

Inhoud

Opmerkingen vooraf	xvii
Vereiste kennis en benodigdheden	xvii
Hoe gebruik je dit boek?	xix
Optie 1: onderwerp na onderwerp	xix
Optie 2: meteen starten met het uitbouwen van een domein	xix
Optie 3: uit het boek halen wat nodig is	xix
Tot slot	xx
Deel 1: Wegwijs in PowerShell	1
Introductie	2
Opmerkingen bij de oefeningen (virtualisatiehulp op volgende pagina)	5
Oracle VirtualBox en VMWare: Oracle VirtualBox	6
Oracle VirtualBox en VMWare: VMWare	10
1 Aan de slag	12
2 Eerste instructie uitvoeren	13
3 Gebruikmaken van Windows PowerShell ISE	13
4 Visual Studio Code	16
5 Typische PowerShell-opmerkingen	16
6 Houd rekening met iedereen	17
7 Escapeteken	17
8 Cmdlets	18
9 Snel werken met PowerShell	19
10 Parameters bij cmdlets	20
11 Anatomie van PowerShell-commando	21
12 Bouwstenen	21
13 Waarde toekennen aan een variabele	22
14 Opbouwen van een stevige instructie	23
15 Gebruik -whatif om instructie te controleren	28
16 Hulp krijgen over cmdlets	28
17 Twee soorten problemen bij cmdlets	29
18 Begrijp de verschillende soorten haakjes	30
19 Beveiligingsbeleid op orde stellen	32

20	Een vertrouwensrelatie (signed) voor eigen scripts	33
21	Credentials meegeven via een object	34
22	Parameters instellen bij starten PowerShell	35
23	PowerShell starten vanuit een batchbestand of Cmd-venster	35
24	PowerShell automatisch starten als administrator – I	36
25	Werken met variabelen	37
26	De pipeline in het kort	38
27	Objecten	38
28	Eigenschappen	38
29	Methoden	38
	Oefeningen	39
30	Get-Member	39
31	Objecteigenschappen wijzigen	40
32	Verkregen output wijzigen	41
	Groeperen en sorteren	42
	Output extern verwijzen	43
33	Een transcript krijgen van de instructieverwerking	44
34	Te veel op het venster: more versus paging	44
35	Aliassen in PowerShell	44
36	PowerShell-profiel maken	45
37	Snap-ins in PowerShell	45
38	Providers in PowerShell	46
39	Registerinstructie klaarmaken: transaction	48
40	Navigeren in PowerShell	48
41	Modules in PowerShell: basis	49
	Locaties van modules in PowerShell	51
	Oefeningen	52
42	Op afstand met PowerShell	52
	VirtualBox	52
	VMWare	53
	Beide machines zitten in hetzelfde netwerk	54
43	PSSessions: remote via PowerShell	58
44	Navigeren via remote sessie op één remote pc	59
	Listeners en nodige configuraties	61
45	Invoke-Command – instructies op afstand uitvoeren	61
46	Double hop	62
	CredSSP inschakelen	62
47	Achtergrondtaken lokaal uitvoeren	63
48	Achtergrondtaken remote uitvoeren	64
49	Achtergrondtaken inplannen	64
	Oefeningen	65
50	Datatypes	65
51	Variabelen en constanten	66

52	Arrays	67
	Eendimensionale array	67
	Multidimensionale arrays	69
53	Hashtables	69
54	List	71
55	Dictionary	72
56	Queues & stack	73
57	Splatting	73
58	Wiskunde en de operatoren	74
59	Vergelijkingsoperatoren	75
60	Logische operatoren	75
61	.NET-operatoren	76
62	Regex: regular expressions	76
63	If-statement	80
64	Switch-statement	83
65	Iteratie: for-statement	85
66	Iteratie: foreach-statement	86
67	Iteratie: while-statement	87
68	Iteratie: do until-statement	88
69	Iteratie: do while-statement	88
70	Casting en werken met resultaten	89
71	Methodes, functies en cmdlets	90
72	Zelfgedefinieerde functies in het kort	91
73	Het bereik van een variabele	92
74	Werken met argumenten	94
75	Stack en heap	94
	Reference- en value-datatypes	95
	Reference- en value-vergelijkingen	95
	Oefeningen	96
76	Inzicht in het helpstelsel	96
77	Metadata van een cmdlet bekijken	97
78	Pipelinebinding (byValue of byPropertyName)	98
79	CSV-bestanden	102
80	XML-bestanden	104
81	HTML-bestanden	105
82	JSON-bestanden	106
83	Where-Object	107
	Filterscript	107
84	Select-Object	107
85	Sort-Object	108
86	Group-Object	109
87	Measure-Object	109
88	ForEach-Object	109
89	Compare-Object	110

