to exclude from scale-in:

The tag name is 'WAP\_SCALE\_IN\_RESTRICTION' and the value needs to be in a specific format:

*2024-06-15T08:00;Central Standard Time*

```
26 $RestrictUntil = (Get-Date).AddHours([int]$Hours)
27 $TimeZoneId = (Get-TimeZone).id
28
29 Write-output "Setting VM Tags"
30 foreach ($VM in $VMs) {
31     $tags = $vm.tags
32
33     # Add the scale in restriction tag to prevent Nerdio from turning the VMs off
34     $tags["WAP_SCALE_IN_RESTRICTION"] = $RestrictUntil.ToString("yyyy-MM-ddTHH") + ";"$TimeZoneId"
35     Set-AzResource -ResourceGroupName $vm.ResourceGroupName -Name $vm.name -ResourceType "Microsoft.Compute/VirtualMachines" -Tag $tags -Force
36 }
```

Power on VMs for X hours

Lab | Azure Runbooks Script



## Testing in NMM:

When ready to test your script in NMM, commit your changes to your repository and sync to GitHub. Then initiate a sync in NMM, and your latest changes will be available in the Scripted Actions section.

Power on VMs for X hours

Lab | Azure Runbooks Script

# Start your scripting!

(and check your lab guides)



A photograph of a smiling man with light-colored hair, wearing a dark suit and tie, holding a large stack of US dollar bills. He is positioned in front of a purple screen with a grid pattern. The background features stacks of books on either side.

# Knowledge Check



**SNACK TIME**

**YUMMY TIME**

# Lesson 5

---

## Lab

- ***Sample Files Walkthrough***





# Lesson 6

---

## Walkthrough

- ***Kevin Stuff...***
- ***Tags***

# Putting it all together



## Define the issue in a S.M.A.R.T. way

Be Specific, Measurable, Achievable, Relevant and TimeBound



## Take inventory

What do you have, where is it, who owns it's lifecycle and updates?



