



 23.-25. September 2024

 10:00 – 11:30 Uhr

 Rheingoldhalle, Mainz

SESSION

10 PowerShell One-Liner, die Dir das Microsoft 365-Management erleichtern



Heiko Brenn
ScriptRunner Software GmbH

DANKE AN UNSERE SPONSOREN

DIAMANT SPONSOREN



PLATINUM SPONSOREN



GOLD SPONSOREN



SILBER SPONSOREN



STARTUP SPONSOREN



MEDIEN SPONSOR



About Me

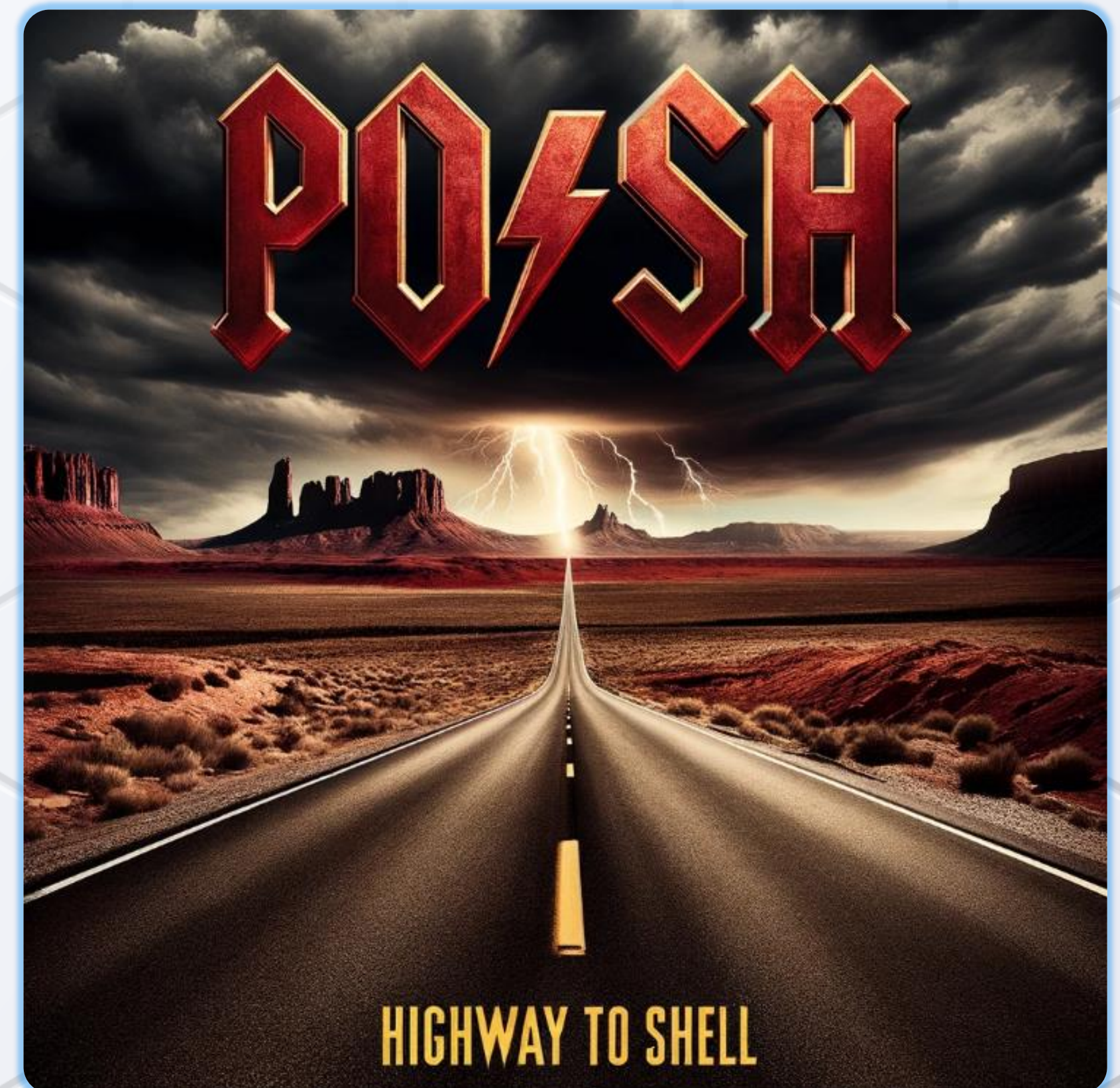
Head of Marketing
Automation Enthusiast

In the IT industry for 25+ years
IT Admin | Consultant | PM | BD | ...