Inhoud

90	Tee-Object	111
91	Bestandscatalogus en werken met hashes	111
	Hashes	111
	Oefeningen	112
92	PowerShell-modules en -scripts online	112
93	Werken met modulemanifests	117
94	.NET	118
95	Nieuwe objecten maken	119
96	Statische mogelijkheden	120
97	Accelerators	122
98	Using	123
99	Assembly's en using	123
100	PowerShell automatisch starten als administrator – II	124
101	.NET-klasse convert	124
102	Stringbewerkingen	125
103	Geavanceerde getalbewerkingen	127
104	Datumbewerkingen	129
	Oefeningen	131
105	Standaard PowerShell-variabelen	131
106	(D)COM	131
107	WMI	132
108	CIM-variabelen	134
109	Eigenschappen van bestanden en mappen	136
110	Rechten van bestanden en mappen	137
111	Rechten wijzigen	138
112	Eigenaarschap overnemen	141
113	Module NTFS	141
114	Eigenschappen zijn ook objecten	143
115	Veilig wachtwoordbeheer	143
	Oefeningen	146
116	Domein starten (voor wie dit grafisch met onderstaande hulp kan)	146
117	PowerShell Web Access	147
118	Centrale informatiemap maken	148
	Oefeningen	149
	Bronnen	150
	Links	150
	Boeken	150
	Schoonmaak	150

Deel 2: Visual Studio Code en PowerShell Core	151
119 Visual Studio Code en PowerShell Core	152
Installatie PowerShell Core	152
Installatie Visual Studio Code	152
PowerShell activeren in Visual Studio Code	153
Command Palette	154
Werkomgeving van PowerShell in Visual Studio Code	155
Nog meer naar een ideale PowerShell-omgeving	156
Bronnen	159
Deel 3: PowerShell Core gebruiken in Linux	161
120 PowerShell Core gebruiken in Linux	162
Installatie van PowerShell in Debian	162
Theoretisch wegwijs in GitHub	163
Installatie van PowerShell op een Red Hat-distributie	165
Werken met PowerShell in Linux in het kort	167
Visual Studio Code installeren op Linux	170
Verbinding op afstand opzetten tussen Linux en Windows	171
Bronnen	174
Schoonmaak	174
Deel 4: Een CLI-netwerkomgeving in de cloud en on-premises	175
Introductie	176
121 Windows Server Core versus GUI	176
122 Installatie Windows Server Core	177
123 Installatie Hyper-V Server 2019	178
124 Installatie GUI-netwerk Windows Server en Windows Client	179
Installatie domeincontroller	179
Client toevoegen aan het domein via PowerShell	182
Een tweede domeincontroller in het netwerk	186
125 WSCore snel configureren	189
Op WSCore werken als domeinadministrator	189
126 DHCP instellen op DC	189
Ga na of de DHCP-rol goed werkt	190
DHCP-redundantie	190
Ga na of DHCP-failover werkt	191
127 DNS instellen op DC	191
128 Replicatie forceren	192

129	Active Directory via PowerShell	193
	OU-aanmaak	193
	Groep maken	194
	Gebruikers toevoegen	194
	Organisatiegericht nadenken	195
	Testen en terug naar de originele omgeving	196
	Active Directory opbouwen via .csv-bestanden	198
130	PowerShell grafisch of via de CLI benaderen	199
131	Einde automatisering in Windows Server	201
132	Microsoft 365 met het netwerk verbinden	201
	Microsoft 365-tenant opzetten	202
	Stappenplan (best uitvoeren in incognito-browser)	202
	Verbinding tussen Windows Server en de tenant	202
	Azure vanuit PowerShell benaderen	204
	Nieuwe gebruiker in Microsoft 365-portaal toevoegen	206
	Licentie aan gebruikers toewijzen	206
	Beveiligingsgroep in Microsoft 365 toevoegen	208
	Microsoft 365-groep in Microsoft 365 toevoegen	208
	SharePoint-configuratie	210
	Teams-configuratie	211
	Introductie in Microsoft Graph	213
133	Docker	213
134	Chocolatey of Winget	216
135	Andere scripttalen	217
	Belangrijk om weten	218
136	Scripttalen combineren	218
137	Extra's installeren voor instructieverwerking	218
	Bronnen	219
	Links	219
	Boeken	219
	Schoonmaak	219

