

# PowerShell - What's new?

Black is the new blue...



# Haiko Hertes

- 86er, Tech-Addict, Hobbyfotograf und Motorradfahrer
- Seit 01.04. bei SoftwareONE (ex. COMPAREX)
- Vorher IT-Leiter im Mittelstand und IT-Trainer
- Schwerpunkte: Azure, PowerShell, Hyper-V, System Center, AD
- Seit 2016 Microsoft MVP (CDM)
- Betreibe u.a. auch die Windows Server Usergroup Leipzig



[www.hertes.net](http://www.hertes.net)



[about.me/haiko.hertes](mailto:about.me/haiko.hertes)



[twitter.com/HHertes](https://twitter.com/HHertes)



Auch auf YouTube...

softwareONE



**Microsoft®**  
Most Valuable  
Professional



**HERTES.NET**  
knowledge - experience - insights



# AGENDA

**01/** PAST – PRESENT – FUTURE

---

**02/** FEATURES

---

**03/** SUPPORT

---

**04/** HOW ABOUT CLOUD?

---



# PAST – PRESENT – FUTURE

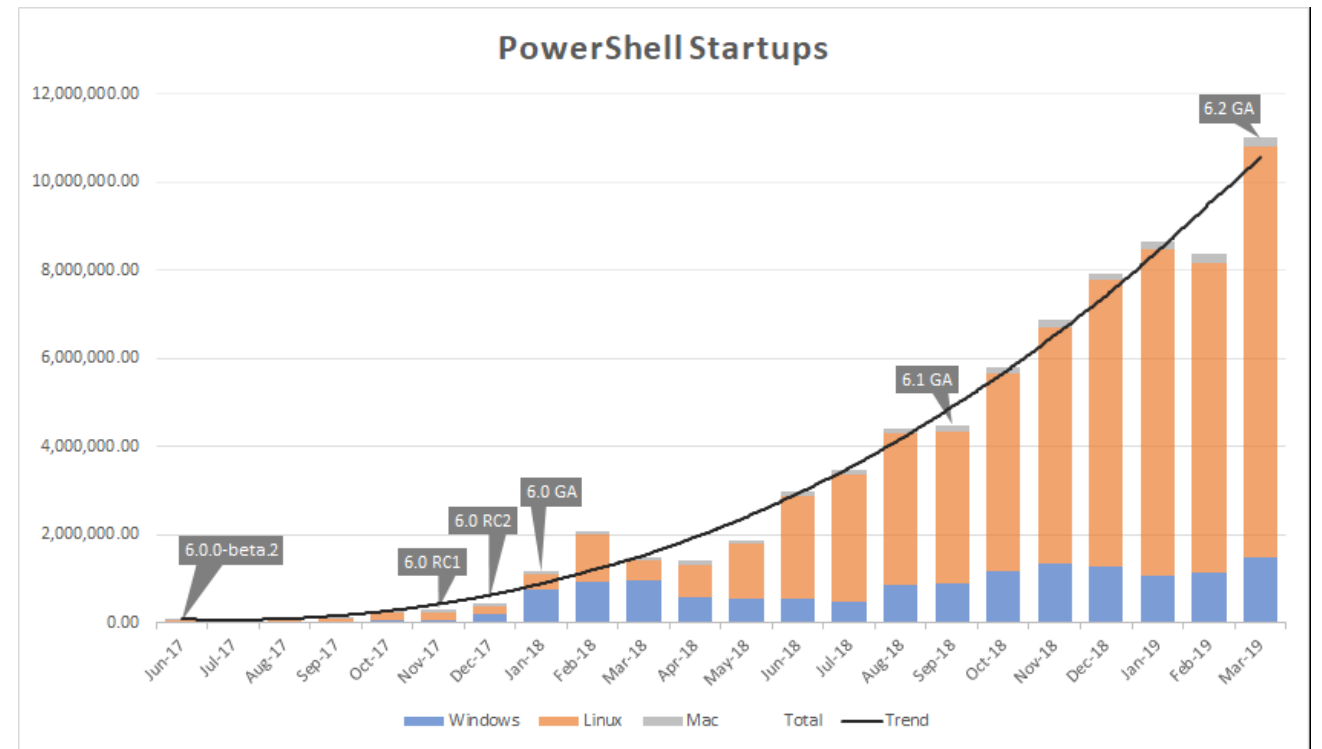
Black is the new Blue...

