

Leer werken met PowerShell *Versie 2021*

Roland Sellis

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: Roland Sellis
Redactie: Martijn Niemeijer
Opmaak: Henk Pel
Titel: Leer werken met PowerShell. Versie 2021
ISBN 978 90 372 5913 1
Eerste druk / eerste oplage
© Boom beroepsonderwijs 2021

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoud

Inleiding	7
Werkwijze	9

Deel I Introductie PowerShell

1	Wat Is PowerShell?	13
2	Basisnetwerk	15
3	De pipeline	28
4	Eenvoudig script	33

Deel II PowerShell & Active Directory

5	Basisnetwerk	53
6	Modules	55
7	Users en groups	57
8	Errorhandling	62
9	Externe data gebruiken	65
10	User input	71

Deel III Remote Administration & Core Server

11	Basisnetwerk	77
12	Remote Server Administration Tools	79
13	Computermanagement	83
14	PowerShell Remoting	84
15	Installatie webservers	90
16	GPO: remote management	95
17	Installatie IIS	101
18	Websites en DNS	107
19	Rol verwijderen	110
20	Domain Services	112

Deel IV PowerShell 7, VS Code en GUI

- 21 Basisnetwerk 117
- 22 PowerShell 7 119
- 23 Visual Studio Code editor 122
- 24 Graphical User Interface 125

Bijlagen

- Bijlage A Scripts 137
- Bijlage B Handige commando's 145
- Bijlage C VM-tools 148

Inleiding

Van oudsher beschikt het Microsoft-platform al over een command-line interface (CLI). In de eerste versies was dat de command.com (MS-DOS) of de cmd.exe (NT-versies). Deze CLI was beperkt en bood alleen beperkte mogelijkheid tot beheertaken. Met bijvoorbeeld Visual Basic Scripting Edition uit 1996 waren er al weer meer mogelijkheden, maar VBScript was niet makkelijk te leren.

In 2003 kondigde Microsoft PowerShell aan: een objectgeoriënteerde scripttaal gebaseerd op het .NET Framework. Met PowerShell kunnen IT-professionals het beheer automatiseren van het Windows-besturingssysteem en talloze toepassingen die in Windows worden uitgevoerd.

Als je veel met beheer van Microsoft-systemen te maken hebt of krijgt is het meer dan raadzaam je te verdiepen in deze scripttaal. In de (nabije) toekomst zal PowerShell meer en meer gebruikt worden. In veel Microsoft-producten is PowerShell naadloos verweven. Opties die je in de grafische interface kiest worden op de achtergrond door PowerShell-commando's uitgevoerd. Om taken makkelijker en sneller te kunnen herhalen, loont het de moeite hiervoor een PowerShell-script te schrijven.

De leercurve voor PowerShell is niet zo steil, wat inhoudt dat het vrij eenvoudig is om een start te kunnen maken. Gedurende het gebruik merk je vanzelf dat je meer vragen krijgt waarop je een antwoord wilt. Daardoor gaat PowerShell steeds meer leven en betekenis krijgen. Er is namelijk zo ontzettend veel mogelijk.

Met dit boek wil ik je zo'n start geven. Het bestaat uit vier delen:

- Deel I Introductie PowerShell
- Deel II PowerShell & Active Directory
- Deel III Remote administration & Core Server
- Deel IV PowerShell 7, VS Code en GUI

In het eerste deel bied ik je de mogelijkheid om te beginnen met PowerShell en zodoende de basis te leren. In deel II ga je al wat geavanceerder te werk en importeer je nieuwe gebruikers in Active Directory met behulp van een PowerShell-script. In het derde deel beheer je, met behulp van tools en PowerShell, servers remote. De servers voorzien van een Windows Server Core installatie lenen zich uitermate voor beheer met behulp van PowerShell aangezien er geen grafische interface is om op terug te vallen. In het laatste deel kijken we naar de laatste versie, PowerShell 7, leren we gebruik maken van een nieuwe editor en maken we een grafische interface voor onze scripts.

Dit boek pretendeert absoluut niet volledig te zijn of de geweldigste oplossing te bieden. Verre van dat. Mijn doel is om je kennis te laten maken met deze krachtige tool en je de basis bij te brengen hoe je ermee kunt werken. Het is aan jou om dit verder te verkennen en er optimaal gebruik van te maken.

