



PowerShell Kurs

14. – 16. Oktober 2024



Darf's noch etwas mehr sein?

- Learning Accelerator Network for Microsoft Solutions
 - mlan-germany.de
- Windows Server Launch Tour - 04.-14.11.2024:
 - [Windows Server 2025 Launch Tour](#) (MUC, Sindelfingen, FRA, HAM, LEJ, DUS)
- Azure Hybrid Cloud Workshop - 27.-29.01.2025:
 - [Azure Hybrid Workshop](#) (online)
- Windows Server Summit 2025 - 09.-10.04.2025 :
 - windows-server-summit.de (Leipzig)
- Experts Live Germany 2025 - 11.04.2025 :
 - expertslive.de (Leipzig)
- Mehr PowerKurse: powerkurs.net



TRAINER



Carsten Rachfahl

Carsten Rachfahl ist seit 1988 in der IT-Branche tätig. In dieser Zeit hat er sich mit den unterschiedlichsten Themen auseinandergesetzt. Aktuell liegt der Schwerpunkt seiner Expertise auf der Microsoft Virtualisierungslösung Hyper-V, Storage Spaces Direct und Azure Stack.

In seinen Webinaren, Videos und Blogpost, teilt er sein Wissen mit einer großen Community. Dafür wurde er von 2011 beginnend, jedes Jahr mit dem Microsoft Most Valueable Award ausgezeichnet. In seinem Fachgebiet war er der erste deutsche MVP. Das coole daran ist, der Kontakt zu der Microsoft Produktgruppe, die für die Entwicklung und Vermarktung der Technologie zuständig ist und wofür er einmal im Jahr zu einem Treffen nach Redmond fliegt.

Seinen großen Traum erfüllte er sich 2016 mit der Organisation der Cloud & Datacenter Conference Germany. Einmal im Jahr trommelt er jährlich hochkarätige Sprecher aus dem In- und Ausland zusammen und bietet so den Teilnehmern und Teilnehmerinnen ein breites Spektrum an Wissen zu dieser Thematik.

Auch er selbst wird regelmäßig als Sprecher auf Microsoft-Veranstaltungen sowie internationalen Konferenzen eingeladen. Er ist Gründungsmitglied der deutschen Hyper-V Community.



Eric Berg



Eric ist Microsoft MVP für Microsoft Azure und für Cloud und Datacenter Management. Er arbeitete in internationalen Leadership-Rollen und ist Inhaber der Mountain IT. Seine Hauptgebiete sind: Microsoft Azure, Cloud Governance, Automatisierung, hybride Infrastruktur, Windows Server und Managementlösungen.

Eric ist ein aktives Mitglied der Microsoft-Community. Er bloggt regelmäßig auf seiner Website www.ericberg.de. Er produziert den GeekSprech Podcast und den GeekSchau Webcast. Er leitet das Azure Thüringen Meetup in Deutschland, mit mehr als 400 Mitgliedern und monatlichen Meetups. Er ist ein bekannter und hochgeschätzter Redner auf Konferenzen wie Microsoft Ignite, MVP Summit, Microsoft Ignite The Tour, Expressive, Azure Saturday, Cloud and Datacenter Conference und vielen mehr.

Eric ist auch Trainer bei LinkedIn Learning und bietet Video-Trainings rund um Microsoft Azure an.

Sein Motto:

„Die Community lebt von der Community und nur wer mitmacht, hält sie am Leben!“



ABLAUF



Tag 1 – 14. Oktober

| | |
|---------------|---|
| 10:00 – 10:30 | Begrüßung, Intro, Erwartungen |
| 10:30 – 12:00 | Einführung PowerShell |
| 12:00 – 13:00 | <i>MITTAGSPAUSE</i> |
| 13:00 – 14:00 | Lab 01 – Zugriff und Einrichtung |
| 14:00 – 14:15 | <i>PAUSE</i> |
| 14:15 – 14:45 | System Administration |
| 14:45 – 15:45 | Lab 02 – AD, DNS & VM |
| 15:45 – 16:00 | <i>PAUSE</i> |
| 16:00 – 17:00 | PowerShell Pipeline |

Tag 2 – 15. Oktober

| | |
|---------------|-------------------------------|
| 09:00 – 09:15 | Recap + Q&A |
| 09:15 – 09:45 | Lab 03 - Pipelines |
| 09:45 – 10:00 | <i>PAUSE</i> |
| 10:00 – 10:30 | Variablen |
| 10:30 – 11:30 | Scripting |
| 11:30 – 12:00 | Lab 04 – Erstes Skript |
| 12:00 – 13:00 | <i>MITTAGSPAUSE</i> |
| 13:00 – 13:30 | Lab 04 – Fortsetzung |
| 13:30 – 14:30 | Parameter & Dokumentation |
| 14:30 – 14:45 | <i>PAUSE</i> |
| 14:45 – 15:30 | PS-Remoting |
| 15:30 – 16:00 | Debugging |

Tag 3 – 16. Oktober

| | |
|---------------|---|
| 09:00 – 09:15 | Recap + Q&A |
| 09:15 – 09:45 | Lab 05 – Parameter, Doku und Debugging |
| 09:45 – 10:00 | <i>PAUSE</i> |
| 10:00 – 10:30 | Azure DevOps Intro |
| 10:30 – 11:30 | Git in Azure DevOps Repos |
| 11:30 – 12:00 | Lab 06 – Repos nutzen |
| 12:00 – 13:00 | <i>MITTAGSPAUSE</i> |
| 13:00 – 13:30 | Azure Cloud Shell |
| 13:30 – 14:15 | GitHub Copilot & AI Nutzung |
| 14:15 – 14:30 | <i>PAUSE</i> |
| 14:30 – 15:30 | Hyper-V & Network ATC |
| 15:30 – 16:00 | Abschluss & Feedback |



INTRO



Kennenlernen

- Name?
- Firma?
- Job?
- Privat?
- PowerShell Skill 1 – 6 (Schulnoten)
- Windows Server Skill 1 – 6 (Schulnoten)
- Erwartung? Wann lohnt sich der Kurs für dich?



LOS GEHT'S





Einführung PowerShell

Grundlagen



PowerShell Einführung

Automatisierungs-Lösung:

- Kommandozeilen-Konsole
- Skriptsprache
- Konfigurationsmanagement Framework

PowerShell Kommandos

Kommandos:

- Cmdlets
- Funktionen
- Filter
- Workflows

Windows PowerShell vs. PowerShell

| | Windows PowerShell | PowerShell |
|-----------------|--------------------|-----------------------|
| Basis | .NET Framework | .NET Core |
| Betriebssysteme | Windows | Windows, MacOS, Linux |
| Entwicklung | mit Windows | Open Source |

PowerShell Applikationen

- Windows PowerShell Konsole
- Windows PowerShell ISE
- PowerShell Core (nicht in ISE unterstützt)
- PowerShell in Visual Studio Code



Einführung PowerShell

Syntax



Cmdlet Struktur

- Verben = Aktionen
 - Get
 - Set
 - New
 - Add
- Substantiv = Resource
 - Service
 - Process

Parameter

- Modifizieren die Aktionen der Cmdlets
- Beginnen mit “-”
- **Optionale** und **Erforderliche** Parameter
- Positional Parameter brauchen keinen Namen
- Tab-Completion hilft bei Parametern

Hilfe about

- Hilfe für Konzepte, Techniken und Features
- Beginnen mit about_
- Get-Help about*
- Gern im Laufe des Kurses auf diese Files zurückkommen ;-)



Einführung PowerShell

Befehle



Module

- Container für Cmdlets
- Müssen in Session geladen sein
- Können automatisch geladen werden
- Haben ggf. Unterschiedliche Funktionen je nach OS
- Pfad: `$Env:PSModulePath`
- `Get-Module`

Cmdlets

- Get-Command
 - -Noun
 - -Verb
 - -Module
 - Findet Cmdlets
- Get-Help
 - Durchsucht Hilfe
 - Nur verwenden, wenn Get-Command nichts findet

Aliase

- Dir
- Cd
- Mkdir
- Type
- ... sind in PowerShell nur Aliase für PowerShell cmdlets
- Get-Alias

Get-Help

Parameter set

Mandatory parameter

```
NAME
    Get-EventLog

SYNOPSIS
    Gets the events in an event log, or a list of the event logs, on

SYNTAX
    Get-EventLog [-LogName] <String> [[-InstanceId] <Int64[]>] [-After
    [-Before <DateTime>] [-ComputerName <String[]>] [-EntryType <Stri
    [-Newest <Int32>] [-Source <String[]>] [-UserName <String[]>] [<C
    Get-EventLog [-AsString [<SwitchParameter>]] [-ComputerName <Stri
    [<CommonParameters>]
```

Positional parameter

Optional parameter

Wichtige Commands

- Dir & Get-ChildItem
- Get-Alias & New-Alias
- Show-Command
- Get-Help -Examples & -Full & -ShowWindow
- Update-Help



Lab 01

Zugriff und Einrichtung



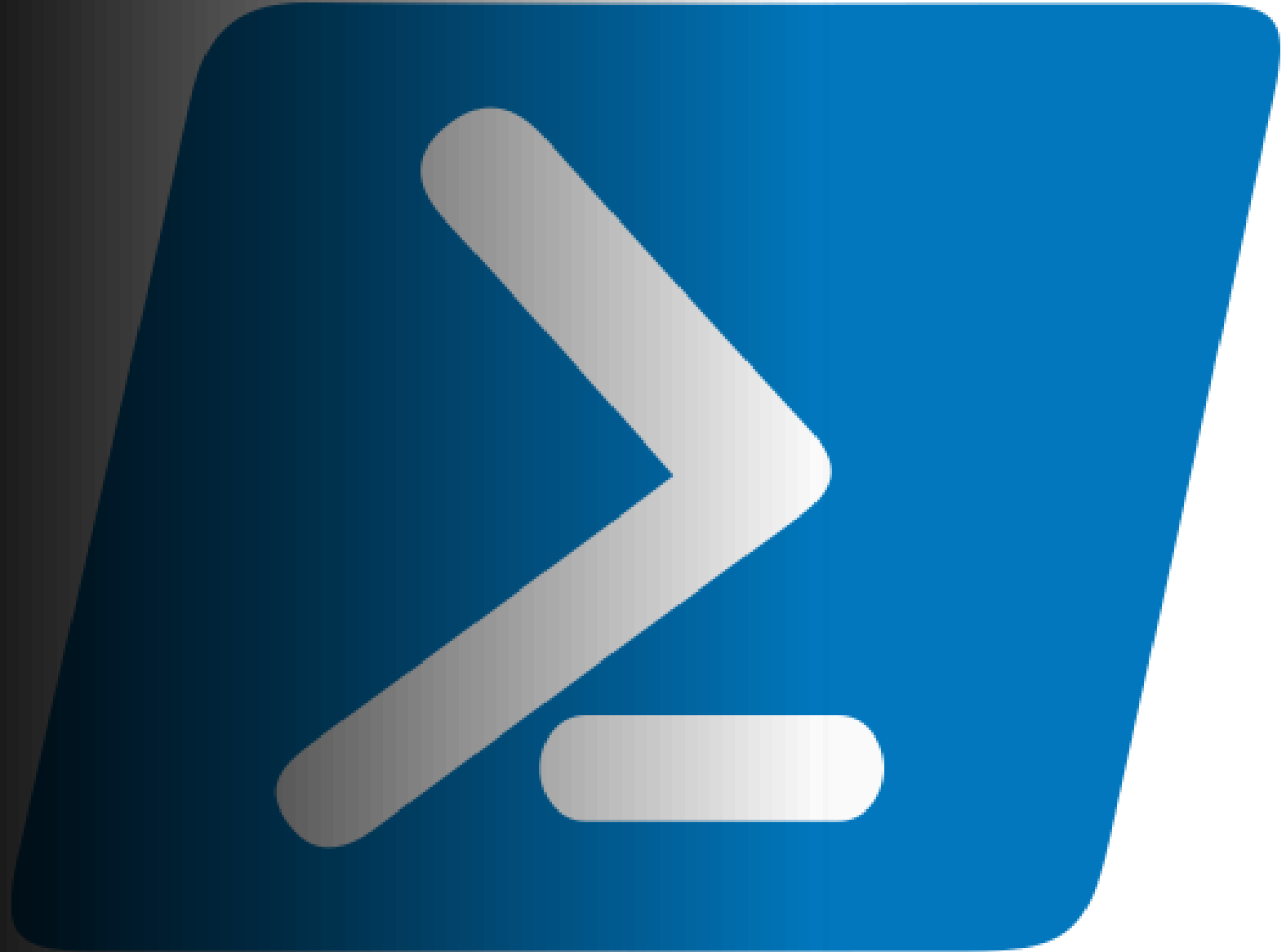
Lab 01

- Zugriff auf Lab sicherstellen
- PowerShell 7.x installieren
- VSCode + PowerShell Erweiterungen installieren
- Terminal Installieren
- Befehl finden, um Network Adapter umzubenennen (SMB1 → Storage1; etc)
- Ist „Name“ required? Positional?
- Welche Aliase gibt es für Get-ChildItem? Und erstellt einen Neuen!