# Vergangenheit

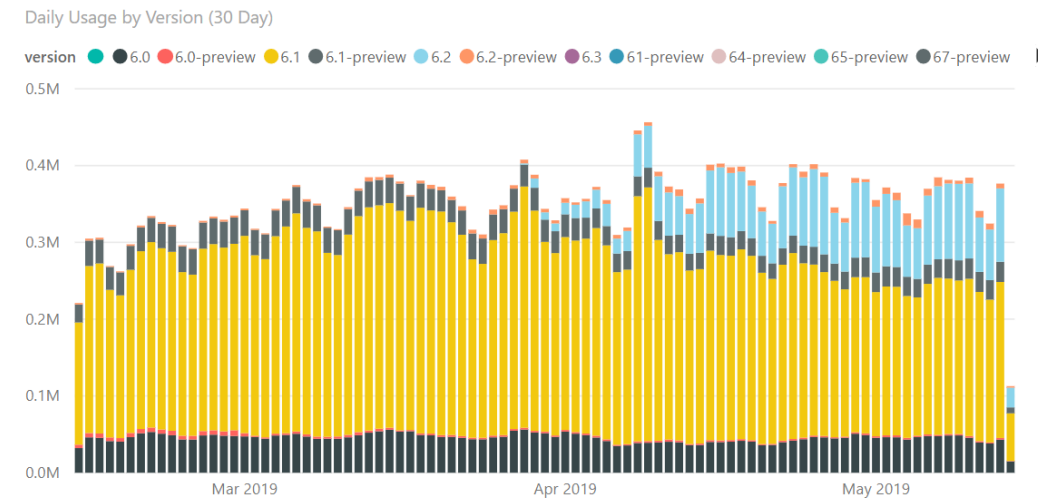
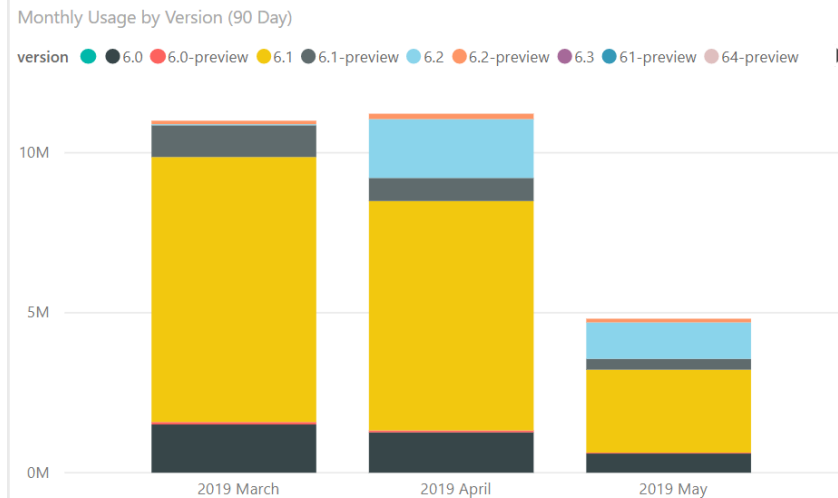
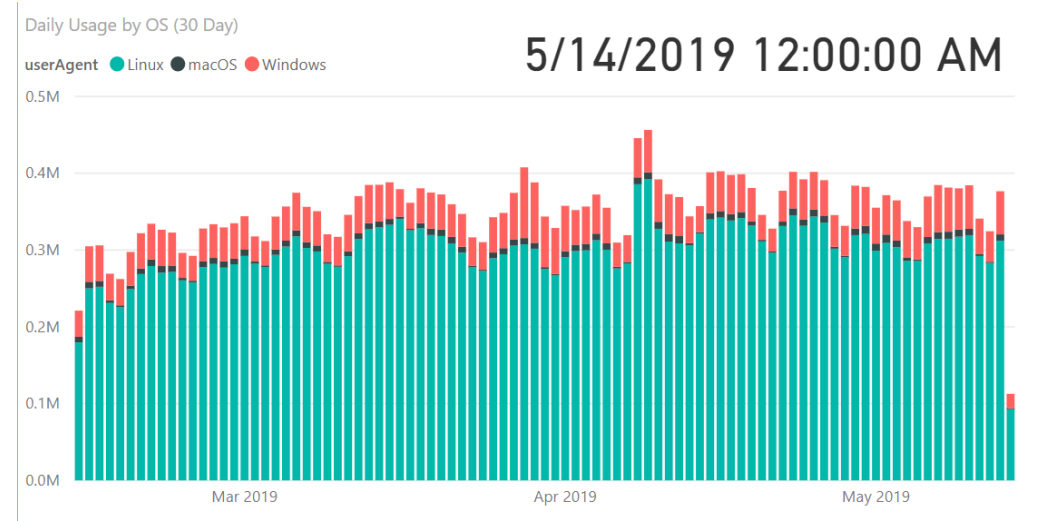
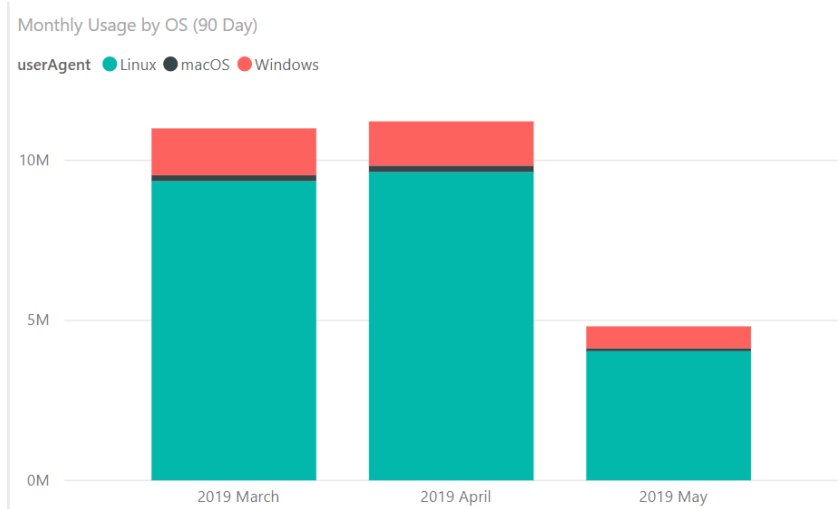
- „Windows PowerShell“ wurde als „**Monad**“ entwickelt
- 2006 in „**PowerShell**“ umbenannt und als Version 1.0 veröffentlicht
- PowerShell wird fester Bestandteil aller Windows- und Windows-Server-Versionen
- Veröffentlichung als Teil des „**Windows Management Framework**“, darüber auch Upgrade älterer OS
- Seit 2009 unterstützt nahezu **jedes Microsoft Datacenter-Produkt** PowerShell
- 2016 wird PowerShell unter MIT-Lizenz zum **Open-Source-Projekt**
- PowerShell 2.0 ist seit 2017 (10 Fall Creators Update) deprecated
- Seit 2018: **PowerShell Core**