Ik wens je veel succes en plezier,

Roland Sellis

Rotterdam, najaar 2020

Werkwijze

In dit boek maak ik gebruik van:

- het Windows Server 2019 besturingssysteem als een virtuele machine;
- de afkorting PS in plaats van het volledige woord PowerShell;
- het lettertype DejaVu Sans Mono als het om codetekst gaat (de commando's);
- de toetsencombinatie CTRL+C om een script af te breken in de shell;
- tab-completion: door genoeg unieke tekens van een commando te typen en vervolgens de tab-toets te gebruiken vult PS het commando verder aan, wat enorm scheelt in het typen van de vele commando's;
- codes die de kleurcodes van PowerShell ISE gebruiken.

Virtuele machine

Alle screenshots en opdrachten in dit boek zijn gemaakt in een zogenaamde virtuele machine (VM), een applicatie waarmee een computer nagebootst wordt. Er zijn meerdere applicaties beschikbaar waarmee dit mogelijk is, van open source zoals VirtualBox tot commerciële applicaties als VMware en Parallels of Hyper-V, die wordt meegeleverd met Windows-versies vanaf Windows Server 2008. Het gebruikte besturingssysteem is Windows Server 2019 met alle updates tot en met augustus 2020.

Documentatie

Maak een Word-document, noem dit document *PowerShell_<je_volledige_naam>* en bewaar het op een plaats waar je makkelijk bij kunt en waarvandaan je het kunt inleveren als daarom wordt gevraagd.

In dit document verzamel je de antwoorden en screenshots van de diverse opdrachten. Op die manier kun je makkelijk een commando terughalen en opnieuw gebruiken.

De opdrachten geef je duidelijk aan met: *Opdracht 1.1*, *Opdracht 1.2* enzovoort.

Verderop in het boek leer je ook hoe je scripts maakt. Deze hebben de extensie .ps1. Maak daarom ook een map met de naam *PS_Scripts* waarin je een kopie van elk script zet. De naam van het script begint met de opdracht, bijvoorbeeld: *4.5_loginscript.ps1*.

De inhoud van het script plak je ook als tekst in het Word-document. Als in dit boek over documentatie wordt gesproken, wordt daarmee dit Word-document bedoeld.

Opdrachten

Alle opdrachten moeten worden gemaakt. Meestal zijn er meerdere mogelijkheden om tot een werkend script te komen. In de docentenhandleiding staan alle opdrachten uitgewerkt. Het kan best dat jouw antwoord afwijkt van het antwoord in deze docentenhandleiding. Dat wil niet altijd zeggen dat jouw antwoord of oplossing fout is.

Voorbeelden

Op internet zijn tal van voorbeelden van PS-scripts te vinden (zoek op PowerShell, cmdlet of PS1). Hier mag je tijdens het maken van de opdrachten in dit boek ook volop gebruik van maken, mits je het script je zo eigen maakt dat je snapt wat het doet en hoe je elementen kunt wijzigen. Kortom, je moet het kunnen toepassen in jouw omgeving. Maak het jezelf dus niet te moeilijk door gigantische scripts te gebruiken die door een ander gemaakt zijn zonder zelf te weten wat het doet.

Productie

Uiteraard maak en test je een script altijd eerst in een lab/testomgeving en nooit direct in een productieomgeving. Pas als je zeker weet wat je script doet en wat je kunt verwachten, kun je het gaan gebruiken in een live-omgeving. Zorg dat je nooit voor verrassingen komt te staan.

Deel I

Introductie PowerShell

1 Wat Is PowerShell?

Microsoft PowerShell is een objectgeoriënteerde scripttaal die gebaseerd is op commando's, de zogenaamde cmdlets. Deze naam is speciaal door Microsoft bedacht en daardoor uniek, zodat het zoeken ernaar via een zoekmachine op internet ook makkelijk gaat. Doel is dat de beheerder zijn/haar taken makkelijker, beter en sneller kan uitvoeren. De leercurve is niet zo steil en daardoor bijvoorbeeld makkelijker te leren dan Visual Basic.

PowerShell kan werken met Windows-besturingssystemen vanaf Windows XP en daarnaast ook met diverse andere Microsoft-producten zoals Exchange, System Centre Operations Manager en System Center Virtual Machine Manager. Sterker nog, met het kiezen van diverse opties in de GUI (graphical user interface) wordt op de achtergrond een PowerShell-script opgebouwd en uitgevoerd. Er zijn al verschillende versies van PowerShell de afgelopen jaren voorbijgekomen, waarvan hieronder een kort overzicht.

Versie 1

Uitgebracht in 2006 en beschikbaar voor de volgende Windows-versies: Windows XP SP2, Windows Server 2003, Windows Vista, Windows Server 2008.