Lab 02

AD, DNS, VM

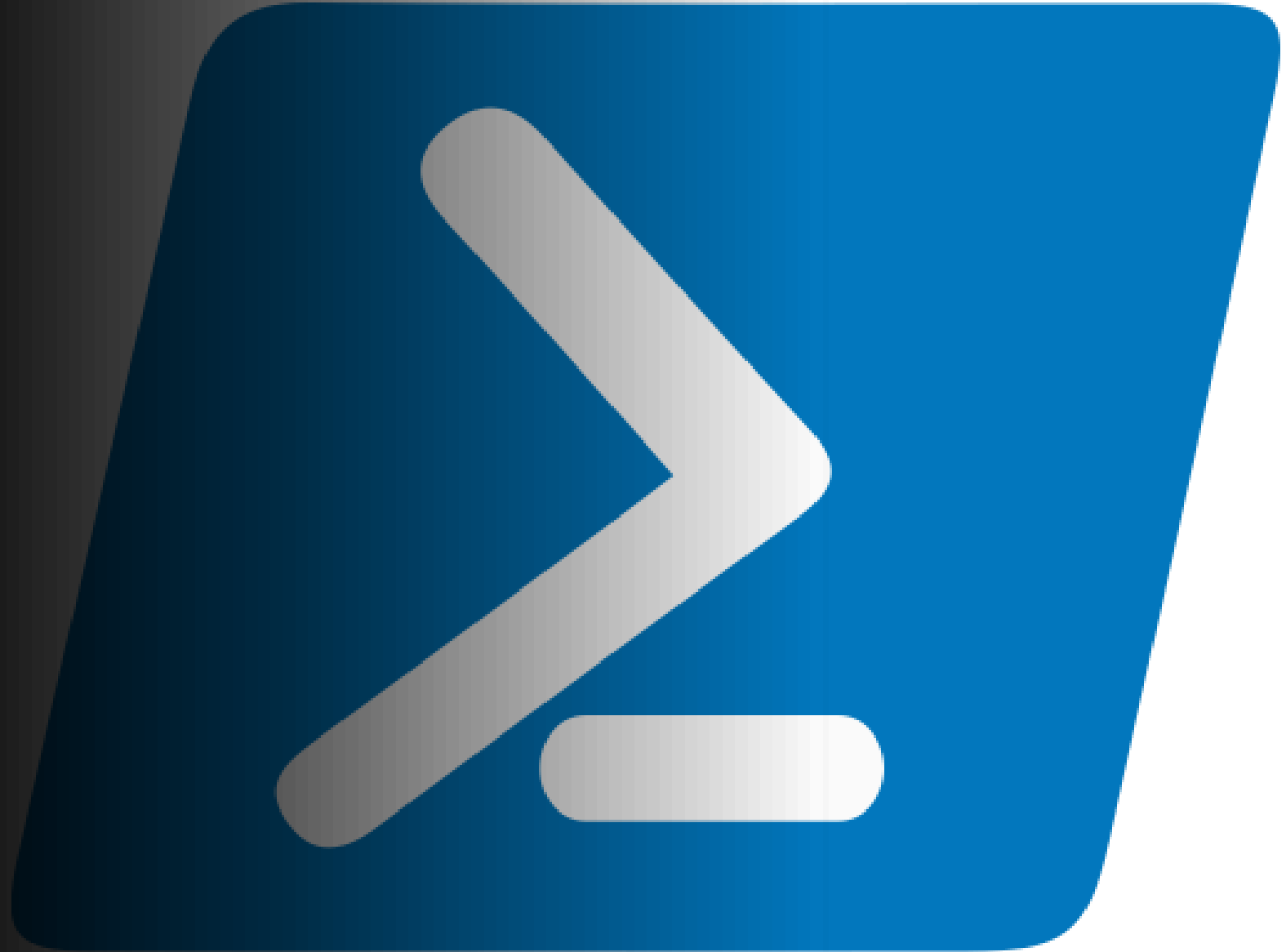


Lab 02

- Erstelle eine neue Hyper-V VM
- Richte deine eigene Domäne ein
- Erstelle 5 Test-User im AD
- Erstelle einen neuen DNS Eintrag
- Erstelle ein Transcript für alle Aufgaben



PowerShell Pipeline



Pipeline

- Kommando-Zeilen sind Pipelines
- Trennung von Kommandos mit “|”
- Ausführung von links nach rechts
- Ausgabe wird “gepiped” (weitergegeben)
- Bspw. Get | Set

Pipeline Ausgabe

- Ausgaben sind Objekte
- Objekte sind wie eine Tabelle

The diagram illustrates the structure of pipeline output objects. A table represents the data, with annotations pointing to its components:

- Object**: Points to the entire table structure.
- Property**: Points to the **DisplayName** column header.
- Collection**: Points to the vertical line on the right side of the table, representing the collection of objects.

| Name | Status | DisplayName |
|-------|---------|---------------------------|
| WinRM | Running | Windows Remote Management |
| VDS | Running | Virtual Disk |

Object Member

- Objekte haben
 - Eigenschaften
 - Methoden
 - Events
- “Get-Member”

Ausgaben formatieren

- Format-List
- Format-Table
- Format-Wide
- -Property Parameter als Ausgabe-Filter

Select, Sort, Measure, Filter

- Sort-Object
 - Group-Object
- Measure-Object (-Property)
 - -Average
 - -Minimum
 - -Maximum
 - -Sum

Select, Sort, Measure, Filter

- Select-Object (-First, -Last, -Skip, -Unique)
- Select-Object -Property
- Kalkulierte Properties
 - Label
 - Expression
 - \$PSItem (oder \$_)

Hash table

Label key

Label string
value

@{

n='VirtualMemory';

e={ \$PSItem.VM }

}

Semicolon

Expression key

Expression
script block

Select, Sort, Measure, Filter

| Comparison type | Case-insensitive operator | Case-sensitive operator |
|--------------------------|---------------------------|-------------------------|
| Equality | -eq | -ceq |
| Inequality | -ne | -cne |
| Greater than | -gt | -cgt |
| Less than | -lt | -clt |
| Greater than or equal to | -ge | -cge |
| Less than or equal to | -le | -cle |
| Wildcard equality | -like | -clike |

Where-Object

Ausgaben

- Out-File
- ConvertTo-CSV / Export-CSV
- ConvertTo-JSON
- ConvertTo-HTML
- Out-Host, Out-Printer, Out-GridView

Merken

- GET-COMMAND
- GET-HELP
- GET-MEMBER



PowerShell Kurs

14. – 16. Oktober 2024

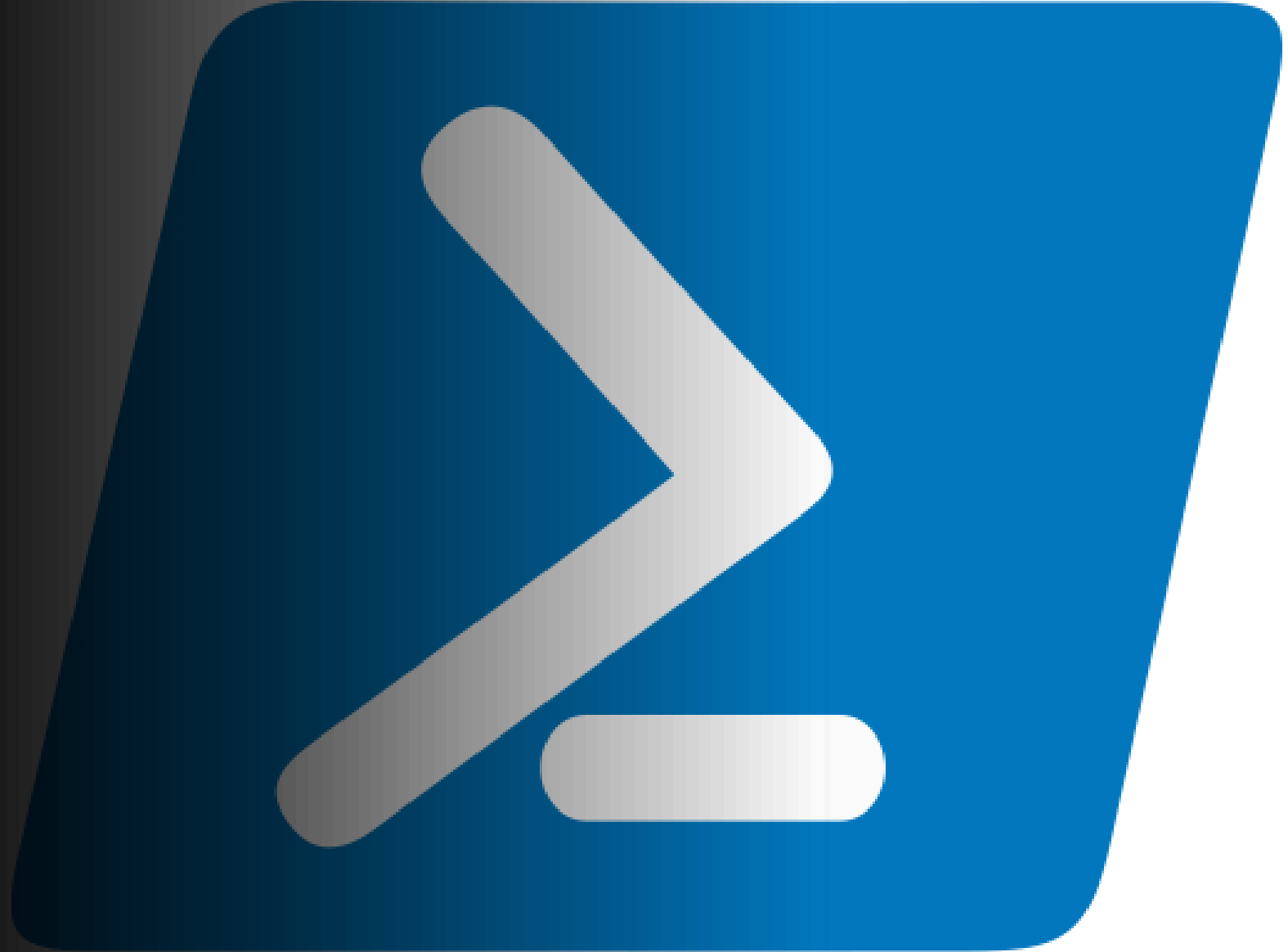


Tag 2 – 15. Oktober

| | |
|---------------|-------------------------------|
| 09:00 – 09:15 | Recap + Q&A |
| 09:15 – 09:45 | Lab 03 - Pipelines |
| 09:45 – 10:00 | <i>PAUSE</i> |
| 10:00 – 10:30 | Variablen |
| 10:30 – 11:30 | Scripting |
| 11:30 – 12:00 | Lab 04 – Erstes Skript |
| 12:00 – 13:00 | <i>MITTAGSPAUSE</i> |
| 13:00 – 13:30 | Lab 04 – Fortsetzung |
| 13:30 – 14:30 | Parameter & Dokumentation |
| 14:30 – 14:45 | <i>PAUSE</i> |
| 14:45 – 15:30 | PS-Remoting |
| 15:30 – 16:00 | Debugging |