# Gegenwart

- „Windows PowerShell“ gilt als Feature-complete
- Es erfolgt keine weitere Feature-Entwicklung
- Sicherheitslücken und Blocker für Geschäftskunden werden voraussichtlich noch behoben
- „Windows PowerShell“ bleibt (vorerst) weiter supported
- Aktueller Schwerpunkt ist „PowerShell Core“
- Nutzung auf non-Win-OS wächst stärker



# Gegenwart



# Zukunft

- Die nächste Version von PowerShell Core wird „PowerShell 7“ sein
- das „Core“ wird aus dem offiziellen Namen entfernt (bleibt aber intern)
- Basis ist dann .Net (Core) 3.0
- Erste Preview für Mai 2019 erwartet, finales Release für Q4/2019 (nach Release von .Net 3.0)
- Vermutlich: PS Core in künftigen Windows/WindowsServer Releases enthalten (ggf. parallel zur „legacy“ PS)
- Vorerst side-by-side mit legacy PowerShell
- Ziel: PowerShell (Core) läuft überall (u.a. auch Multi-Cloud-Szenario) und verwaltet (fast) alles



# PowerShell 7

- Basis ist .Net (Core) 3.0
- Kompatibilität zu PowerShell 5 soll weitgehend sichergestellt sein
- Erste Preview noch für diese Woche erwartet
- Ziel: PowerShell 5.1 ablösen
- Keine „significant breaking changes“ erwartet
- Schwerpunkt: Kompatibilität zu älteren Modulen und Skripten
  - Angestrebt sind 90%!

# PowerShell Versionen

Name / Version	Veröffentlicht	Basiert / benötigt	Ausgeliefert in...	Nachrüstbar in...
Windows PowerShell 1.0	2006	.Net 2.0	Gesonderter Download	WS2003, WXP
Windows PowerShell 2.0	2009	.Net 2.0	WS2008R2	WS2003, WS2008, WXP, W7
Windows PowerShell 3.0	2012	.Net 4.0	WS2012, W8	WS2008, WS2008R2, W7
Windows PowerShell 4.0	2013	.Net 4.5	WS2012R2, W8.1	WS2008R2, W2012, W7
<del>Windows PowerShell 5.0</del>	<del>2014</del>	<del>.Net 4.5</del>	<del>W10</del>	<del>WS2008R, WS2012, WS2012R2, W7, W8.1</del>
Windows PowerShell 5.1	2016	.Net 4.5	WS2016, WS2019, W10	WS2008R, WS2012, WS2012R2, W7, W8.1
PowerShell Core 6.0	01/2018	.Net Core 2.0	-	
PowerShell Core 6.1	09/2018	.Net Core 2.1	-	WS2008R2-WS2019, W7-W10, Linux, macOS
PowerShell Core 6.2	03/2019	.Net Core 2.1	-	WS2008R2-WS2019, W7-W10, Linux, macOS, ARM
PowerShell 7	05/2019 (?)	.Net Core 3.0	-	?

# PowerShell, Windows PowerShell, PowerShell Core ?!?

- Die ersten Versionen von PS hießen „Windows PowerShell“
- Diese wurden aber oft nur „PowerShell“ genannt
- Mit der Einführung von „PowerShell Core“ wurde die Abgrenzung zu „Windows PowerShell“ wieder wichtig(er)
- Ab PowerShell 7 wird das „Core“ wieder entfallen, die „Windows PowerShell“ wird nicht weiterentwickelt
- powershell.exe wurde mit PS 6.0 zu pwsh.exe

# Die IDE



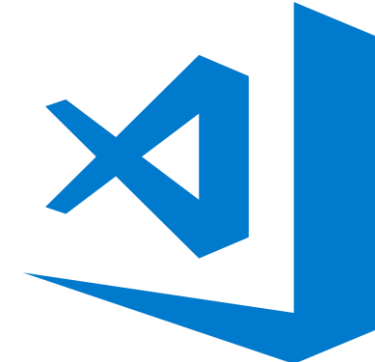
Windows PowerShell  
(Legacy)



PowerShell ISE



PowerShell (Core)



Visual Studio Code



# WindowsCompatibility Module

- Einige bekannte Cmdlets laufen nicht in PS Core
  - Bspw. Get-EventLog, (Get-NetAdapter, Get-LocalUser ...)

```
PowerShell 6 (x64)
PowerShell 6.2.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS C:\Users\Haiko.Hertes> Get-Eventlog
Get-Eventlog : The term 'Get-Eventlog' is not recognized as the name of a cmdlet, function, script file, or executable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and that the file exists.
At line:1 char:1
+ Get-Eventlog
```

Nutzt  
PSRemoting...