First PowerShell experience:
Exchange 2007

Co-Founder of the PowerShell
User Group Rhein-Neckar

Music Lover



Let's Connect



@HeikoBrenn



@HeikoBrenn



heiko.brenn@scriptrunner.com



Github.com/HeikoBrenn



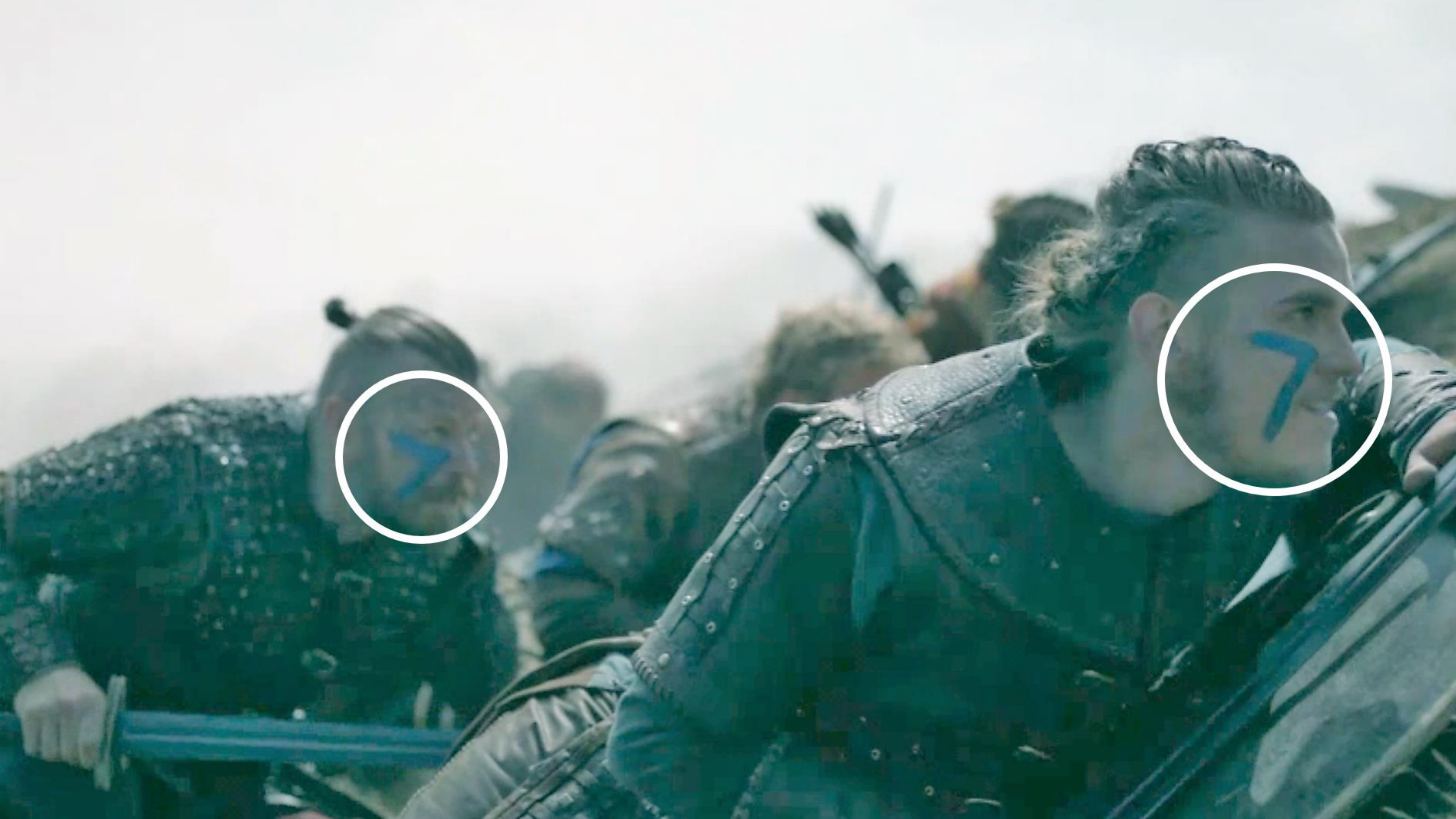
@HeikoBrennPowerShell



AGENDA

- Warum M365 mit PowerShell managen?
- Die wichtigsten M365 PowerShell Module
- PowerShell One-Liner – Was ist das?
- Use Cases für PS One-Liner
- Demotime
- Q&A





Manage M365 With PowerShell – Why?