Recap + Q&A



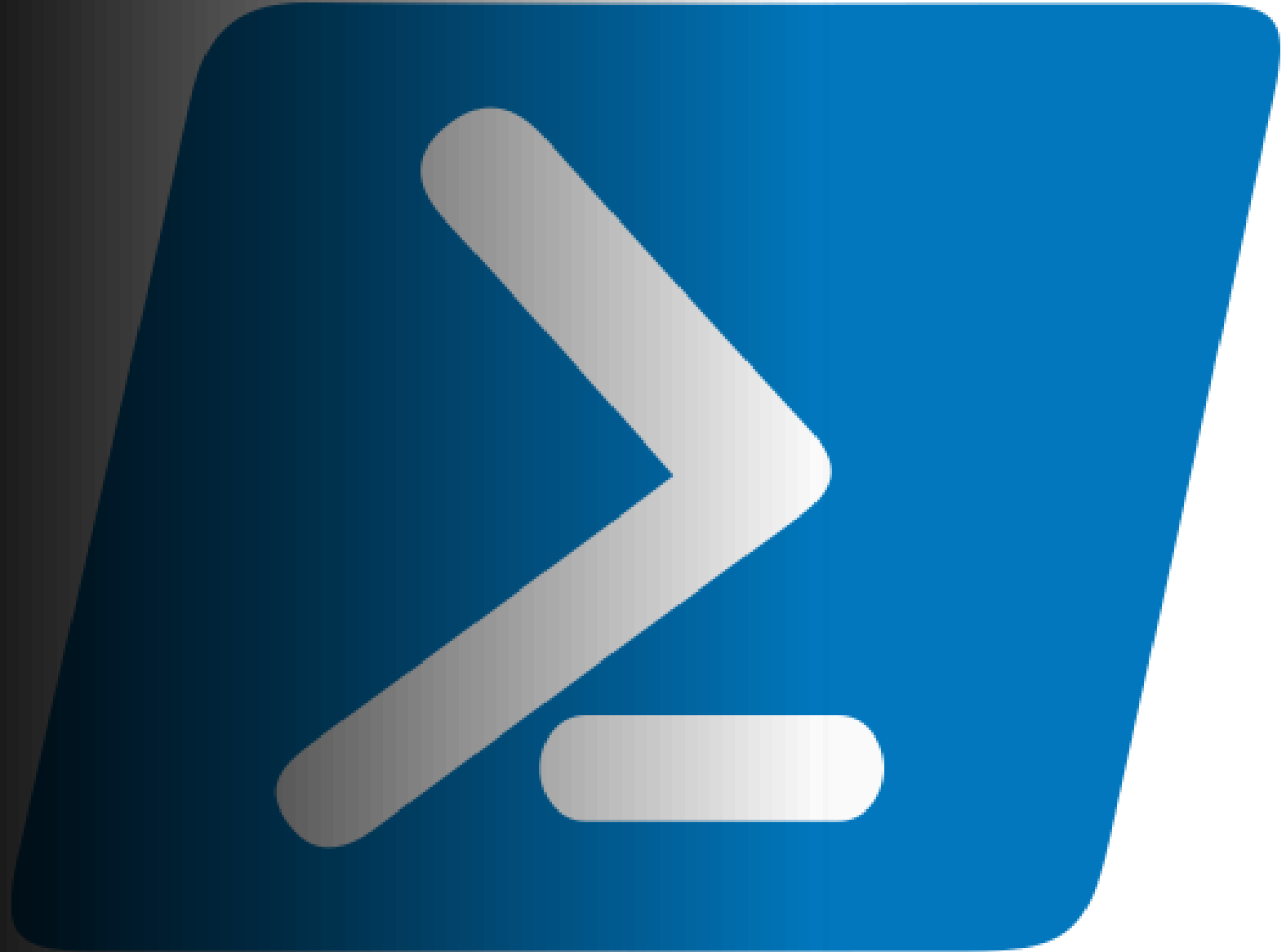
Recap

- GET-COMMAND
- GET-HELP
- GET-MEMBER



Lab 03

Pipelines

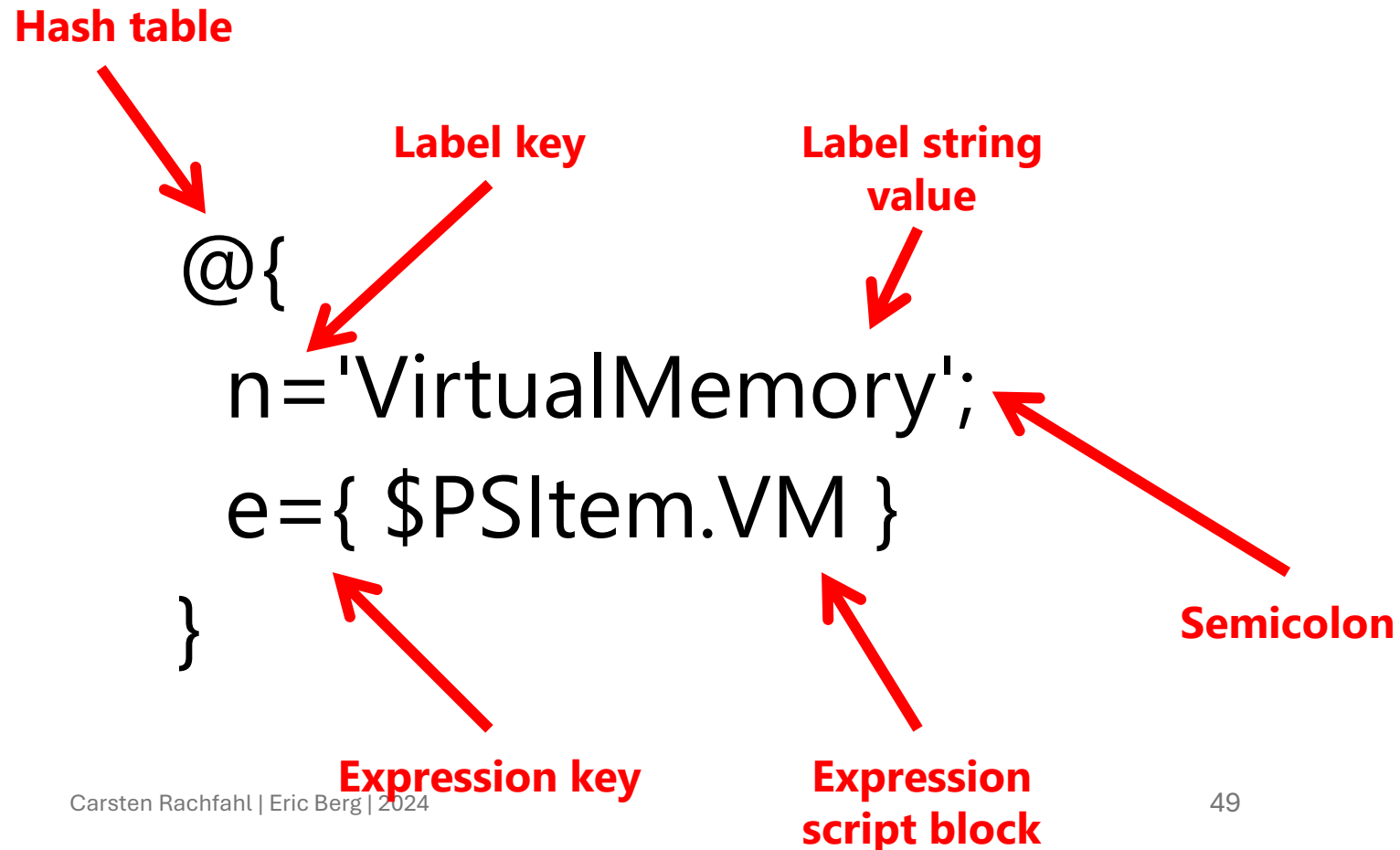


Lab 03

- Importiere eine CSV mit zu stoppenden Prozessen und beende die Prozesse in einer Zeile
- Welcher Befehl ist schneller? Beweise?
 - `Get-Process -Name Notepad`
 - `Get-Process | Where-Object Name -eq Notepad`
- Erstelle eine Liste aller Services mit Namen kleiner 5 Buchstaben
 - Warum funktioniert `Get-Service | Where Name.Length -lt 5` nicht?

Calculated Property

- Kalkulierte Properties
 - Label
 - Expression
 - \$PSItem (oder \$_)
- Nützlich für Passing
ByPropertyName



Calculated Property

```
Get-Volume |  
Select-Object -Property DriveLetter,  
    @{  
        n='Size(GB)';  
        e='{0:N2}' -f ($PSItem.Size / 1GB)}  
    },  
    @{  
        n='FreeSpace(GB)';  
        e='{0:N2}' -f ($PSItem.SizeRemaining / 1GB)}  
    }
```

Wichtige Commands

- `Import-Csv -Path "C:\Code\proc.csv" -Delimiter ";" | stop-process`
- `Import-Csv -Path "C:\Code\proc.csv" -Delimiter ";" | get-process | foreach kill`
- `Measure-Command {get-service -name winrm}`
- `Measure-Command {get-service | where name -eq winrm}`
- `get-service | where -Filter {$PSItem.Name.Length -lt 5}`
- `Get-service | foreach name`
- `Get-EventLog -List | where Log -eq System | ForEach Clear`



Variablen



Variablen

- Variablen speichern Werte oder Objekte im Speicher
- Objekt-Eigenschaften in einer Variable sind zugreifbar
- Variablen sind in einem PSDrive enthalten
 - cd Variable:

Variablen

- Sollten leicht verständlich sein
- Können Leerzeichen enthalten wenn in Klammern → NICHT MACHEN
- Nur alphanumerische Werte
- Nicht case-sensitive
- Oftmals mit CamelCasing
 - \$ComputerName
 - \$StartDate
 - \$TenantID

Variablen Zuweisen

- Mathematische Operationen mit Wertzuweisung durch “=”
 - `$zahl = 5`
 - `$Pfad = "C:\Code"`
 - `$service = Get-Service W32Time`
- Zugriff auf Variable
 - `$zahl`
 - `Write-Host "Die Zahl ist $zahl"`