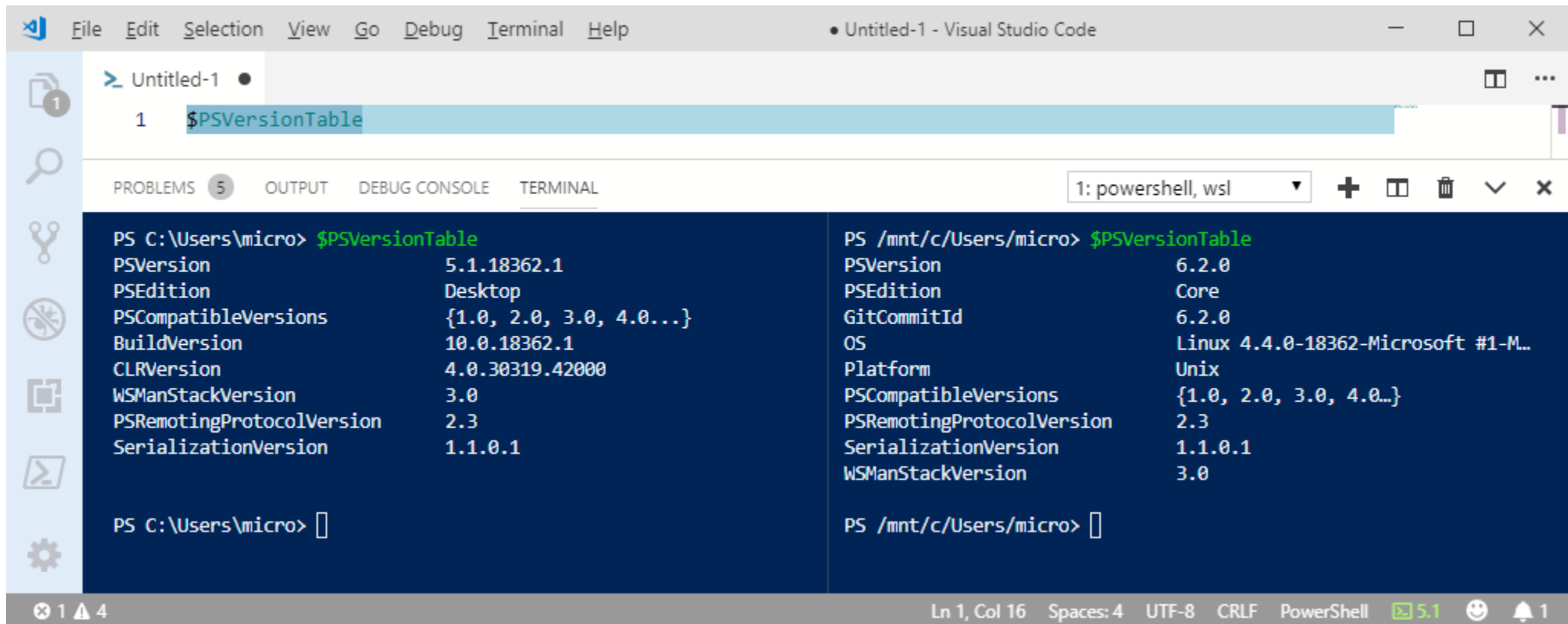
- Abhilfe unter Windows: WindowsCompatibility
- Install-Module WindowsCompatibility
- Invoke-WinCommand
- Import-WinModule

```
PowerShell 6 (x64)
PS C:\Users\Haiko.Hertes> Install-Module WindowsCompatibility

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the
modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
PS C:\Users\Haiko.Hertes> Invoke-WinCommand {Get-Eventlog}
```

# Side by Side

- VS Code erlaubt „Split Terminal“
- Dadurch nebeneinander Ausführen von Code in verschiedenen PS Versionen möglich



The screenshot shows the Visual Studio Code interface with a split terminal. The top menu bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The title bar indicates 'Untitled-1 - Visual Studio Code'. The left sidebar shows the Explorer, Search, and Run and Debug views. The main editor area displays a file named 'Untitled-1' with the command '\$PSVersionTable' on line 1. The terminal panel at the bottom is split into two panes, each running a PowerShell session. The left pane shows the output of '\$PSVersionTable' for a Windows PowerShell session (5.1.18362.1). The right pane shows the output for a WSL PowerShell session (6.2.0). The status bar at the bottom indicates 'Ln 1, Col 16', 'Spaces: 4', 'UTF-8', 'CRLF', 'PowerShell', and '5.1'.

```
File Edit Selection View Go Debug Terminal Help • Untitled-1 - Visual Studio Code
```

```
1 $PSVersionTable
```

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 1: powershell, wsl
```

```
PS C:\Users\micro> $PSVersionTable
PSVersion          5.1.18362.1
PSEdition           Desktop
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
BuildVersion        10.0.18362.1
CLRVersion           4.0.30319.42000
WSManStackVersion    3.0
PSRemotingProtocolVersion 2.3
SerializationVersion 1.1.0.1

PS C:\Users\micro>
```

```
PS /mnt/c/Users/micro> $PSVersionTable
PSVersion          6.2.0
PSEdition           Core
GitCommitId         6.2.0
OS                  Linux 4.4.0-18362-Microsoft #1-M..
Platform            Unix
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion 2.3
SerializationVersion 1.1.0.1
WSManStackVersion    3.0