Versie 2

Beschikbaar voor de volgende Windows-versies: Windows XP SP3, Windows Server 2003 SP2, Windows Vista SP 1, Windows 7 en Windows Server 2008 R2.

Versie 3

Beschikbaar voor de volgende Windows-versies: Windows 7 SP1, Windows Server 2008 SP1, Windows Server 2008 R2 SP1, Windows 8 en Windows Server 2012.

Versie 4

Beschikbaar voor de volgende Windows-versies: Windows 8.1 en Windows Server 2012 R2.

Versie 5

Beschikbaar voor de volgende Windows-versies: Windows 8.1, Windows 10, Windows Server 2012 R2 en Windows Server 2016.

Versie 6

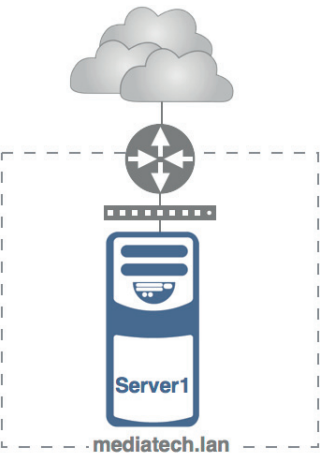
Met versie 6 veranderde er veel, waaronder de naam: PowerShell Core. In tegenstelling tot voorgaande versies, die gebouwd zijn op het .NET Framework is versie 6 gebouwd op .NET Core (spreek uit: *dot net core*). Door deze wijziging is PowerShell cross-platform en open-source geworden.

Versie 7

Met versie 7 wijzigt de naam naar PowerShell, zonder toevoegingen. Ook PowerShell is gebaseerd op .NET Core, al is het wel een hogere versie (3.1). PowerShell versie 7.1 is gebouwd op .NET 5, waar de toevoeging 'Core' ook weggelaten wordt.

2 Basisnetwerk

Dit boek starten we met een basisnetwerk en basisconfiguratie. De situatie zie je hieronder weergegeven. We maken in deel I gebruik van 1 Domain Controller in het domein mediatech.lan.



Afb. 2.1 Overzichtstekening netwerk mediatech.lan

In deel I worden de volgende hosts gebruikt:

	Naam	Server1
	IP-adres	172.16.X.10
	Subnetmasker	255.255.255.0
	Default gateway	172.16.X.1
	DNS-server	172.16.X.10
	OS	Windows Server 2019
	Services	ADDS, DNS

	Naam	Router
	WAN	Automatisch
	LAN	172.16.X.1
	Subnetmasker	255.255.255.0

Opdracht 2.1

Om de opdrachten te kunnen maken en testen heb je een labomgeving nodig. Deze bestaat uit een domain controller met Windows Server 2019. Maak een virtuele machine met Windows Server 2019 als besturingssysteem. Je mag zelf kiezen of je de server als router of een andere router gebruikt. Zorg ervoor dat beide VM's up-to-date zijn als je met de PowerShell-opdrachten begint.

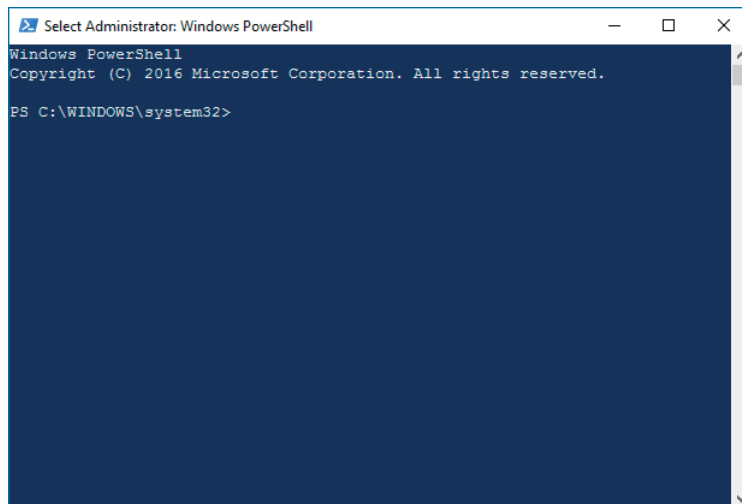
De omgeving

Binnen Windows heb je twee omgevingen waarbinnen je direct kunt werken met PowerShell-commando's:

- Windows PowerShell
- Windows PowerShell ISE (Integrated Scripting Environment)