Variablen Zuweisen

`$string = "Hallo"`

`$string → ???`

`Set-Variable $string -Value "Welt"`

`$string → ???`

Variablen Typen

- String → speichert Text inkl. Sonderzeichen
- Int32 → Integer ohne Kommastellen
- Double → Zahlen mit Kommastellen
- DateTime → Datum und Zeit Werte
- Bool → true oder false

Typ wird automatisch gewählt, oder kann spezifiziert werden

[string]\$zahl = 5 #“5“ wird als String gespeichert

Arbeit mit Strings

- Strings haben nur eine Eigenschaft „Length“
- Aber viele Methoden
 - Contains(string value)
 - Insert(int startindex,string value)
 - Remove(int startindex,int count)
 - Replace(string value,string value)
 - Split(char separator)
 - ToLower()
 - ToUpper()

Arbeit mit Datum und Zeit

- Get-Date gibt aktuelles Datum und Zeit zurück
- DateTime Object hat Eigenschaften wie Date, DayOfWeek, Month, etc.
- Viele Methoden
 - AddDays(double value)
 - AddMonths(int value)
 - ToShortDateString()

Arrays

- Arrays enthalten mehrere Werte oder Objekte

- Liste: `$Procs = "notepad","firefox","iexplore"`

- Output: `$VMs = Get-Vm`

- Leeres Array: `$array = @()`

- Variable: `[Array]$user = "user01"`

- Zugriff auf einzelne Werte

- `$Procs[0]`

- `$Procs.Remove("notepad")`

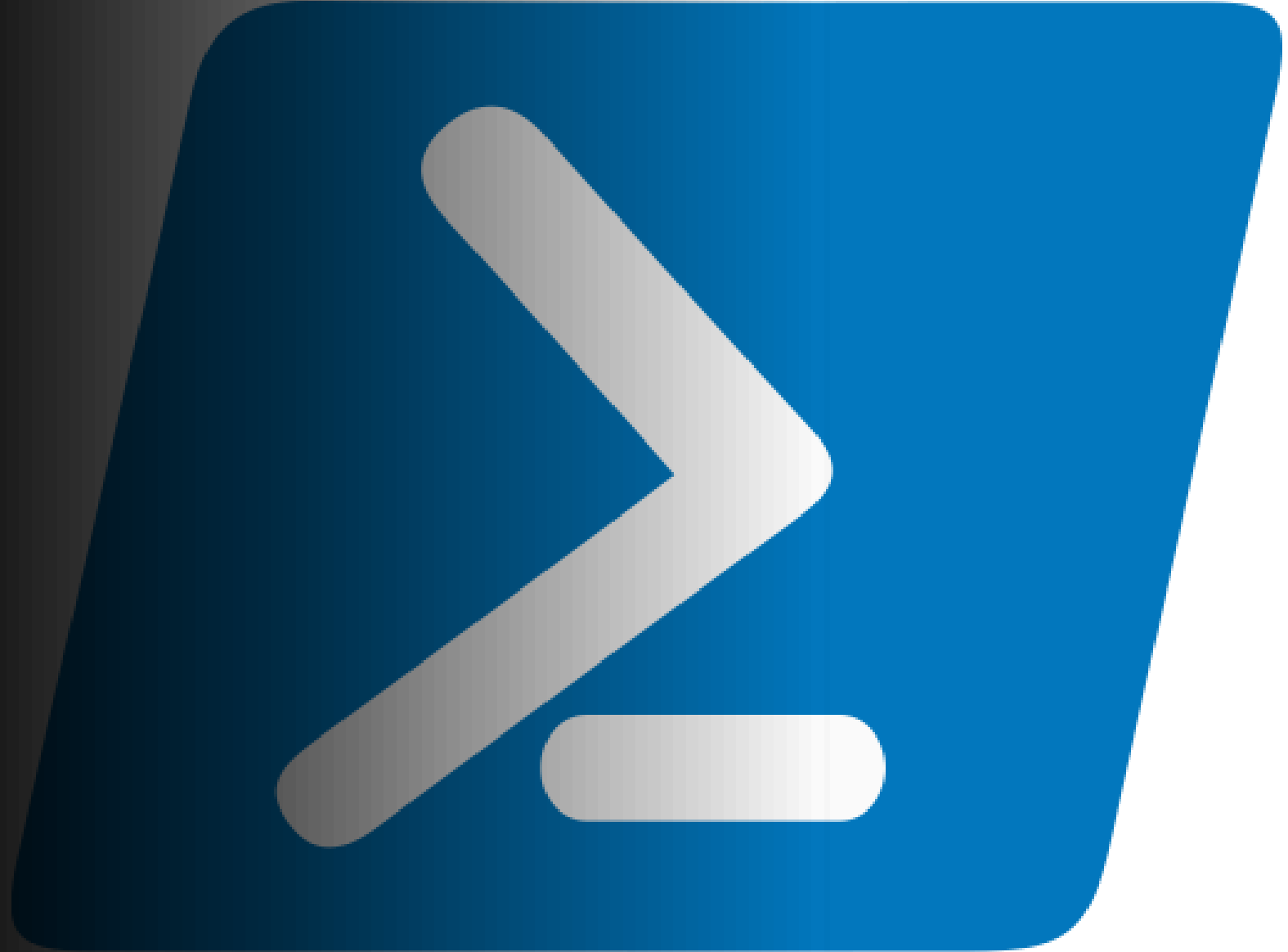
Hash Tables

- Liste von Name-Wert-Paaren
- Zugriff via `$users.['pkuser1']`
- Leere Table: `$hash = @{}`

| Key | IP address |
|---------|-----------------|
| pkuser1 | 192.168.209.201 |
| pkuser2 | 192.168.209.202 |
| pkuser3 | 192.168.209.203 |



Scripting



Skripte

- Skripte werden verwendet für
 - Wiederholende Aufgaben
 - Komplexe Aufgaben
 - Reporting
- PowerShell Skripte
 - Verwenden ForEach, If, und Switch
 - Haben die .ps1 Dateierweiterung

Skript Sicherheit

- .ps1 Dateierweiterung mit Editor verknüpft
- Ausführung von Skripten via
 - Pfad: C:\Code\Script.ps1
 - Relativer Pfad: \Code\Script.ps1
 - Aktueller Ordner: .\Script.ps1
- Execution Policy beachten
 - Unblock-File
 - Powershell.exe -ExecutionPolicy Bypass

Skript Konstrukte

- ForEach (\$user in \$users) {}
- If (\$VMName -eq 'DC01') {} Else {}
- Switch (\$auswahl) {
1{XXX}
2{YYY}
Default {ZZZ}
}
- For (\$i=1; \$i -le 10; \$i++) {}
- Do...While Do..Until While

User Interaktion

- Read-Host
- Get-Credential
 - Store Credentials via \$cred | Export-Clixml C:\cred.xml oder via SecretManagement module
- Out-GridView
 - \$Auswahl = \$procs | Out-GridView -PassThru

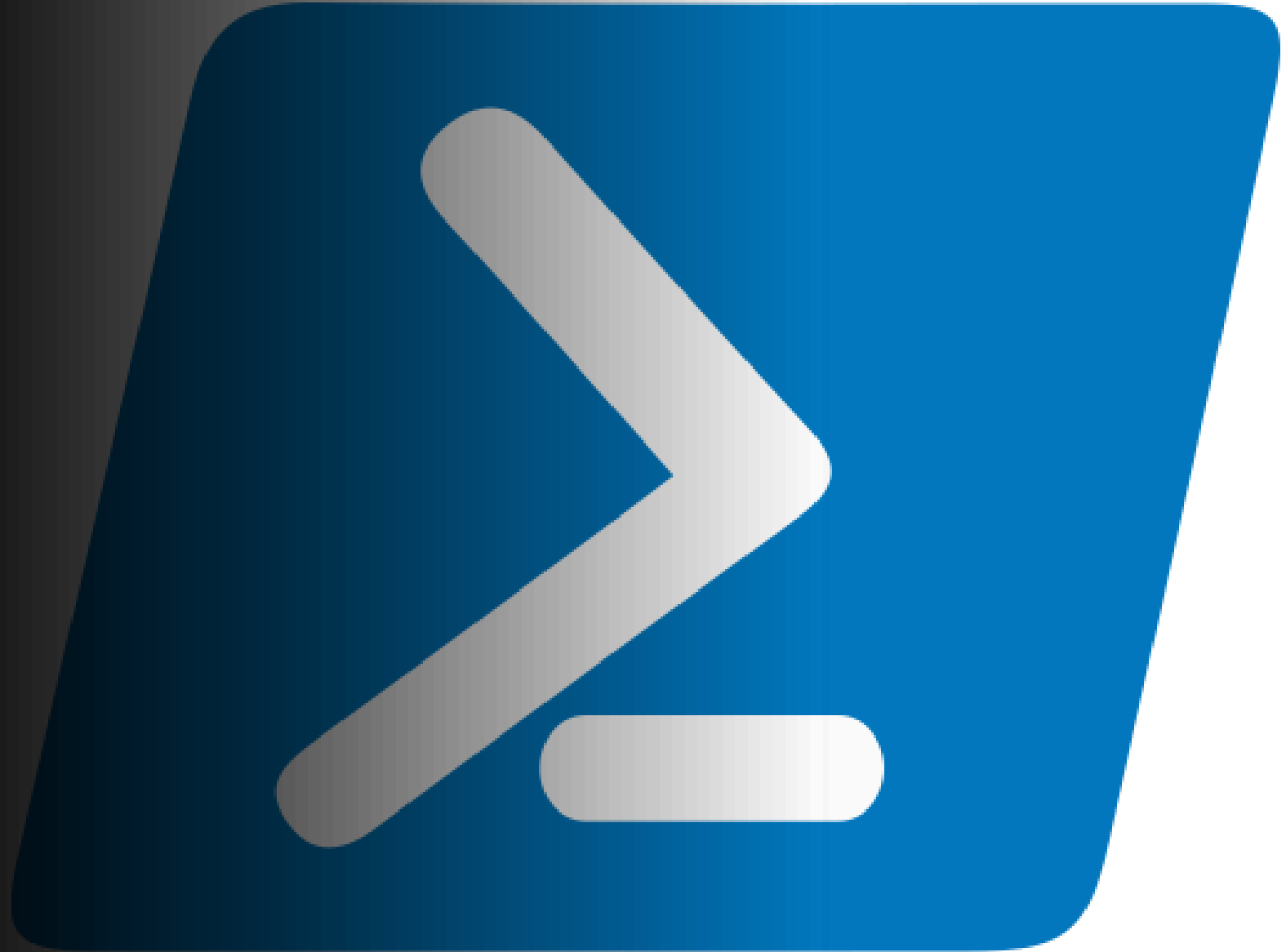
User Interaktion

- Read-Host
- Get-Credential
 - Store Credentials via \$cred | Export-Clixml C:\cred.xml oder via SecretManagement module
- Out-GridView
 - \$Auswahl = \$procs | Out-GridView -PassThru



Lab 04

Erstes Skript



Lab 04

- Erstellt ein Skript mit folgender Funktion:
 - Anlegen einer VM
 - Anfügen einer Disk
 - Konfiguration Netzwerk-Adapter auf vSwitch
 - Anhängen eines DVD Laufwerks
 - Einlegen der ISO
- KEIN HARD CODING → Arbeitet mit Variablen