PS /mnt/c/Users/micro>
```

```
Ln 1, Col 16 Spaces: 4 UTF-8 CRLF PowerShell 5.1
```

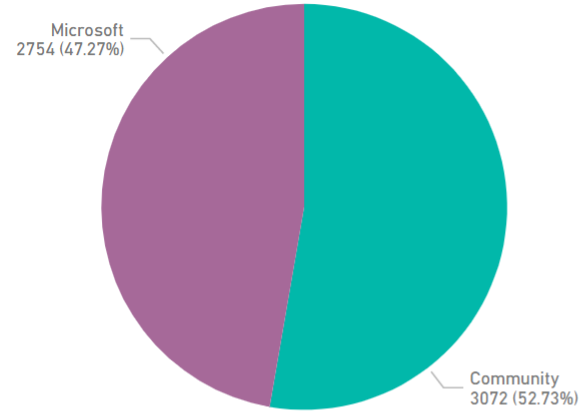
A background image of a modern building with a curved glass facade, partially obscured by a dark overlay and a red diagonal shape in the bottom right corner.

# Demo

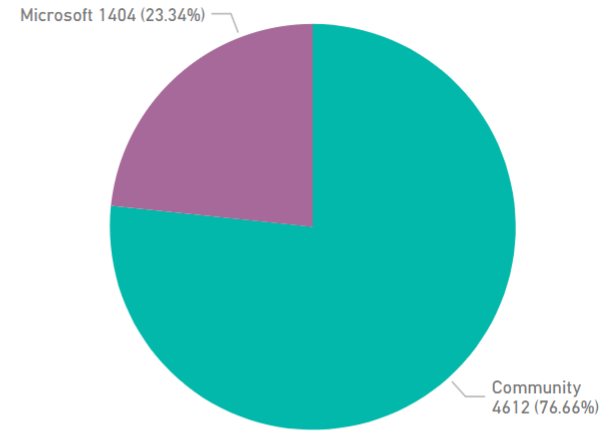
## Visual Studio Code und PowerShell Core

# Community

Pull Requests Created

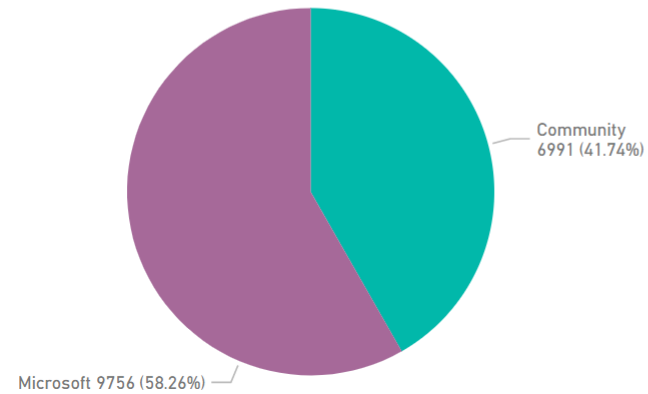


Issues Created

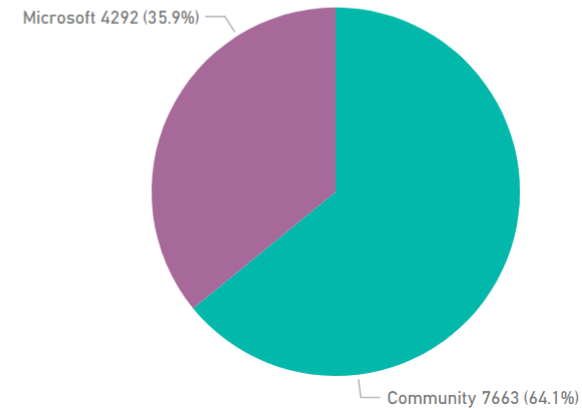


PowerShell/PowerShell repo

Pull Request Comments



Issue Comments



[Aka.ms/PSGitHubBI](https://aka.ms/PSGitHubBI)



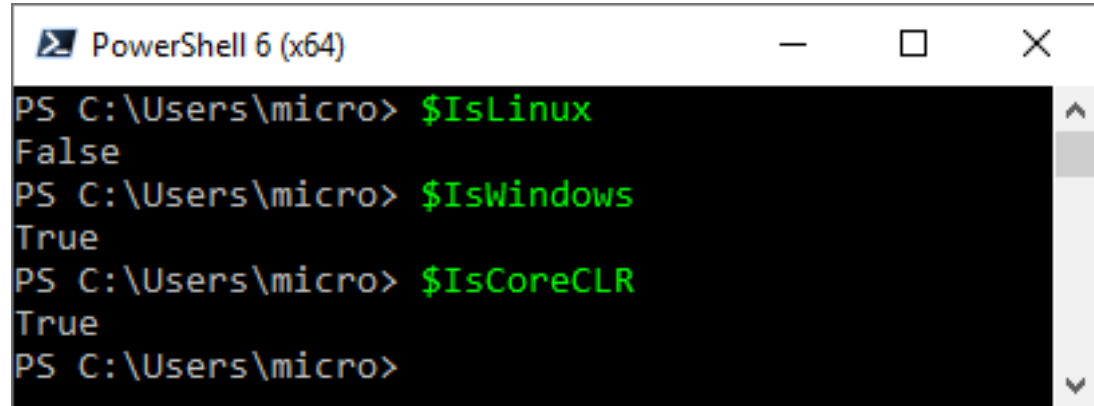
# FEATURES

# pwsh als native Shell

- „.\pwsh -i“ öffnet PowerShell interaktiv
- Dadurch Verwendung als native Shell möglich

# Automatic variables

- Einige neue Automatic Variables
  - \$IsCoreCLR
  - \$IsWindows
  - \$IsLinux
  - \$IsMac
- Erleichtern es, Skripte zu schreiben, die „überall“ funktionieren