Deel 5: PowerShell-bouwstenen **221**

138	Scripts en modules	222
139	Scripts	222
140	Functies leiden vaak tot een start	222
141	Van script naar module	223
142	Voorbeeld van een module	223
143	Voorbeeld van een volledige module	224
144	Scripts, modules en functies opbouwen	226
	Requires-statement	226
	Using-statement	226
	Commentaar	226

145	Hetzelfde doen in Visual Studio Code	229
146	(Geavanceerde) functies	229
	Beter begrijpen van begin-, proces- en eindblok	232
	Return	233
	Optionele optie: CmdletBinding	233
	Optionele optie: Alias	234
	Optionele optie: OutputType	234
	Optionele optie: Parameters	234
147	Modulemanifest	234
148	Module met meer dan één functie opslaan	235
149	Fouten opvangen	236
150	Validatie	238
151	Try, catch en finally	238
152	Werken met validatieattributen	239
153	Mocking	240
	Opdracht	240
154	Debuggen	241
	Breakpoints op een regel plaatsen	241
	Breakpoint plaatsen via de CLI	247
	Extra mogelijkheden in Visual Studio Code	249
	Bronnen	251
	Links	251
	Boeken	251
Deel 6: Extra's via PowerShell		253
155	PowerShell Desired State Configuration (DSC)	254
	Introductie tot de pushmethode	255
156	API en webrequests	257
	Omgaan met bepaalde webstatussen	257
	Webrequest gebruiken om bestanden binnen te halen	258
	Webrequestmethoden	261
	Webrequest via REST	262
	Webrequest via SOAP	264
	Authenticatie	264
157	E-mail versturen vanuit PowerShell	266
	Bronnen	267
158	Packages	267
	Find-PackageProvider	267
	Install-PackageProvider	268
	Werken met PackageSource	268
	Package installeren	269
	Packages weergeven	269
	Aan de slag met MailKit	269
	Bronnen	271

159	Pester	271
	Test met fouten	273
	Describe en Context	273
	Houd de test gescheiden	273
	Om mee te geven	274
	Bronnen	274
160	Theoretisch: workflow	274
	Een voorbeeld	275
	Parallele uitvoeringen	275
	Conclusie	276
	Bronnen	276
161	Zeer kort theoretisch: Microsoft Azure Automation	276
	Bronnen	277
162	Azure Cloud Shell	277
	Bronnen	278
163	Windows Terminal en subsystem Linux	278
	Terminal versus console versus shell	278
	Windows Terminal	279
	Subsystem Linux	280
	Linux-shell gebruiken in Windows Terminal	282
	Bronnen	283
164	PSScriptAnalyzer	283
	Get-ScriptAnalyzerRule	285
	Invoke-Formatter	285
	Invoke-ScriptAnalyzer	285
165	Unified Write Filter (UWF)	286
	Installatie UWF	287
166	Werken met items: snelle uitleg	289
	Bronnen	289
167	&-operator	289
168	\$lastexitcode	290
169	Tips om goede formats weer te geven	290
170	Wachten totdat ...	291
171	Een PowerShell-script aan een GPO koppelen	291
172	PowerShell-scripts via Intune (Mobile Device Management)	292
173	PowerShell-script koppelen aan Taakplanner	293
174	Self-signed script met eigen certificaat maken	295
	Testen op een client	298
	Bronnen	299

175	Nieuw in PowerShell 5.1, 6 en 7	300
	Algemeen	300
	& als pipelineoperator	300
	Variabelen toegevoegd: \$IsWindows, \$IsMacOs en \$IsLinux	300
	Toevoeging aan contextmenu	300
	Terugkeren naar vorige locatie in de CLI	300
	Snel een if-statement testen	301
	Null-conditie testen	301
	Tab-completion soms mogelijk na =	301
	Zoeken binnen inhoud via PowerShell	301
	Eén, meerdere of geen instructie(s) uitvoeren	301
	Werken met experimentele mogelijkheden	302
	Duidelijkere foutweergave	302
	ForEach-Object met parameter -parallel	302
	Laatste vernieuwingen	303
	Bronnen	303
176	Einde	304
177	Verdere verdieping	304
	Aanbevolen boeken	304
	Aanbevolen links	304
	Aan te raden	304
Index		305