Parameter & Dokumentation



Parameter

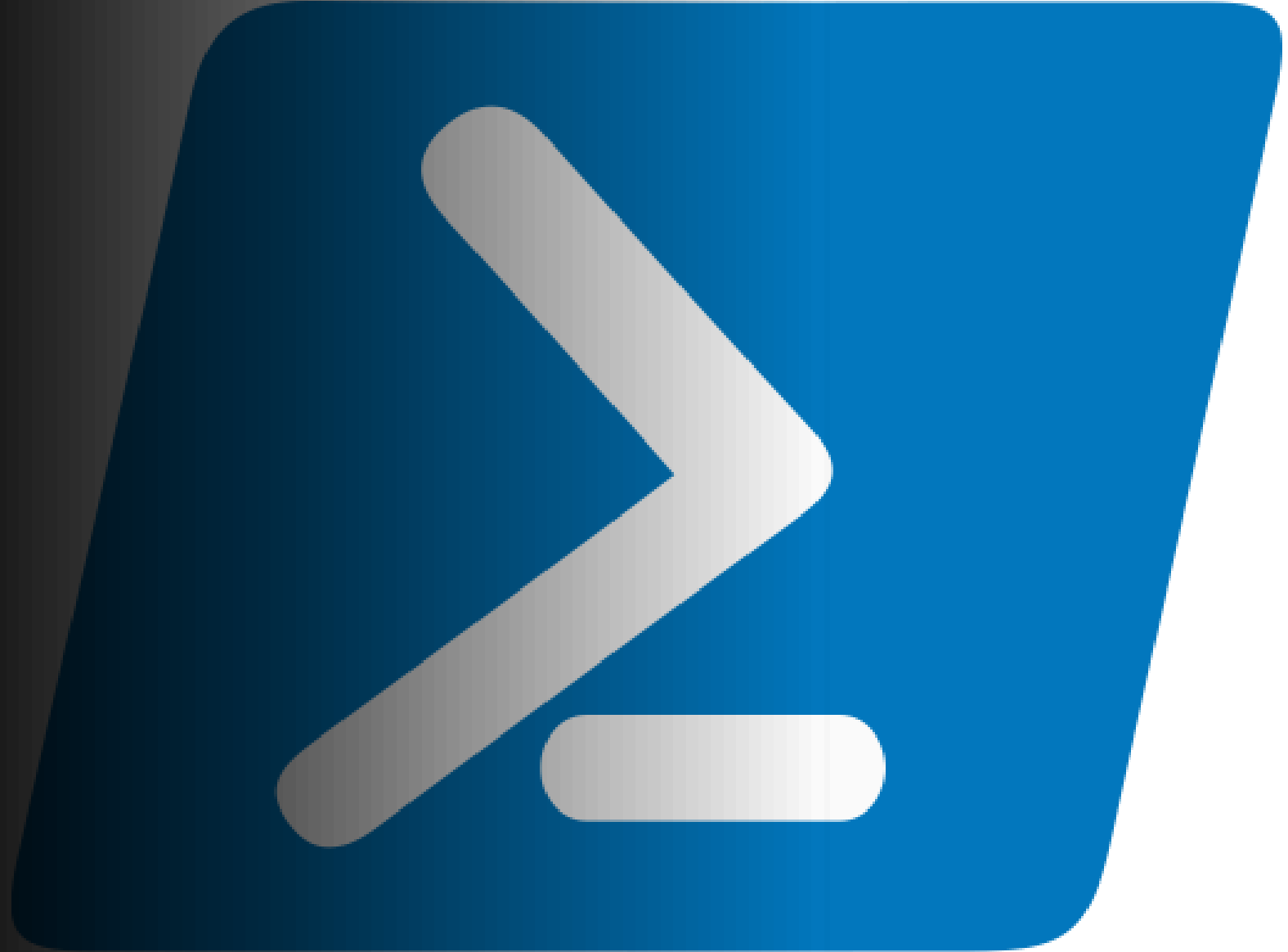
- Anstatt Variablen im Skript anzupassen
- Parameter bei Aufruf des Skripts übergeben
 - Param (
[string]\$VMName
[int]\$DiskSize = 30 #Default Wert
)
- Advanced Param Eigenschaften

Kommentare & Hilfe

- # Einzeilige Kommentare
- <# #> mehrzeilize
- Syntax Description
 - <#
 .Description
 blablabla
 #>



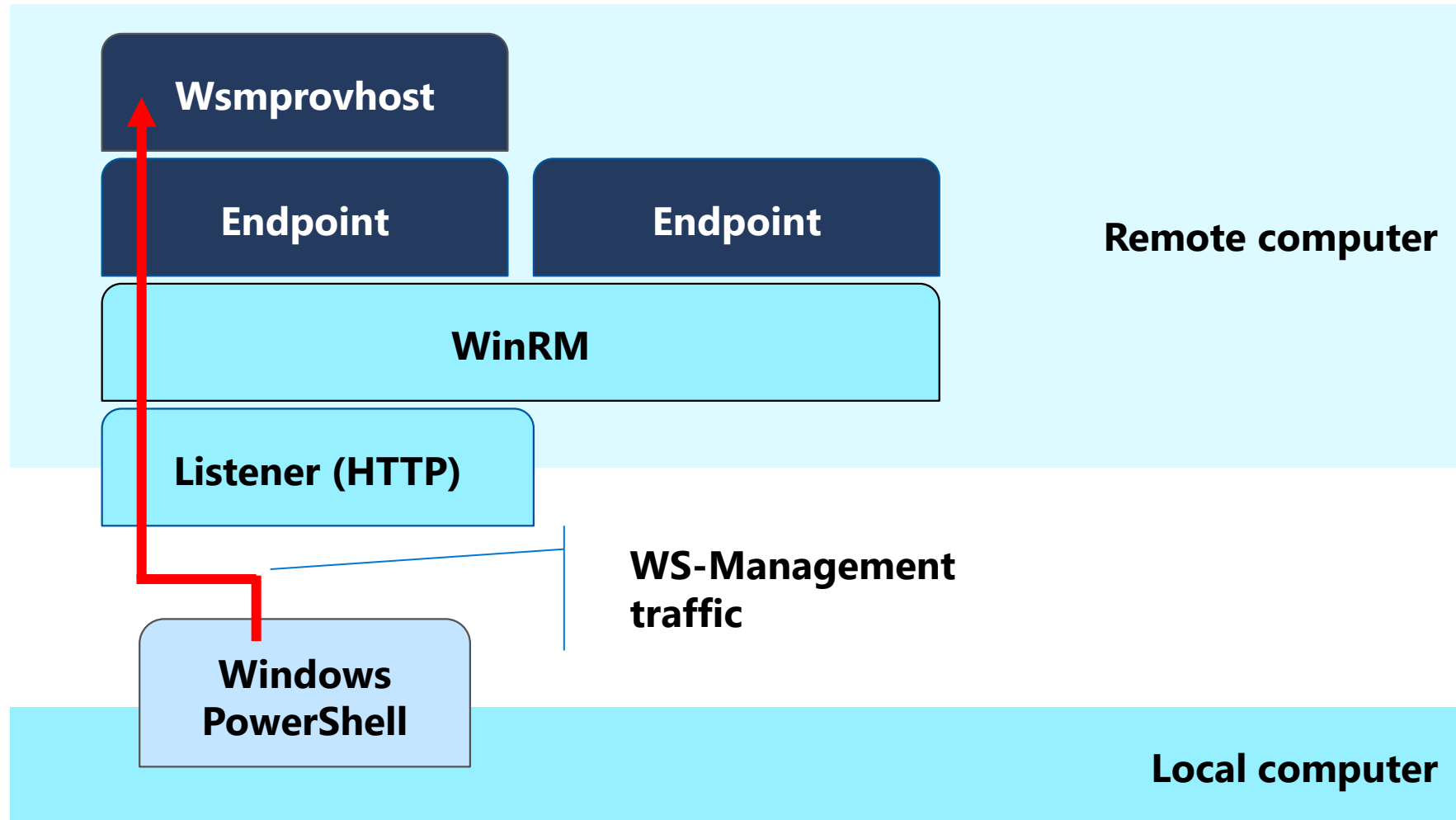
Remoting



Remoting

- Nutzt WS-MAN via HTTP (default) oder HTTPS
- Implementiert im WinRM Service
- Default enabled seit Server 2019 (nicht im Client)
- Auch via SSH für Linux verfügbar
- -ComputerName ist nicht immer „Remoting“

Remoting

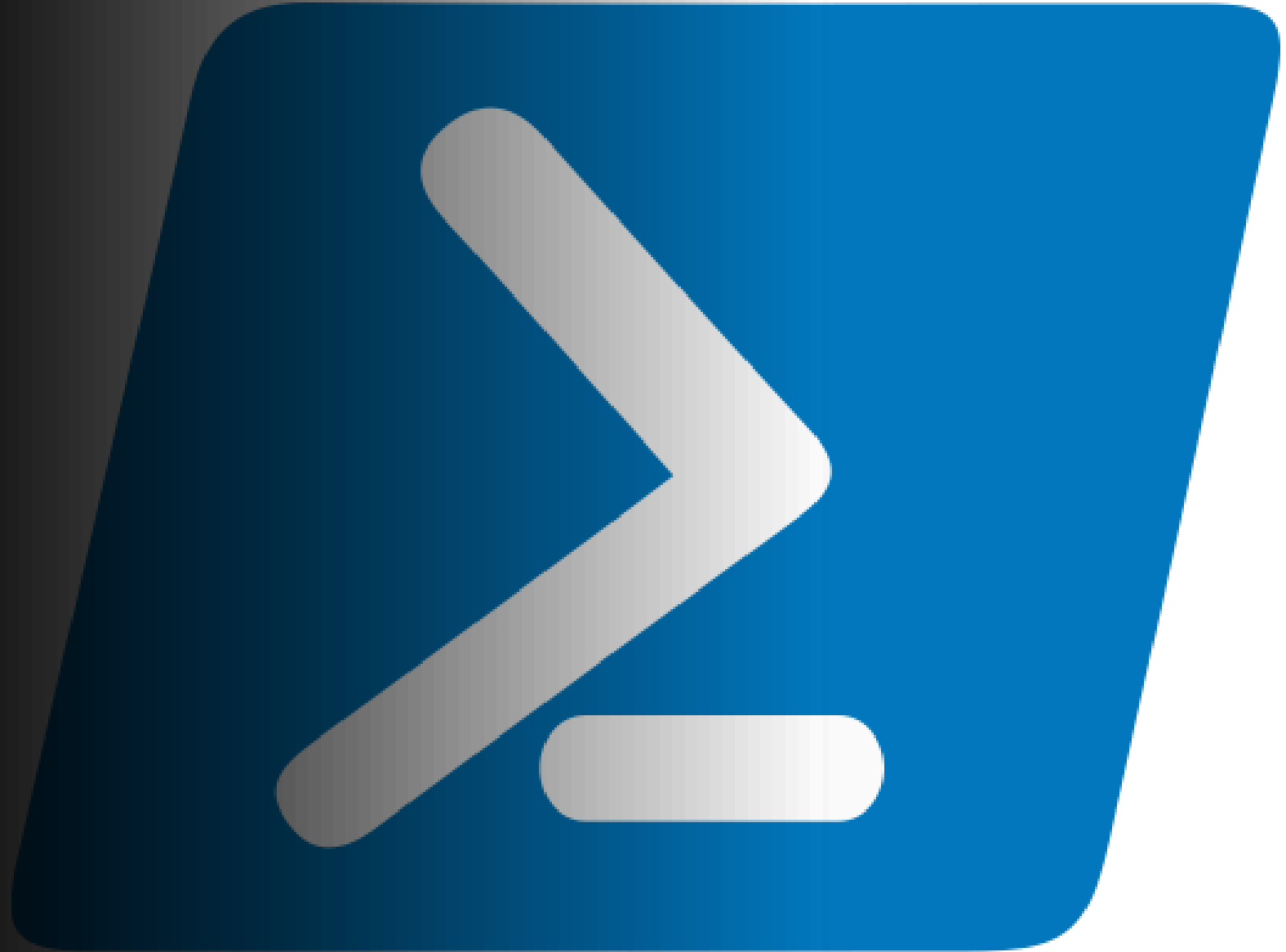


Remoting – Security

- Mutual Authentication
 - Default in Domain Umgebungen
 - Kann SSL
 - **Disabled durch „TrustedHosts“ list!!!!**