```
PowerShell 6 (x64)
PS C:\Users\micro> $IsLinux
False
PS C:\Users\micro> $IsWindows
True
PS C:\Users\micro> $IsCoreCLR
True
PS C:\Users\micro>
```

# CTRL+Space

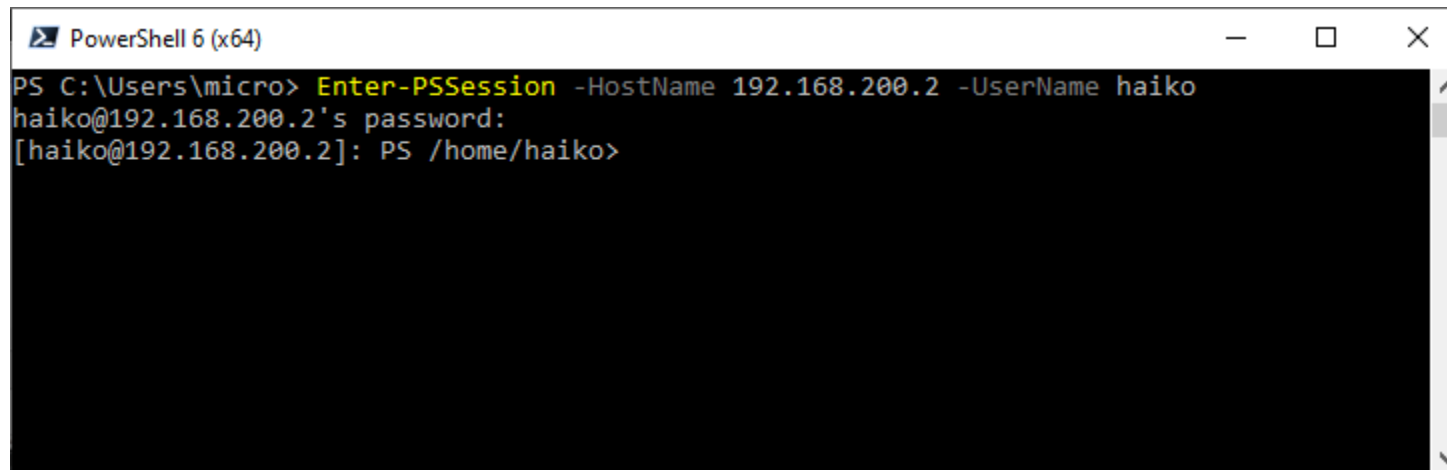
- [CTRL]+[Space]
- Vorschläge für mögliche Vervollständigung von Cmdlets, Parametern und co.





# SSH-based Remoting

- Neben WinRM auch SSH für PowerShell Remoting möglich
- Dadurch PSRemoting gegen alle Plattformen möglich
- Unterstützt sowohl Passwort-basierte als auch Zertifikat-basierte Authentifizierung



```
PowerShell 6 (x64)
PS C:\Users\micro> Enter-PSSession -HostName 192.168.200.2 -UserName haiko
haiko@192.168.200.2's password:
[haiko@192.168.200.2]: PS /home/haiko>
```

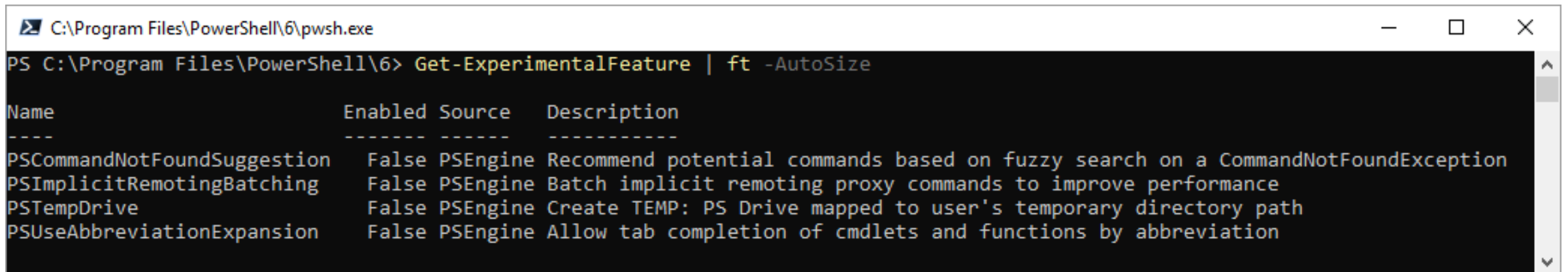
A modern building with a glass facade and a courtyard. The image is split into two parts: a dark, low-key view of the building's exterior on the left, and a bright, high-key view of the building's interior or courtyard on the right. The right side is partially obscured by a red geometric overlay.

# Demo

## Features I

# Experimental Features

- Seit dem Release von PowerShell Core 6.1 werden „Experimental Features“ unterstützt (auf Grund User-Request!)
- Sind per default deaktiviert
- Müssen erst mit `Enable-ExperimentalFeature` aktiviert werden (+Neustart!)

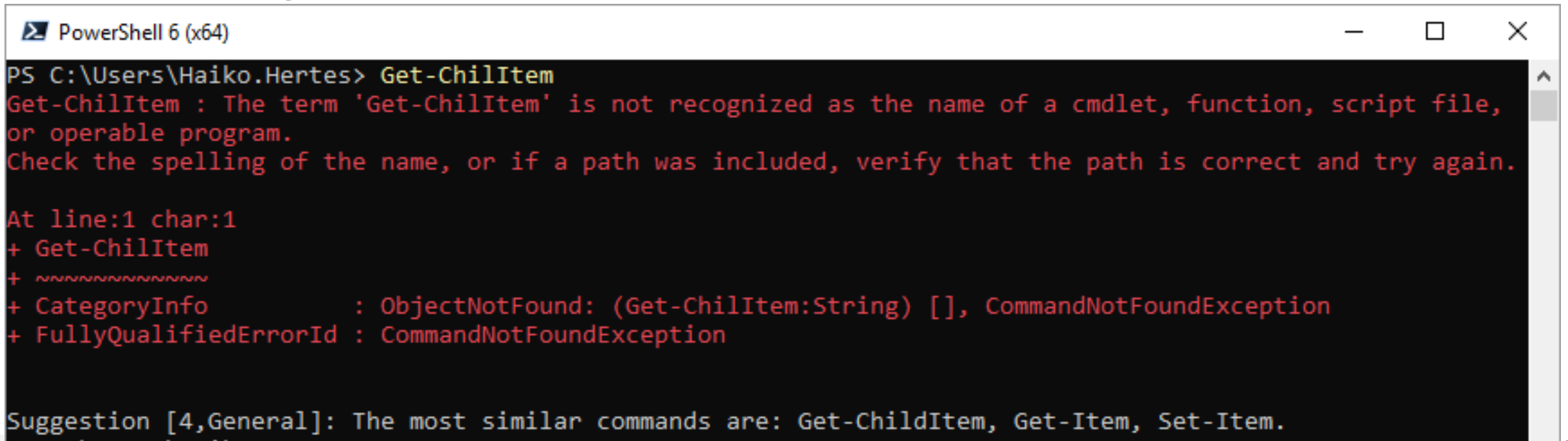


The screenshot shows a PowerShell console window titled "C:\Program Files\PowerShell\6\pwsh.exe". The command `Get-ExperimentalFeature | ft -AutoSize` has been executed, resulting in a table of experimental features. The table has four columns: Name, Enabled, Source, and Description. All features listed are currently disabled (Enabled: False) and originate from the PSEngine.

Name	Enabled	Source	Description
PSCommandNotFoundSuggestion	False	PSEngine	Recommend potential commands based on fuzzy search on a CommandNotFoundException
PSImplicitRemotingBatching	False	PSEngine	Batch implicit remoting proxy commands to improve performance
PSTempDrive	False	PSEngine	Create TEMP: PS Drive mapped to user's temporary directory path
PSUseAbbreviationExpansion	False	PSEngine	Allow tab completion of cmdlets and functions by abbreviation

# PSCommandNotFoundSuggestion

- `Enable-ExperimentalFeature -Name PSCommandNotFoundSuggestion`
- Bei `CommandNotFoundException` werden Vorschläge für Cmdlets ähnlicher Schreibweise gemacht