Automate boring, time-consuming, error-prone repetitive tasks

Bulk Operations

Advanced Configuration Options

Efficient Reporting

Integration and Extensibility



Important M365 PowerShell Modules



AzureAD
Entra PowerShell



Exchange Online
Management



Microsoft Teams



SharePoint.PowerShell
SP PnP PowerShell



Microsoft Graph

What is a PowerShell One-Liner?

Single-line commands or script snippet

Combines multiple PowerShell CmdLets using pipelines ("|")

Chaining multiple PowerShell commands in one line (";")

Efficient for quick administration tasks

Ideal for rapid problem-solving and automation



The Anatomy of a PowerShell One-Liner

Passes the output of
Get-UnifiedGroupLinks
to
ForEach-Object

Operator (i.e. Pipeline)

Defines the actions to be performed
on each object in the pipeline

Script Block

Get-UnifiedGroupLinks -Identity "Finance" -LinkType Members | ForEach-Object { Set-Mailbox -Identity \$_.PrimarySmtpAddress -RetainDeletedItemsFor 30 -WhatIf }

Finding the desired objects

Getting all members of the
specified distribution group

Modifying setting for the desired objects

Setting the new mailbox limit for every object
that was passed through the pipeline

Safety net

Simulates
the "Set"
execution

10 Usefull PowerShell One-Liner Use Cases

Show | Change Mailbox Settings

Tracking Messages

Show | Change Distribution Groups

Create | Delete Mailboxes

Configure Out-of-Office Settings

Show | Change Teams Settings

Archive Teams

Analyzing Teams

Create new SharePoint Site

License Management

The background of the image is a repeating pattern of hexagons, resembling a honeycomb or a tiled surface. The hexagons are arranged in a staggered grid. Most hexagons are a light gray color, but several are a darker gray, creating a subtle contrast. The text "DEMO TIME" is centered in the middle of the image.

DEMO TIME

PS /home/sradmin>




```
PS /home/sradmin> $cred = Get-Credential -Credential scriptrunner@scriptrunnerdemo01.onmicrosoft.com
```


PS /home/sradmin> █



```
PS /home/sradmin> Get-MessageTrace -StartDate ((Get-Date).AddDays(-10)) -EndDate (Get-Date) | Where-Object {$_.Status -eq "Delivered"} | Select-Object Received, SenderAddress, RecipientAddress, Subject, Status | Format-Table -AutoSize
```

```
PS /home/sradmin> Get-Mailbox | ForEach-Object { $mailbox = $_.PrimarySmtpAddress; if (-not (Get-DistributionGroupMember -Identity "IT Administrators" | Where-Object { $_.PrimarySmtpAddress -eq $mailbox }))) { Add-DistributionGroupMember -Identity "IT Administrators" -Member $mailbox; Write-Output "Added $mailbox to IT Administrators" } else { Write-Output "$mailbox is already a member of IT Administrators" } }
```



```
PS /home/sradmin> invoke-oaichat 'Here is a one-liner. Please turn it into a proper script and use splatting: $webhookUrl = "https://scriptrunnerdemo01.webhook.office.com/webhookb2/5f1a1898-a613-46ad-b9aa-e8c03966023c@aad7eb5d-eae1-4de2-8f8f-0694e868ff75/IncomingWebhook/679b42a541cd4b718bc02b7e4a933c0e/d581a13d-546f-4b46-9b24-e76f9296ee18"; $userInfoList = @(); Get-DistributionGroupMember -Identity "IT Administrators" | ForEach-Object { Set-MailboxAutoReplyConfiguration -Identity $_.PrimarySmtpAddress -AutoReplyState Enabled -InternalMessage "I am currently out of the office and will respond to your email upon my return." -ExternalMessage "I am currently out of the office. I will respond to your message when I return."; $userInfoList += "Auto-reply enabled for: $($_.PrimarySmtpAddress)<br>" }; $messageBody = @{ text = $userInfoList -join "`n" } | ConvertTo-Json; Invoke-RestMethod -Uri $webhookUrl -Method Post -ContentType 'application/json' -Body $messageBody | Out-Null'
```