Debugging



Skript Output

- Write-Host für Ausgaben während Skript-Ausführung
- Start-Sleep oder Read-Host um Ablauf zu verlangsamen
- Mit CmdletBinding() auch Write-Verbose und Write-Debug
- Ausgabe nur wenn Skript entsprechend aufgerufen wird

Try-Catch

```
try {  
    get-process notepad -ErrorAction Stop  
}  
catch {  
    Write-Host "Der gesuchte Prozess existiert nicht"  
}
```

Error Actions

- Fehler werden in \$Error gespeichert
- Error-Reaktion über -Error-Action
 - Continue
 - SilentlyContinue
 - Inquire
 - Stop



PowerShell Kurs

14. – 16. Oktober 2024

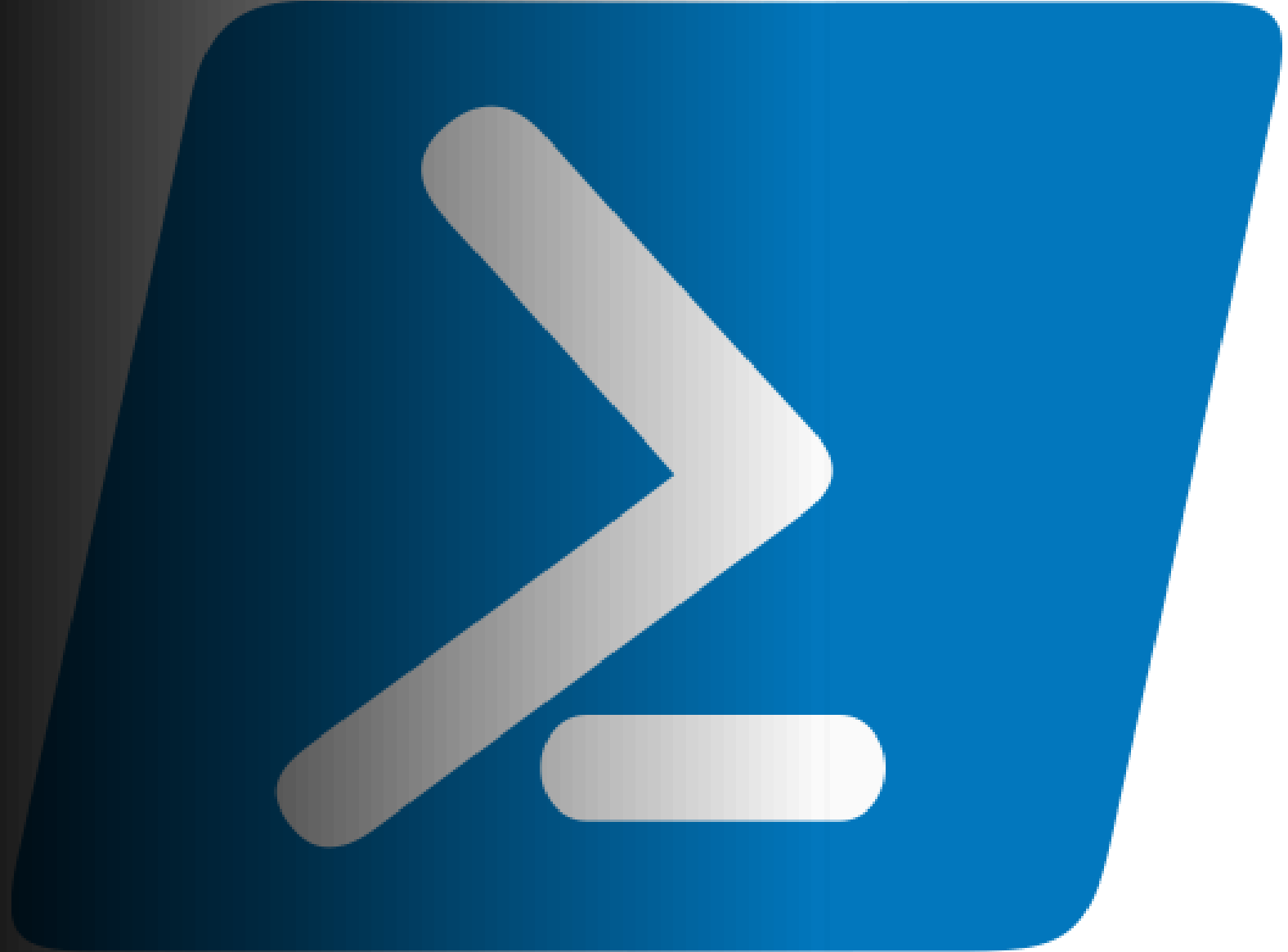


Tag 3 – 16. Oktober

| | |
|---------------|---|
| 09:00 – 09:15 | Recap + Q&A |
| 09:15 – 09:45 | Lab 05 – Parameter, Doku und Debugging |
| 09:45 – 10:00 | <i>PAUSE</i> |
| 10:00 – 10:30 | Azure DevOps Intro |
| 10:30 – 11:30 | Git in Azure DevOps Repos |
| 11:30 – 12:00 | Lab 06 – Repos nutzen |
| 12:00 – 13:00 | <i>MITTAGSPAUSE</i> |
| 13:00 – 13:30 | Azure Cloud Shell |
| 13:30 – 14:15 | GitHub Copilot & AI Nutzung |
| 14:15 – 14:30 | <i>PAUSE</i> |
| 14:30 – 15:30 | Hyper-V & Network ATC |
| 15:30 – 16:00 | Abschluss & Feedback |



Recap + Q&A



Recap

- GET-COMMAND
- GET-HELP
- GET-MEMBER

Recap

- Param ()
- # oder <# #> für Kommentare
- Comment-help für Get-Help Hilfe
- Write-Host, Write-Verbose, Write-Debug
- Try {
 #Aufgabe
} catch {
 #Fehler-Aktion
}



Lab 05

Parameter, Doku, Debugging

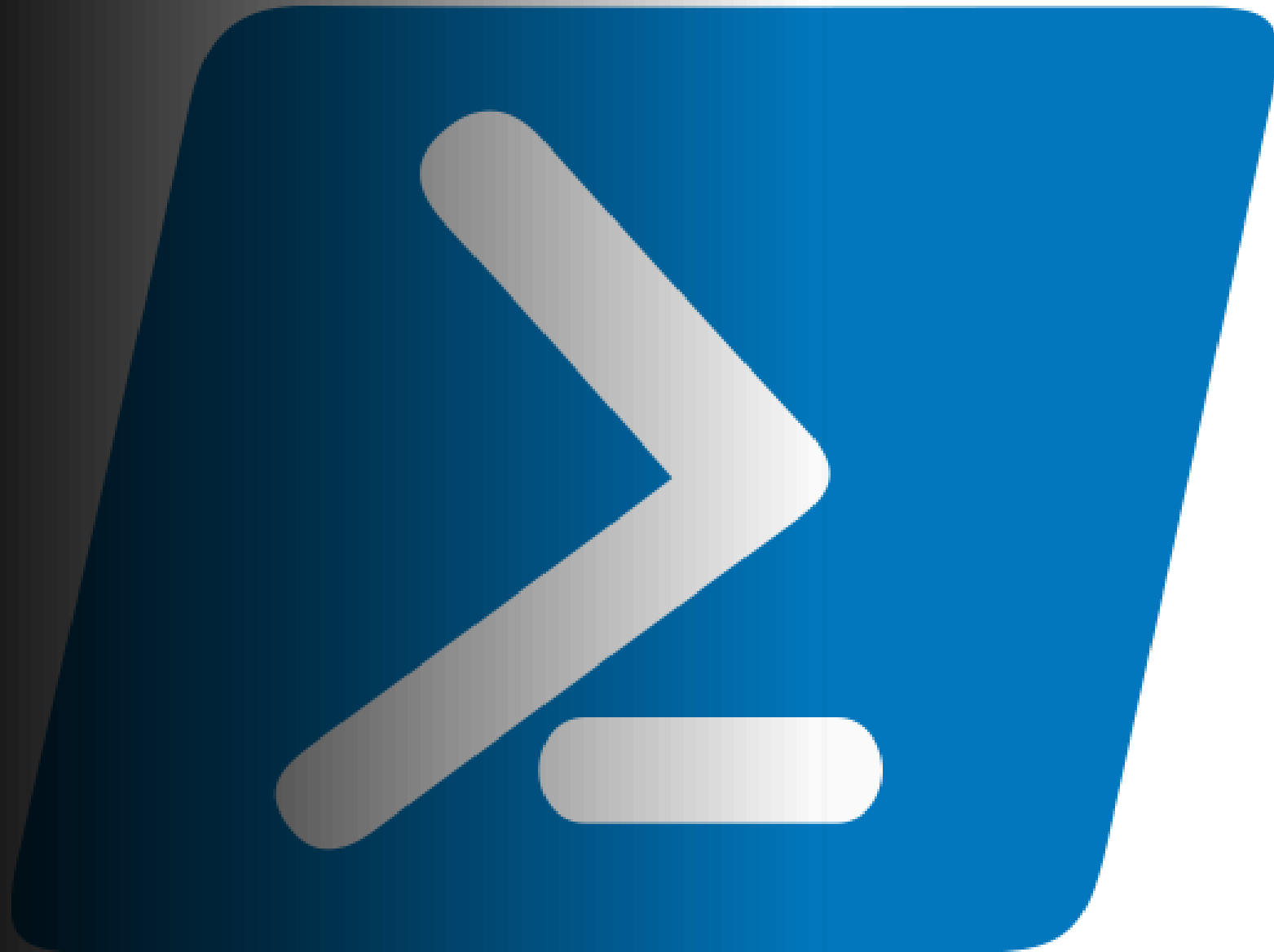


Lab 05

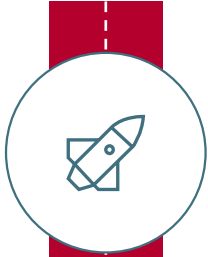
- Modifiziert euer Skript so, dass es mit Parameter für VMName aufgerufen werden kann
- Kommentiert das Skript und erstellt eine (kurze) Doku
- Ergänzt Output für bessere Information der Nutzenden
- Unter \\svHyperV18\C\$\Debug\Script1.ps1 findet ihr ein nicht lauffähiges Skript → kopiert es auf euren Host → bringt es zum Laufen