Windows PowerShell

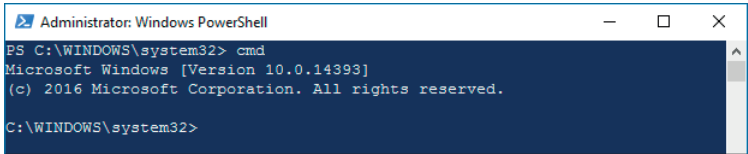


Afb. 2.2 De klassieke PowerShell-omgeving zonder al te veel hulpmiddelen

Tip Soms staat de breedte van het venster anders ingesteld dan je monitor of je VM toelaat. Je ziet dan een horizontale schuifbalk, wat niet praktisch is tijdens het schrijven van commando's. Om dit te veranderen kun je in de eigenschappen van het venster, in het tabblad *Layout*, de waarde van de *screenbuffer size* gelijk maken aan de waarde van de *Window Size*. Overigens zijn er nog veel meer instelmogelijkheden in de eigenschappen te vinden die erg handig kunnen zijn.

Je ziet in het PS-venster eigenlijk niets anders dan wat gegevens en de prompt. Het verschil met een normale command prompt zie je nu niet alleen aan de kleur, maar ook aan de letters PS voor de prompt.

Overigens kun je door het commando `cmd` te typen ook gewoon naar de command prompt en heb je nog steeds dezelfde achtergrondkleur, maar kun je op dat moment geen PS-commando's meer uitvoeren. Met het commando `exit` keer je weer terug naar PowerShell.

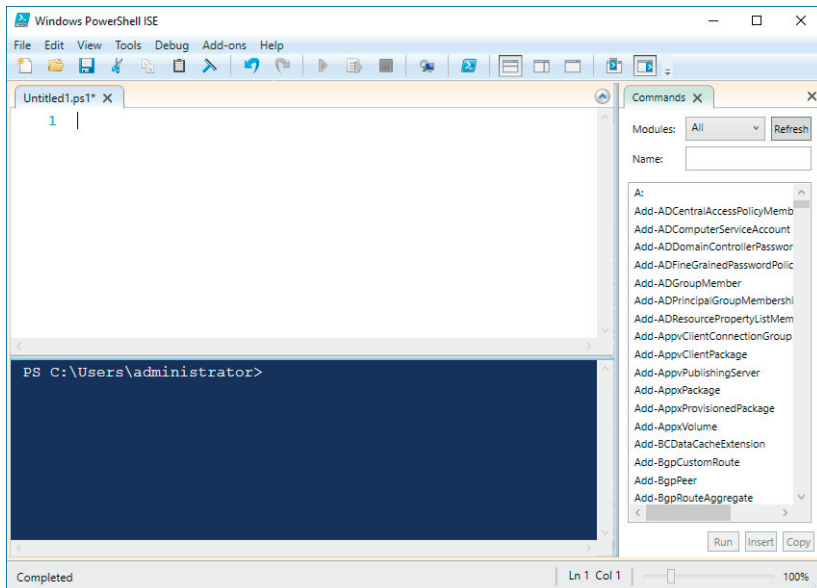


Afb.2.3 Met het commando `cmd` verschijnt de command prompt



Windows PowerShell ISE

Er is ook een uitvoering van PowerShell die de toevoeging ISE heeft, wat staat voor Integrated Scripting Environment.

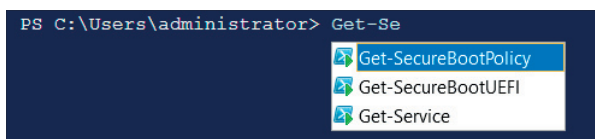


Afb.2.4 Windows PowerShell ISE

Tip Eenregelige commando's kun je direct typen in het commandovenster (blauwe achtergrond). Scripts bestaan vaak uit meerdere regels. Die typ je in het venster met de witte achtergrond. Met een Enter ga je naar de volgende regel om de volgende instructie te typen. Met de afspeelknop (groene driehoekje) voer je het script uit. Dat zie je dan ook gebeuren in het commandovenster.

De ISE-uitvoering kent een aantal handigheidjes waaronder:

- gebruik van standaard toetsenbord-snelkoppelingen (CTRL+a = alles selecteren; CTRL+c/CTRL+v = kopiëren/plakken);
- door middel van tabs aan meerdere sessies en scripts tegelijk kunnen werken;
- handige zoek- en hulpmiddelen;
- ingebouwde IntelliSense: tijdens het typen worden suggesties getoond.



