

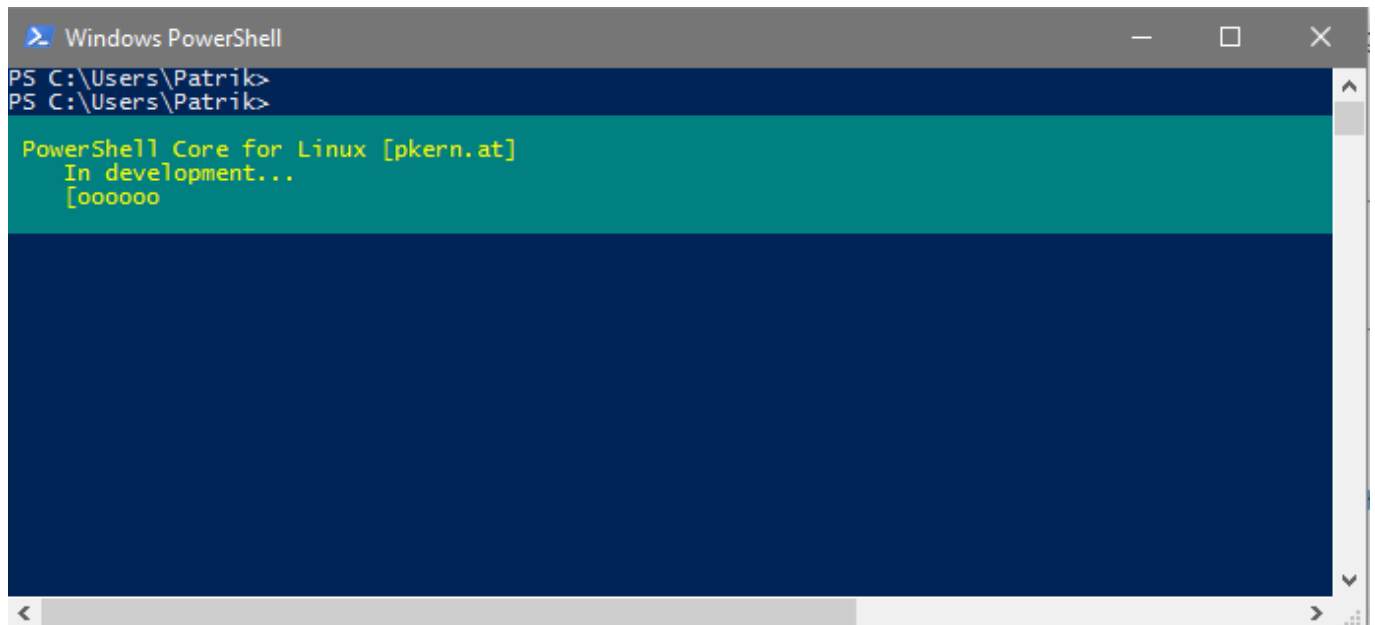
PowerShell für Linux und macOS

Author: Patrik Kernstock

Categories: [Allgemein](#), [Linux](#), [Microsoft](#), [Microsoft Windows](#), [Tipps & Tricks](#)

Date: 22. August 2016

Here it comes... PowerShell Core.



```
Windows PowerShell
PS C:\Users\Patrik>
PS C:\Users\Patrik>

PowerShell Core for Linux [pkern.at]
In development...
[ooooooooo]
```

PowerShell auf Windows

Überraschung! Microsoft hat vor wenigen Tagen die von der Windows-Welt allbekannte **PowerShell für Linux- und macOS Systeme öffentlich freigegeben**. Und zwar steht nun das Projekt **auf GitHub** unter der MIT-Lizenz - also **OpenSource** - für jeden frei zur Verfügung. Es gab bereits vor einigen Wochen einige Spekulationen und mehrfache inoffizielle Aussagen, dass Microsofts Shell OpenSource auch für andere Systemen erscheinen soll. Diese haben sich [hiermit nun bestätigt](#)...