Azure DevOps Intro



Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

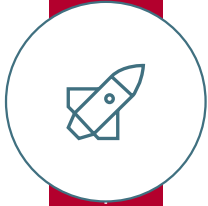


Azure Artifacts



Azure Test Plans

Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

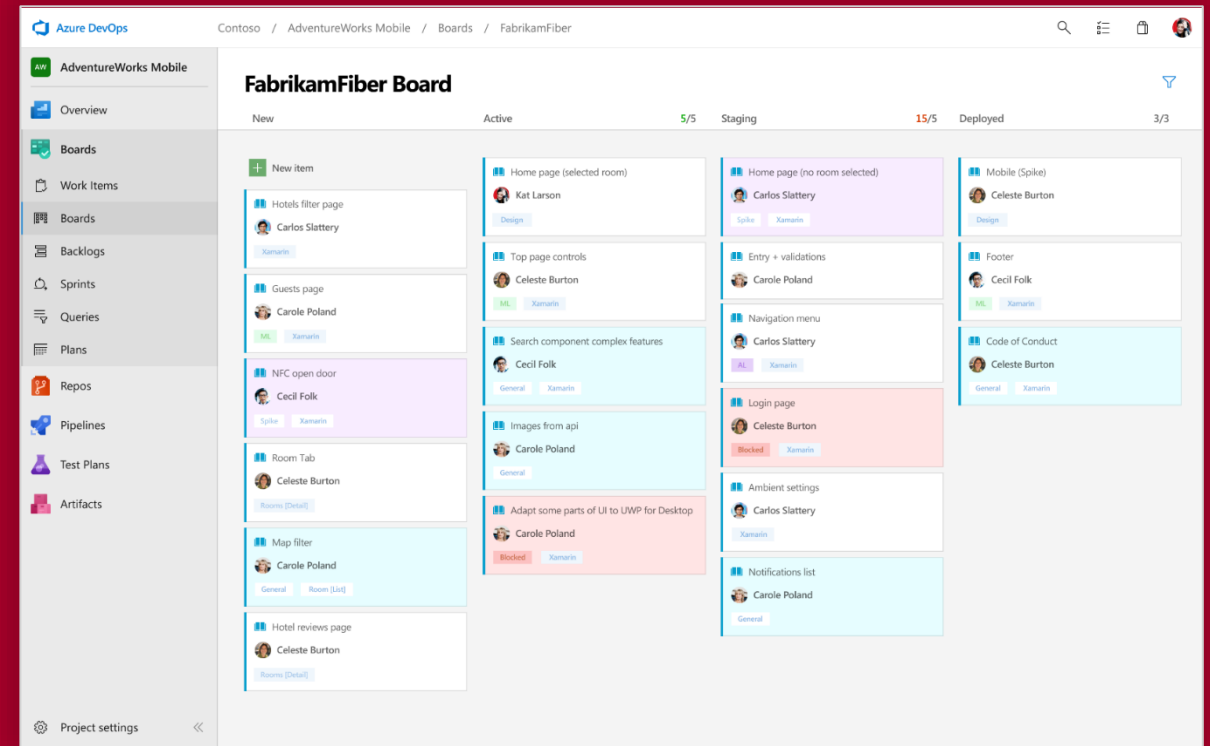


Azure Artifacts



Azure Test Plans

Carst



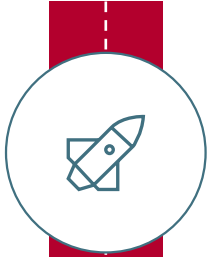
Connecting ideas to releases

Scrum ready to help your teams run sprints, stand-ups, and plan work

Integrated with GitHub commits and pull requests

Insights into project status and health

Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

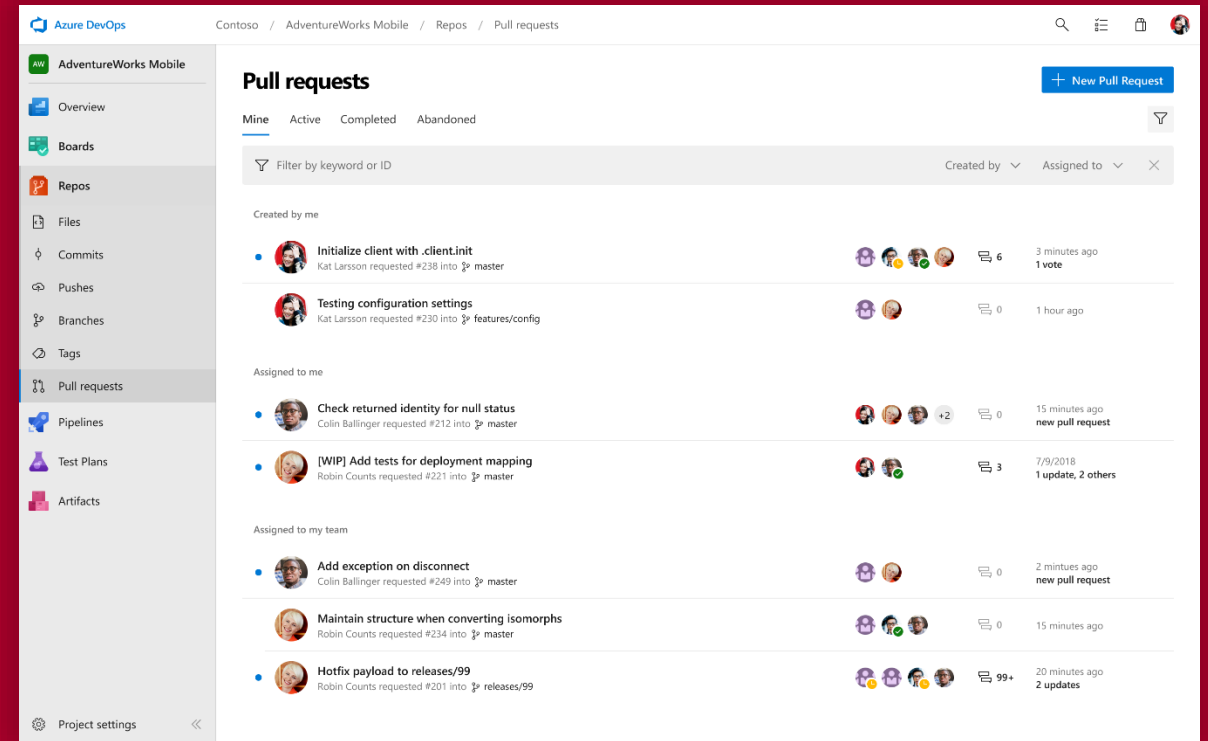


Azure Artifacts



Azure Test Plans

Carst



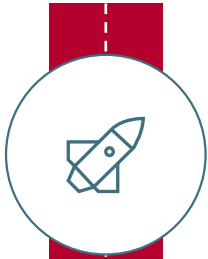
Private Git and TFVC repos for your teams

Code review via branch pull requests

Branch policies and build validation

Easy migration path to / from GitHub

Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines



Azure Artifacts



Azure Test Plans

Carst

Azure DevOps interface showing a pipeline run for 'Enabling feature flags for Preview Attachment and Grid Views'. The pipeline is running on a Linux agent. The summary shows the following jobs and their durations:

| Job Name | Status | Duration |
|-------------|---------|----------|
| Windows Job | Running | 1m 53s |
| Linux Job | Running | 3m 29s |
| macOS Job | Running | 3m 07s |

The Linux Job details show the following steps and their durations:

| Step Name | Duration |
|----------------------|----------|
| Prepare job | <1s |
| Initialize job | 1s |
| Get sources | 24s |
| Cmdline | 28s |
| Nodetool | 3s |
| Install dependencies | 2m 31s |

The terminal output for the Linux Job shows the following commands and their output:

```
yarn install v1.7.0
$ node build/npm/preinstall.js
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
$ npm run compile
[#####] 152/243
> code-oss-dev-build@1.0.0 compile ./adventureworks/build
> tsc -p tsconfig.build.json
Done in 4.89s.
$ node ./postinstall
[##] 2/2removed './adventureworks/extensions/node_modules/typescript/lib/tsc.js'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/tsserverlibrary.js'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.d.ts'
removed './adventureworks/extensions/node_modules/typescript/lib/typescriptServices.js'
```

Cloud-hosted pipelines for Linux, macOS and Windows

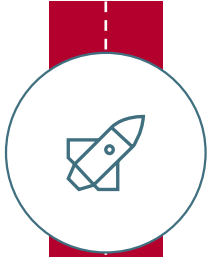
Any language, any platform, any cloud

Native support for containers and Kubernetes

Best-in-class for open source



Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

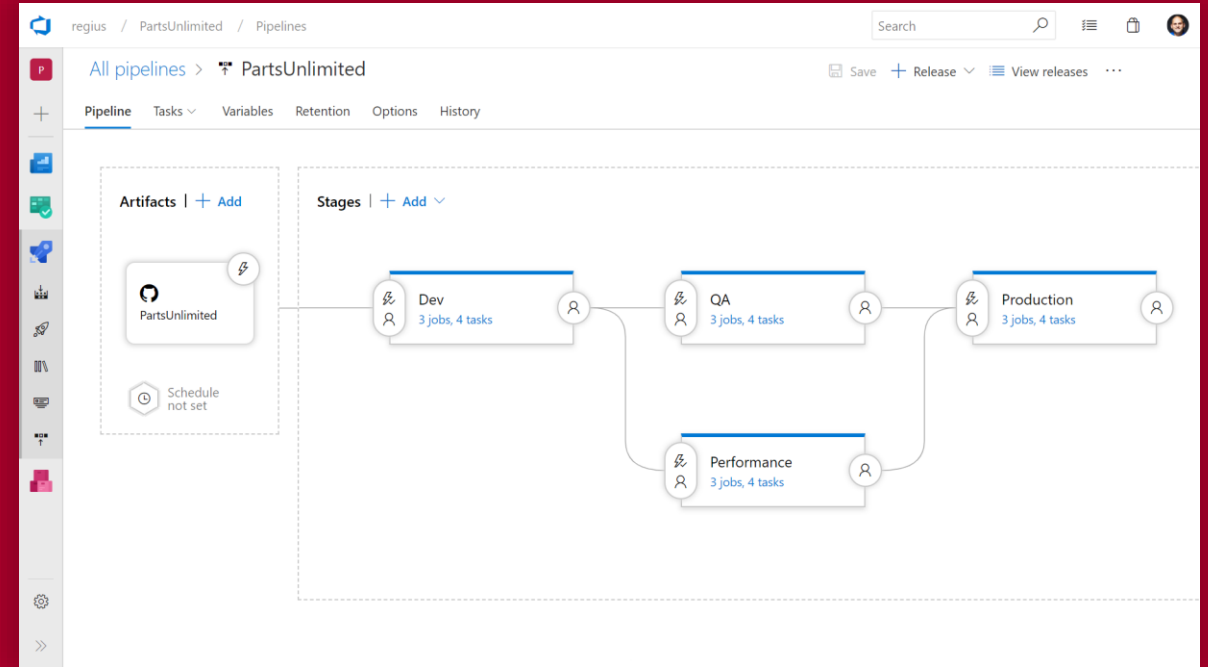


Azure Artifacts



Azure Test Plans

Carst



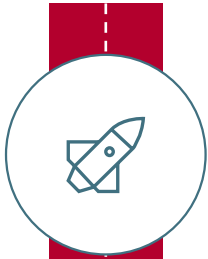
Deploy to on-premises, ANY cloud or a hybrid of cloud and on-prem

Staged environment releases

Pre and post deployment approvals with gates to automate approval based on conditions



Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

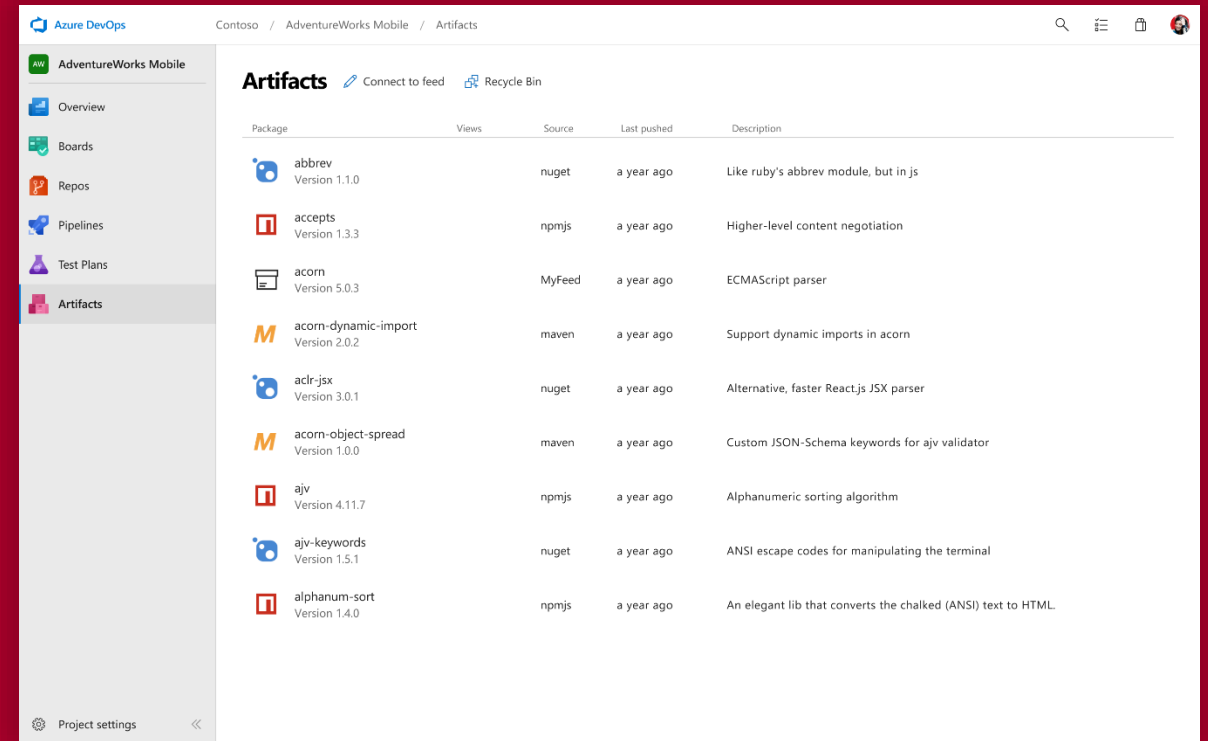


Azure Artifacts



Azure Test Plans

Carst



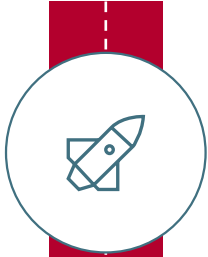
Share code efficiently

Keep your Maven, npm, NuGet and Python packages and more in the same place

Aggregate from public registries and internal teams

Publish and track from any pipeline

Azure DevOps



Azure Boards



Azure Repos



Azure Pipelines

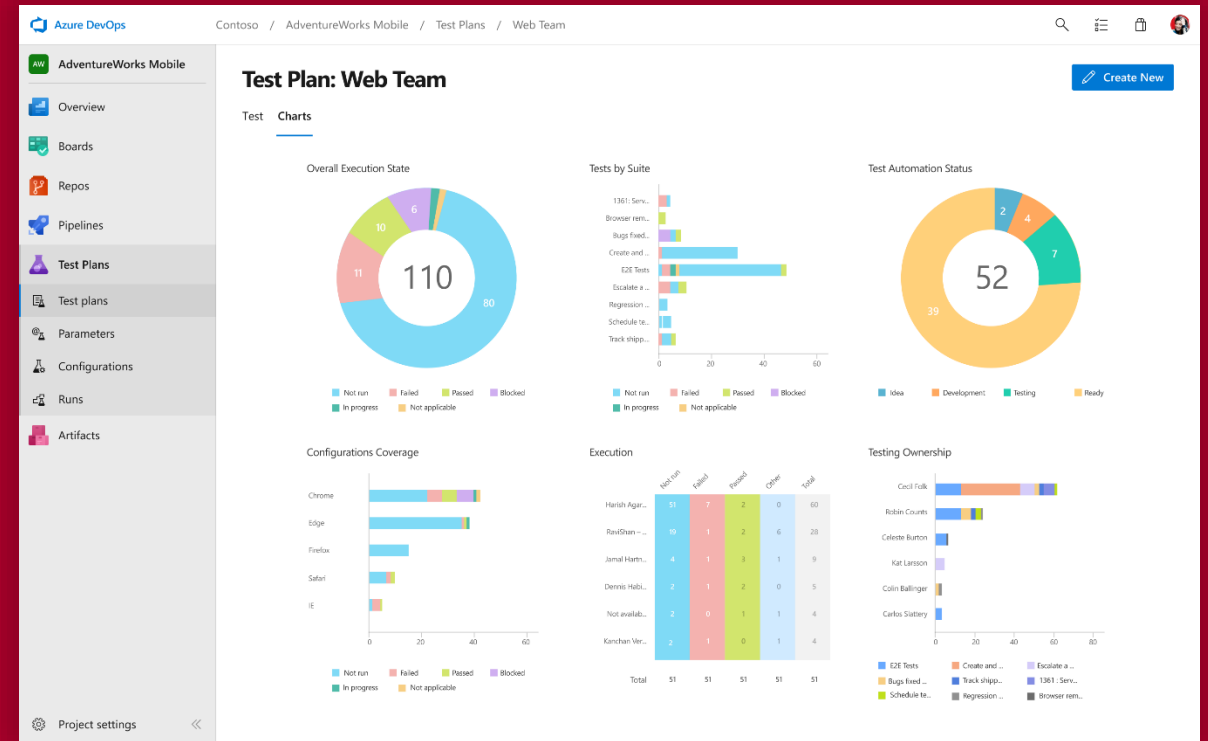


Azure Artifacts



Azure Test Plans

Carst



Run tests and log defects from your browser

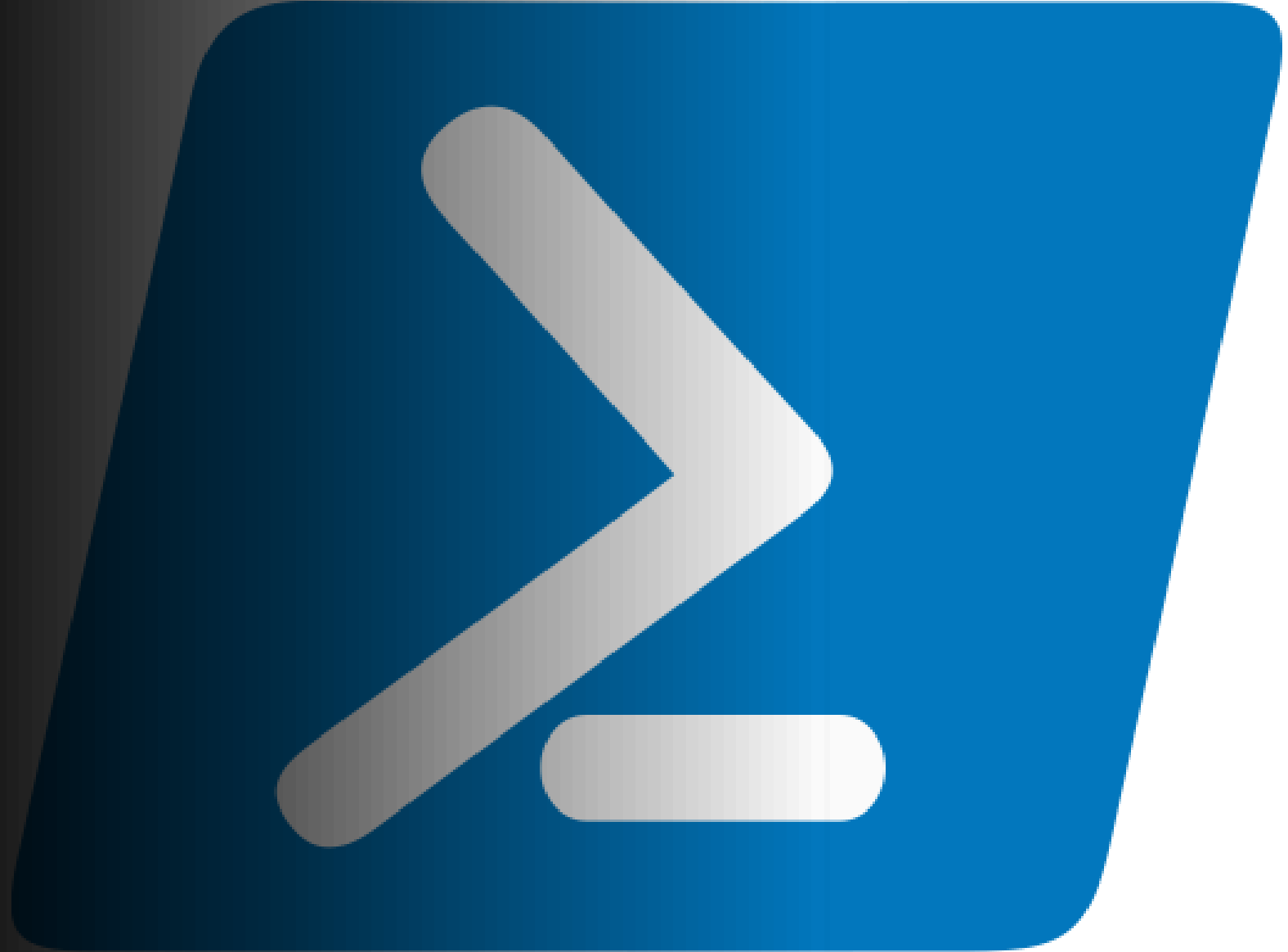
Track and assess quality throughout your lifecycle

Capture rich data for reproducibility

Create tests directly from exploratory sessions



Git in Repos



Git

- Versions-Kontroll-System
- Cross-Platform
- Open-Source
- De-Facto Standard
- GitHub ist eine web-basierte Git Hosting Plattform ... Alternative: Azure DevOps, GitLab, Bitbucket, etc.

Git - Befehle

- `git init`
- `git clone <repo>`
- `git add script.ps1`
- `git commit -m „Info Details“`
- `git push & pull`
- u.v.m → oder ihr nehmt die GUI :-)



Lab 06

Repos nutzen



Lab 06

- Installiert Git in eurer Umgebung <https://git-scm.com/downloads>
- Anmeldung am Azure Portal <https://portal.azure.com>
- Neue Azure DevOps Organisation einrichten
- Repo erstellen und clonen
- Skript und Änderungen synchronisieren



Feedback



Darf's noch etwas mehr sein?

- Learning Accelerator Network for Microsoft Solutions
 - mlan-germany.de
- Windows Server Launch Tour - 04.-14.11.2024:
 - [Windows Server 2025 Launch Tour](#) (MUC, Sindelfingen, FRA, HAM, LEJ, DUS)
- Azure Hybrid Cloud Workshop - 27.-29.01.2025:
 - [Azure Hybrid Workshop](#) (online)
- Windows Server Summit 2025 - 09.-10.04.2025 :
 - windows-server-summit.de (Leipzig)
- Experts Live Germany 2025 - 11.04.2025 :
 - expertslive.de (Leipzig)
- Mehr PowerKurse: powerkurs.net

Feedback

- PowerShell Skill 1 – 6 (Schulnoten)
- Windows Server Skill 1 – 6 (Schulnoten)
- Erwartung erfüllt?
- Hat sich der Kurs für dich gelohnt?
- Offenes Feedback ...



Done