Afb. 2.5 Met de hulp van IntelliSense wordt het allemaal nog makkelijker

In dit boek zullen we voornamelijk de ISE-shell gebruiken. Voor beide uitvoeringen geldt wel dat je een aantal zaken goed moet controleren:

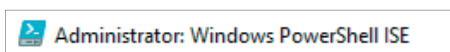
- Versienummer: deze moet minimaal versie 5.0 zijn. Controleer dit om problemen te voorkomen met het commando `$PSVersionTable`.

```
PS C:\> $PSVersionTable

Name                           Value
----                           -
PSVersion                      5.1.18362.752
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.18362.752
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

Afb. 2.6 Controleer het versienummer van PowerShell

- PS moet als administrator uitgevoerd worden. Dit is te zien in de menubalk. Mocht hier geen Administrator staan, dan moet je PS afsluiten en met een rechtermuisklik kiezen voor de optie *Run as Administrator*.



Afb. 2.7 Voer PowerShell als administrator uit

De syntax

Zoals in elke scripttaal heeft ook PS een zogenaamde syntax, met andere woorden: de manier waarop een commando in elkaar steekt, waaronder de volgorde. Als voorbeeld het volgende commando:

```
Get-ChildItem -path C:\Windows\System32 -Filter *.exe -Recurse
```

Het eerste deel is (meestal) een werkwoord: `Get`

Om alle mogelijke werkwoorden te zien kun je het commando `Get-Verb` gebruiken. Niet alle mogelijkheden zijn werkwoorden. Zo zie je ook `New` staan, dit had wellicht beter `Create` kunnen zijn.

```
PS C:\WINDOWS\system32> Get-Verb

Verb      Group
----      -
Add       Common
Clear     Common
Close     Common
Copy      Common
Enter     Common
Exit      Common
Find      Common
Format    Common
```

Afb. 2.8 Met `Get-Verb` eenvoudig cmdlets zoeken

Tip Het aantal mogelijkheden in commando's, parameters en opties is afhankelijk van de installatie van het besturingssysteem. Naarmate er meer rollen of features geïnstalleerd zijn, neemt het aantal mogelijkheden in PowerShell ook toe.

Vervolgens wordt het werkwoord altijd gevolgd door een afbreekstreepje/divisie (-), waarna een zelfstandig naamwoord volgt. In bovenstaand voorbeeld: `ChildItem`.

Vervolgens komen de parameters (opties) met daarachter de value (waarde). In het voorbeeld staan er drie:

- `path` met daarachter de value (waarde) `C:\Windows\System32`
- `Filter` met daarachter de value (waarde) `*.exe`
- `Recurse` met daarachter geen value (waarde).

Het totale commando betekent dus:

laat alle bestanden (`Get-ChildItem`) zien in de folder `C:\Windows\System32` (`-Path`) die elke naam mogen hebben (*) maar moeten eindigen met de extensie `.exe` en daarvoor moet je ook zoeken in de submappen onder `System32` (`-Recurse`)

De `*` wordt hier gebruikt als een wildcard en kan vervangen worden door alle tekens.

*.txt betekent dus zoiets als: er mag van alles voor de punt staan, maar het moet wel eindigen op .txt. Hier kun je uiteraard handig gebruik van maken. Hieronder nog enkele voorbeelden:

verslag*.docx alle bestanden die beginnen met het woord verslag en eindigen met de extensie .docx

Help Get-Proc* geeft in PS alle commando's weer die beginnen met `Get-Proc` zoals `Get-Process`, maar bijvoorbeeld ook `Get-ProcessMitigation`

Een tweede wildcard die je kunt gebruiken is het vraagteken (?). Hiermee kun je net als het * werken, met de uitzondering dat een vraagteken maar voor één teken staat. Voorbeelden:

Verslag?.docx geeft wel verslag1.docx en verslagB.docx, maar niet verslag23.docx

Verslag??.docx geeft wel verslag23.docx maar niet verslag_conferentie.docx

Beide wildcards kun je dan ook nog eens combineren:

- Verslag?.*
- *Verslag*.txt
- *Verslag*.doc?

```
PS C:\WINDOWS\system32> Get-Help Get-Print*

Name                                     Category  Module
----
Get-PrinterPort                         Function  PrintManagement
Get-PrinterProperty                     Function  PrintManagement
Get-PrinterDriver                       Function  PrintManagement
Get-PrintConfiguration                  Function  PrintManagement
Get-PrintJob                            Function  PrintManagement
Get-Printer                             Function  PrintManagement
```

Afb. 2.9 Gebruikmaken van wildcards maakt het zoeken eenvoudiger

Opdracht 2.2

- Controleer de versie van PowerShell.
- Update de help file van PowerShell.