Verfügbarkeit

PowerShell Core steht **aktuell für diverse Windows-Systeme, Ubuntu 14.04, Ubuntu 16.04, CentOS 7 und OS X 10.11 als vorbereitete Pakete zur Installation bereit**. Mehr sind für die Zukunft geplant. Die Shell ist hauptsächlich in der Microsoft-eigenen Programmiersprache C# geschrieben, welche nun durch die vergangenen Züge für eine Cross-Plattform-Unterstützung (Stichwort [.NET Core](#)) seitens Microsoft nun auch auf anderen Systemen

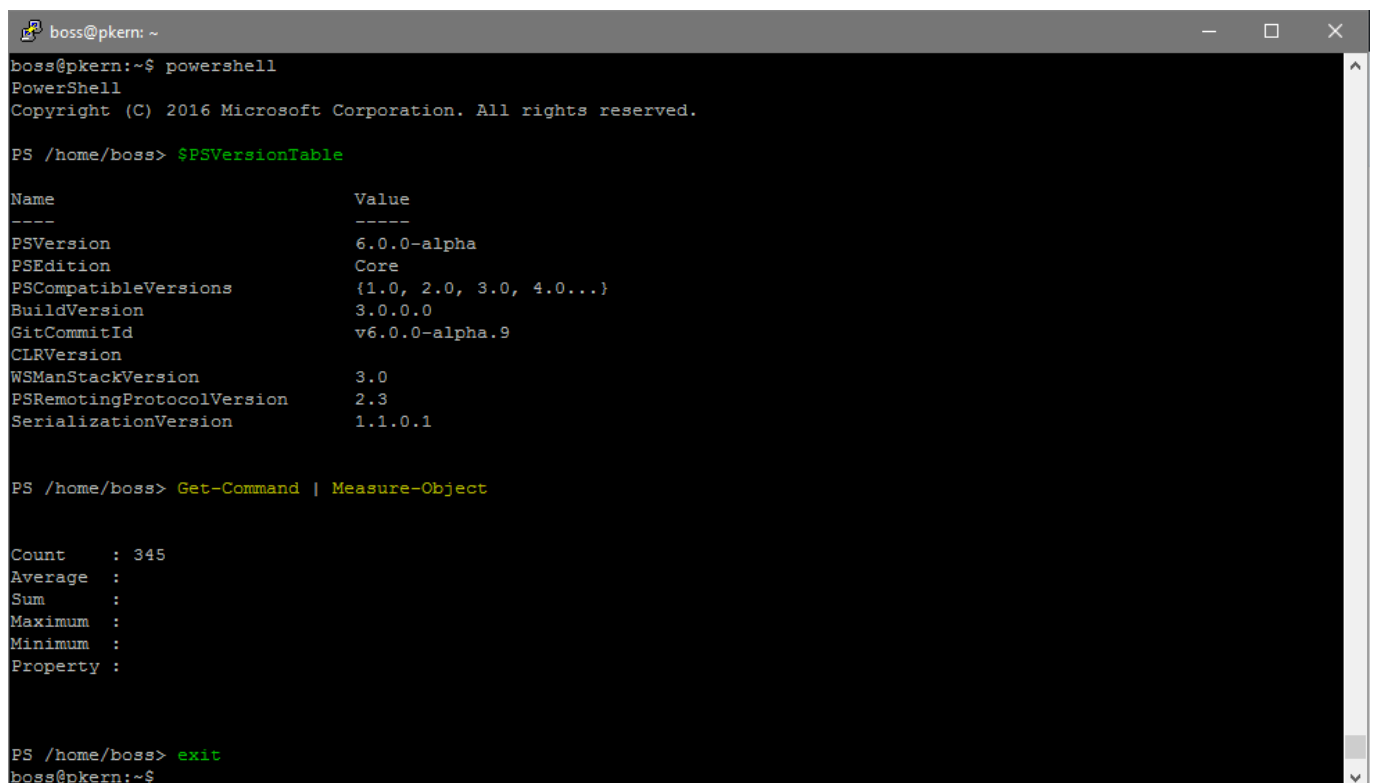
lauffähig ist.