PS /home/sradmin> █



PS /home/sradmin> Get-Team | Where-Object {\$_.Displayname -Like "Test*"} | ForEach-Object { if (-not (Get-TeamChannel -GroupId \$_.GroupId | Where-Object { \$_.DisplayName -eq "NewChannel02" }))) { New-TeamChannel -GroupId \$_.GroupId -DisplayName "NewChannel02" -MembershipType Shared } }

Windows PowerShell

sradmin@NBKSRITP01: ~

```
PS /home/sradmin> Get-Team | ForEach-Object { $groupId = $_.GroupId; $teamName = $_.DisplayName; $user = "angusyoung@scriptrunnerdemo01.onmicrosoft.com"; if
(Get-TeamUser -GroupId $groupId | Where-Object { $_.User -eq $user }) { Write-Host "$user is already a member of the team '$teamName'" } else { Add-TeamUser
-GroupId $groupId -User $user -Role Owner; Write-Host "$user has been added as an owner to the team '$teamName'" } }
```

```
PS /home/sradmin> #Find Teams with only one owner and show more details
PS /home/sradmin> Get-Team | Where-Object { (Get-TeamUser -GroupId $_.GroupId | Where-Object { $_.Role -eq 'Owner' }).Count -eq 1 } | Select-Object DisplayName, @{Name='Owner'; Expression={(Get-TeamUser -GroupId $_.GroupId | Where-Object { $_.Role -eq 'Owner' }).User}}, Visibility, Description, Archived | Format-Table -AutoSize
```




Windows PowerShell

sradmin@NBKSRITP01: ~

```
PS /home/sradmin> if (-not (Get-PnPTenantSite -Url "https://scriptrunnerdemo01.sharepoint.com/sites/NewSite004" -ErrorAction SilentlyContinue)) { New-PnPSite
-Type CommunicationSite -Title "New Site 004" -Url "https://scriptrunnerdemo01.sharepoint.com/sites/NewSite004" -Owner "sradmin@scriptrunnerdemo01.onmicroso
ft.com"; Write-Host "Site created." } else { Write-Host "Site already exists." }
```



```
PS /home/sradmin> Get-PnPTenantSite | Where-Object { $_.Template -eq "SITEPAGEPUBLISHING#0" -AND $_.Url -like "*NewSite30*" } | Select-Object Title, Url, Owner, Template
```

```
PS /home/sradmin> #Show License infos for all users
PS /home/sradmin> Get-MgUser -All | ForEach-Object { $userUPN = $_.UserPrincipalName; Get-MgUserLicenseDetail -UserId $_.Id | Select-Object @{Name='UserPrincipalName';Expression={$userUPN}}, SkuId, ServicePlans }
█
```



```
PS /home/sradmin> $joke = "**Chuck Norris Joke of the Day:**<br>$((Invoke-RestMethod https://api.chucknorris.io/jokes/random).value)"; $null = Invoke-RestMethod -Uri "https://scriptrunnerdemo01.webhook.office.com/webhookb2/5f1a1898-a613-46ad-b9aa-e8c03966023c@aad7eb5d-eae1-4de2-8f8f-0694e868ff75/IncomingWebhook/679b42a541cd4b718bc02b7e4a933c0e/d581a13d-546f-4b46-9b24-e76f9296ee18" -Method Post -ContentType 'application/json' -Body (@{summary = "Chuck Norris Joke"; themeColor = "00FF00"; text = $joke} | ConvertTo-Json)
```

Let's Connect



@HeikoBrenn



@HeikoBrenn



heiko.brenn@scriptrunner.com



Github.com/HeikoBrenn



@HeikoBrennPowerShell



**Vielen Dank für Eure
Aufmerksamkeit!**





Geben Sie uns Feedback

1. Barcode mit dem Handy abfotografieren
2. Kleines Video (1 min) als Feedback aufnehmen
3. Name und Titel eintragen







M365 Summit MAINZ 2024

Die Konferenz rund um Microsoft® 365

www.m365-summit.de