## Build for Scale

Repos, Scheduling, variables



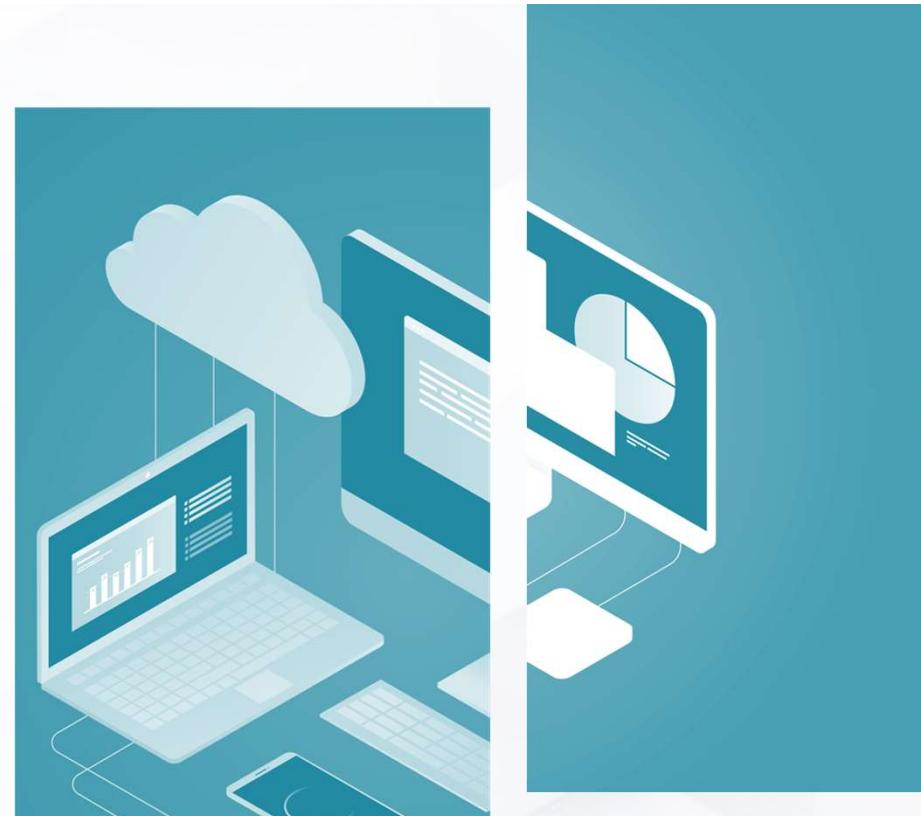
## Validate

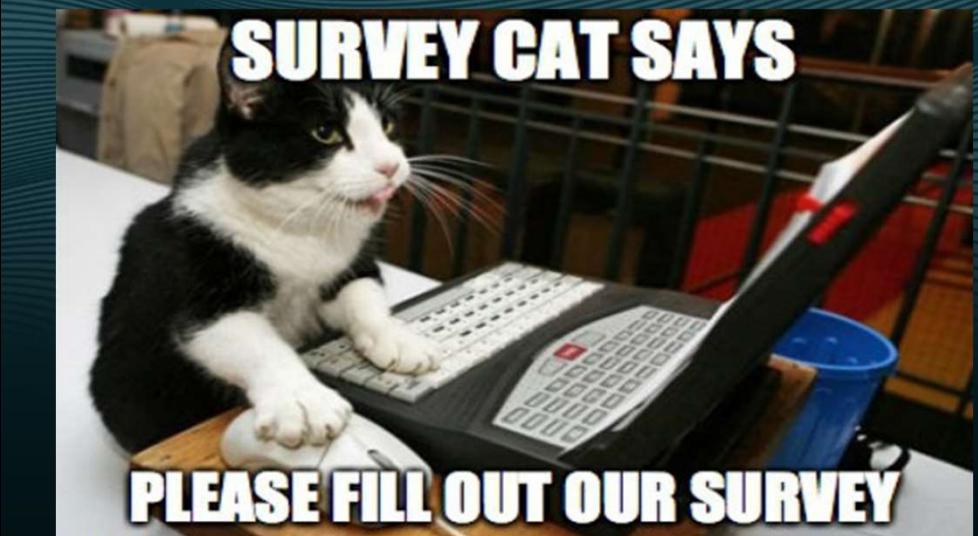
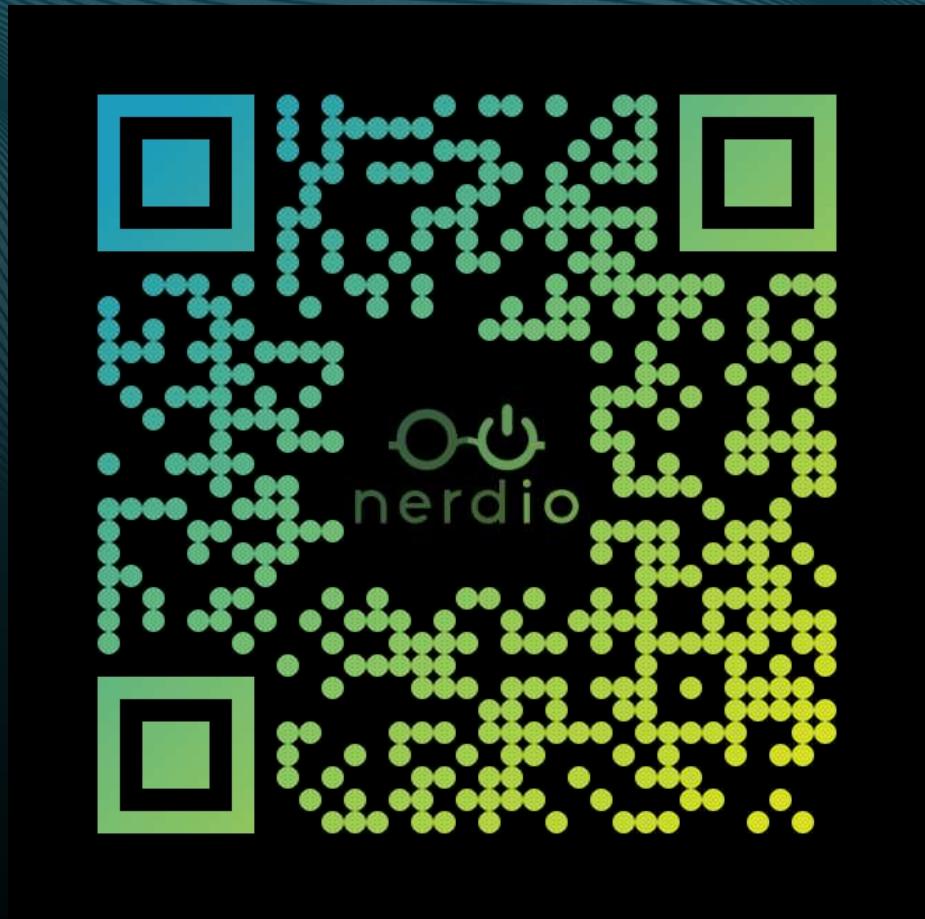
Everything works until it doesn't



## Notify/Track/Report

Create protections, awareness and insights





<http://nerdio.co/altsaeod>

# Ask Us Anything!