Installation

Die Auflistung sowie **Instruktionen zur Installation** für die jeweiligen Systemen gibt es auf GitHub [hier zu finden](#). Die Installation ist mit wenigen Befehlen und **innerhalb relativ kurzer Zeit getan**. Möchte man PowerShell für eine nicht-gelistete Distribution nutzen, so lässt sich die Shell auch selbst compilieren - wozu auf GitHub ebenfalls [Anleitungen bereitstehen](#).

Verwendung

Wenig überraschend kann die PowerShell durch die **simple Eingabe von "powershell" gestartet werden**. Nach einer kurzen Ladezeit ist dann die Shell verwendbar. Wie sich in dem Screenshot bei meinem Test entnehmen lässt, trägt die Shell derzeit die Versionsnummer 6.0.0 mit einem "alpha"-Tag. Im aktuellen Entwicklungsstadium sind **etwa 345 Befehle implementiert** und verwendbar.



```
boss@pkern: ~
boss@pkern:~$ powershell
PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS /home/boss> $PSVersionTable

Name                           Value
----                           -
PSVersion                       6.0.0-alpha
PSEdition                       Core
PSCompatibleVersions             {1.0, 2.0, 3.0, 4.0...}
BuildVersion                     3.0.0.0
GitCommitId                      v6.0.0-alpha.9
CLRVersion                       3.0
WSManStackVersion                3.0
PSRemotingProtocolVersion        2.3
SerializationVersion             1.1.0.1

PS /home/boss> Get-Command | Measure-Object

Count      : 345
Average    :
Sum         :
Maximum    :
Minimum    :
Property   :

PS /home/boss> exit
boss@pkern:~$
```