```
PowerShell 6 (x64)
PS C:\Users\Haiko.Hertes> Get-ChilItem
Get-ChilItem : The term 'Get-ChilItem' is not recognized as the name of a cmdlet, function, script file,
or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

At line:1 char:1
+ Get-ChilItem
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Get-ChilItem:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Suggestion [4,General]: The most similar commands are: Get-ChildItem, Get-Item, Set-Item.
```

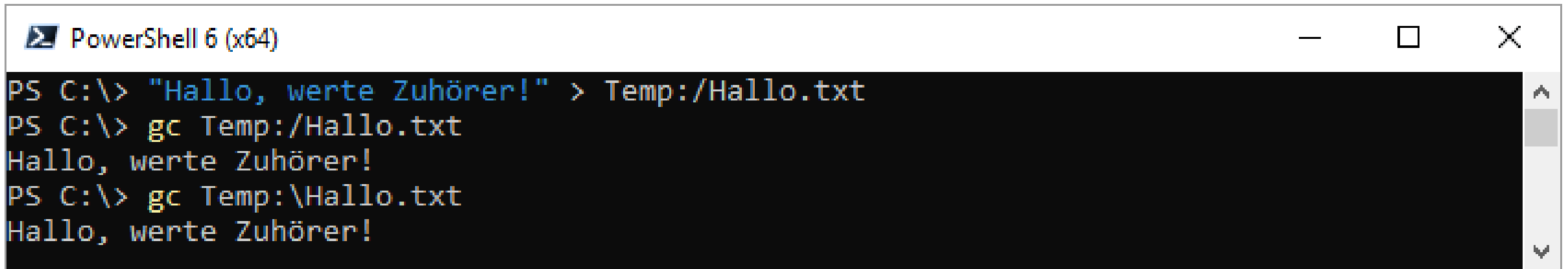


# PSImplicitRemotingBatching

- `Enable-ExperimentalFeature -Name PSCommandNotFoundSuggestion`
- Explizites Remoting:
  - `Invoke-Command -ScriptBlock { Get-ADComputer -Filter * } -ComputerName MyDC1`
- Implizites Remoting:
  - `$DCSession = New-PSSession -ComputerName MyDC1`
  - `Import-PSSession -Session $DCSession -CommandName *-AD* -Prefix 'r'`
  - `Get-rADComputer -Filter *` ← läuft jetzt remote!
- Ohne „Implicit Remote Batching“: Objekte werden bei Implicit Remoting und Verwendung der Pipeline wiederholt serialisiert und de-serialisiert (kostet Zeit)
- Mit „Implicit Remote Batching“ wird vorher evaluiert, ob Befehl remote ausgeführt werden kann und dann wird gesamte Pipeline remote ausgeführt (Serialisierung/Deserialisierung nur einmal bei Rückgabe des Ergebnisses)

# PSTempDrive

- `Enable-ExperimentalFeature -Name PSTempDrive`
- TEMP: zeigt auf TEMP-Pfad des jeweiligen OS
- Egal wo sich dieser befindet
  - Struktur bei verschiedenen OS anders
  - ENV-Variable heißt bei diversen OS anders