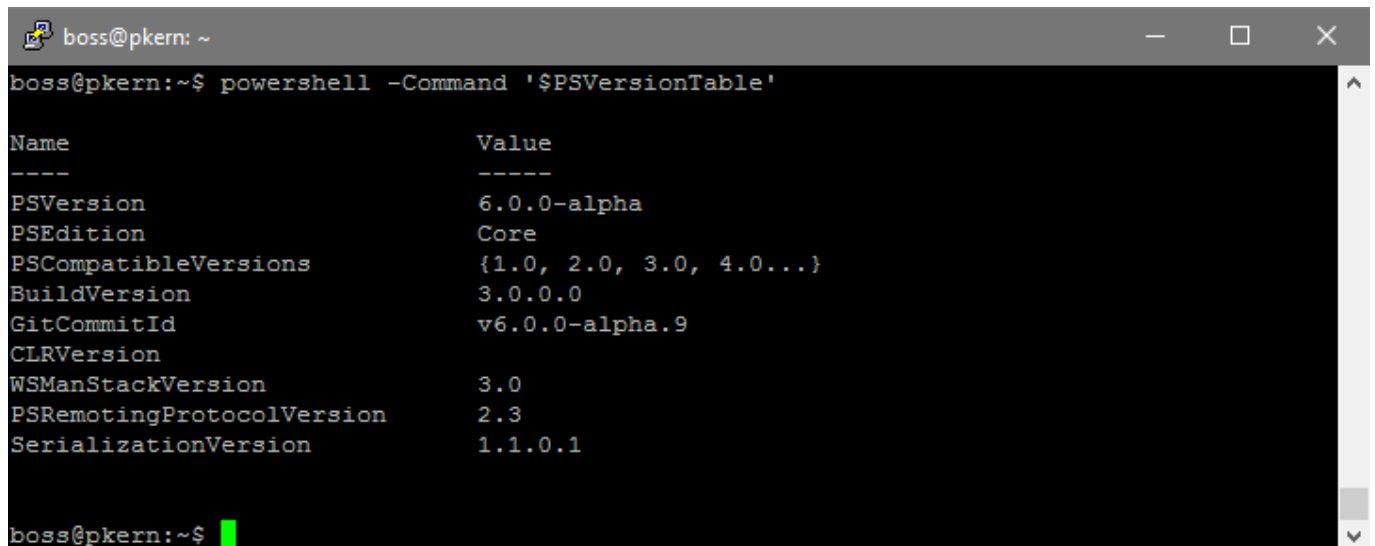
Interaktive PowerShell

Was ich auch noch probiert habe, ist es **PowerShell-Befehle via Parameter direkt an die Shell zu übergeben** und diese auszuführen - wie dies auch bei der Windows-Variante möglich ist. Und siehe da, auch das funktioniert wie erwartet:

[pkernstock@pkern.at]\$

portfolio, computer & more

https://pkern.at



```
boss@pkern: ~  
boss@pkern:~$ powershell -Command '$PSVersionTable'  
Name                Value  
----                -  
PSVersion           6.0.0-alpha  
PSEdition           Core  
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}  
BuildVersion        3.0.0.0  
GitCommitId         v6.0.0-alpha.9  
CLRVersion            
WSManStackVersion   3.0  
PSRemotingProtocolVersion 2.3  
SerializationVersion 1.1.0.1  
  
boss@pkern:~$
```

Direkte Ansteuerung der PowerShell

Ein paar Wörter zum Thema PowerShell Module...

Nicht zu vergessen gilt es, dass es sich **hierbei um PowerShell Core handelt und nicht über die vollwertige PowerShell**, welche man von den Windows-Systemen her kennt. PS Core ist **eine Art abgespeckte Version**, die auf der neuen **cross-platform Laufzeitumgebung** namens *".NET Core Common Language Runtime"* (in kurz: *CoreCLR*) **aufbaut**.

Das ist mitunter auch der Grund, wieso **nicht alle PowerShell-Module auch auf der PS Core lauffähig** sind. Gewisse Module besitzen diverse **Abhängigkeiten von anderen Bibliotheken**, die bislang **nicht in PS Core implementiert** sind. Als konkretes Beispiel lässt sich hier SMBShare heranziehen. Dieses Modul basiert auf den CIM-Klassen, welche lediglich auf Windows-Systemen beheimatet sind. Daher läuft dieses Modul nicht auf PS Core.

Diverse Unternehmen mit bereits veröffentlichten PowerShell Modulen haben bereits bekannt gegeben, auch **angepasste Bibliotheken für PS Core entwickeln und veröffentlichen zu wollen**. Vorreiter sind hier AWS und VMware. Laut Alan Renouf von VMware sind in der PowerCLI-Version für PS Core alle Befehle zur grundlegenden Administration mit einem VMware vCenter bereits möglich.

Fazit

Ich **begrüße durchaus den Schritt von Microsoft**, die Shell auch auf andere Systeme zu portieren. Meiner Meinung nach sollte man aber die PS Core nicht als Konkurrent/Ersatz von den Linux-Shell's wie beispielsweise Bash sehen, sondern eher als **ergänzende Lösung** für etwaige verschiedenste Automatisierungs- und Konfigurationsszenarien - besonders für

Windows-Maschinen als Zielsystem.

Die Installation verlief bei meinen Tests auf unterschiedlichen Testsystemen problemlos, und war auch ruck-zuck erledigt. Ich habe mich zwar bisher noch nicht intensiv damit auseinandergesetzt, dennoch **macht die Shell einen guten Eindruck**. Und nicht vergessen: PowerShell Core befindet sich aktuell noch **in der frühen Alpha-Phase** und ist noch stark in der Entwicklung. Die Shell kann daher auch noch Fehler beinhalten und zu unerwünschten Ergebnissen führen. Daher ist die **Verwendung in Produktivumgebungen nicht empfohlen**.

Ich bin sehr gespannt, wie sich das Ganze in der Zukunft noch entwickeln wird...

Quellen:

- [Microsoft MSDN Blog](#)
- [PowerShell GitHub Seite](#)
- [FAQ-Artikel auf Powershell.org](#)