```
PowerShell 6 (x64)
PS C:\> "Hallo, wert Zuhörer!" > Temp:/Hallo.txt
PS C:\> gc Temp:/Hallo.txt
Hallo, wert Zuhörer!
PS C:\> gc Temp:\Hallo.txt
Hallo, wert Zuhörer!
```

# PSUseAbbreviationExpansion

- `Enable-ExperimentalFeature -Name PSUseAbbreviationExpansion`
- Cmdlets sollen „deskriptive“ Nouns enthalten
- => PowerShell Cmdlets können sehr lange Namen haben
- Dieses Feature erlaubt „tabben“ auf Basis der Anfangsbuchstaben!
- `g-awaccdu => Get-AzWebAppContainerContinuousDeploymentUrl`

# Achtung

- Diese Features sind nicht ohne Grund „experimental“!
- Änderungen bis zum Release sind möglich und denkbar
- Mit `Disable-ExperimentalFeature` lassen sich die Features wieder abschalten

A modern building with a glass facade and a courtyard. The image is split into two parts: a dark, low-key view of the building on the left and a bright, high-key view of the courtyard on the right. A large red triangle is in the top right corner.

# Demo

## Features II



A photograph of a modern office interior with large windows. Several people are seated at a long table, their silhouettes visible against the bright light coming from the windows. The image is split diagonally: the left side is dark grey, and the right side is a bright blue sky. A red triangle is in the bottom right corner.

# SUPPORT

# Support

- PowerShell Core nutzt Microsoft Modern Lifecycle Policy (noch)
- Ein neues (minor) Update alle 6 Monate vorgesehen
- Für Support: Update spätestens 6 Monate nach Release neuer Minor-Version
- Neue Patch-Versionen (6.1.2 -> 6.1.3) müssen binnen 30 Tagen installiert werden
- [aka.ms/pslifecycle](https://aka.ms/pslifecycle)

# Support

- Mit PS 7: Engere Bindung an .NET Core Lifecycle
- LTS und nicht-LTS Releases geplant
- Weiterhin monatliche Preview Releases (frühzeitiges Feedback)

# HOW ABOUT CLOUD?

# PowerShell und Azure

- PowerShell in Azure Cloud Shell seit Ende 2017 (ursprünglich PS 5.1)
- Aktuell kommt PS Core 6.2 zum Einsatz:
- Seit 04/2019 auch ExchOnline support
  - Connect-ExoPSSession

```
PowerShell | ? | ? | ? | ? | ?
Requesting a Cloud Shell. Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

MOTD: Cmdlet help is available: help <cmdlet name>

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
Azure:/
PS Azure:\> $PSVersionTable

Name                           Value
----                           -
PSVersion                      6.2.0
PSEdition                      Core
GitCommitId                    6.2.0
OS                              Linux 4.15.0-1041-azure #45-Ubuntu SMP Fri Mar 15 14:41:00 UTC 2019
Platform                       Unix
```

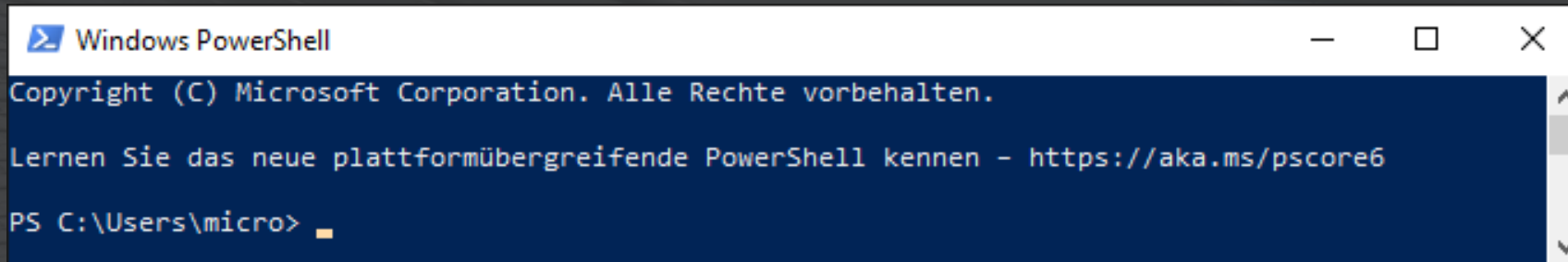
- Azure Functions mit PowerShell (Core, what else...) seit 04/2019 (Public Preview)
- Mehr morgen! (Track 2, 15:30, Thomas Maurer)





ONE LAST THING...

# Danke für eure Aufmerksamkeit!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\Users\micro>
```

[www.softwareone.com](http://www.softwareone.com)

This publication contains proprietary information that is protected by copyright. SoftwareONE reserves all rights thereto.

SoftwareONE shall not be liable for possible errors in this document. Liability for damages directly and indirectly associated with the supply or use of this document is excluded as far as legally permissible.

The information presented herein is intended exclusively as a guide offered by SoftwareONE. The publishers product use rights, agreement terms and conditions and other definitions prevail over the information provided herein. The content must not be copied, reproduced, passed to third parties or used for any other purposes without written permission of SoftwareONE

Copyright © 2019 by SoftwareONE. All Rights Reserved. SoftwareONE is a registered trademark of SoftwareONE. VARassist is a trademark of SoftwareONE Incorporated and "it pays to partner" is a service mark of the VARassist program offered by SoftwareONE. All other trademarks, service marks or trade names appearing herein are the property of their respective owners.

# Danke an unsere Partner!

Platinum Sponsor



Gold Sponsoren





# Danke an unsere Partner!

Gold Sponsoren

**TAR•X**

veeam

**Lenovo**

**DELL**EMC

interxion™



***SecureGUARD***

 **Azure Stack**  
ALLIANCE