Wegwijs in PowerShell

1

Deel
1

Introductie

PowerShell is een objectgeoriënteerde shell- en scripttaal voor Microsoft Windows met als doel taken te automatiseren. Stel dat een zelfde taak 200 keer per jaar moet worden uitgevoerd en dat die taak tien minuten kost. Wat is er dan beter? Ieder jaar die 200 taken handmatig uitvoeren of even tijd stoppen in een script om de taak te automatiseren? Dit script moet zowel voor mens als pc te begrijpen zijn en net hiervoor dient PowerShell. Het script omvat een uitlijning hoe een taak perfect moet worden uitgevoerd. Dit betekent dat die taak steeds opnieuw op dezelfde manier wordt uitgevoerd en dus veiliger is dan dat de mens dit handmatig zou doen.

PowerShell stamt uit 2006 en er zijn twee verschillende versies: 5.1 en 7.X. Deze twee versies slaan op de klassieke Windows PowerShell (5.1) en PowerShell Core (7.X). Deze laatste zal in de toekomst vooral gebruikt en als enige uitgebreid/vernieuwd worden, terwijl de klassieke enkel nog aanpassingen zal krijgen voor veiligheid (bijvoorbeeld kwetsbaarheden oplossen). Sommige oudere commando's zijn in de huidige PowerShell niet meer mogelijk, maar de basisprincipes van PowerShell zijn zo goed als onveranderd sinds versie 2.

De klassieke PowerShell is gebaseerd op het .NET-Framework en PowerShell Core op .NET Core, waardoor beide steeds up-to-date moeten zijn.

.NET is een framework ontwikkeld door Microsoft en biedt een ontwikkelaarsplatform voor applicaties met bibliotheken en tools waarvan ontwikkelaars gebruik kunnen maken om services makkelijker, sneller en betrouwbaarder te gebruiken. .NET Framework is de oudste en is closed source. Het bestaat uit:

- de CLR (Common Language Runtime) die gebruikt wordt om apps uit te voeren en het geheugengebruik in de gaten te houden;
- een grote bibliotheek met betrouwbare, te hergebruiken code voor ontwikkelaars.

Omdat het closed source is, heeft Microsoft .NET Core opgericht als opensourcevariant, waardoor het veel meer interesse wekt bij ontwikkelaars en eveneens omdat het cross-platform gebruikt kan worden over meerdere besturingssystemen heen (Windows, Linux, Mac). Alles wordt ingezet op .NET Core, waardoor deze versie de nieuwste, maar ook de beste features van het .NET-framework moet omvatten.

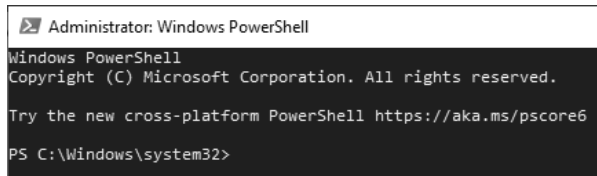
PowerShell maakt gebruik van cmdlets (commandlets). De maker, Jeffrey Snover, heeft de term cmdlet in het leven geroepen zodat via een zoekmachine (bijvoorbeeld Google) steeds relevante PowerShell-informatie terug wordt gegeven.

PowerShell is zo ontworpen dat het makkelijk is om aan te leren en ook eenvoudig is om scripts te lezen vanwege het gebruik van cmdlets. Waar zit dan de valkuil? PowerShell kan in héél veel omgevingen gebruikt worden, bijvoorbeeld Azure en Microsoft 365. Het is net in die omgevingen dat PowerShell moeilijker wordt om te gebruiken en te begrijpen.

De cmdlets die in PowerShell gebruikt worden, hebben vaak een duidelijke naam die doet wat de naam zegt te doen. Met de komst van PowerShell Core zal een zelfde cmdlet tot dezelfde

uitvoer komen, maar waarschijnlijk op een iets andere manier, afhankelijk van het besturings-systeem. Die iets andere manier maakt voor de gebruiker niet uit, aangezien de uitvoer hetzelfde blijft.

PowerShell is heel krachtig en kan zelfs meer dan wat handmatig met een pc ook zou kunnen. Het is belangrijk om PowerShell steeds als administrator te starten. Klik met de rechtermuis-knop en kies **Als administrator uitvoeren**. Bij het openen wordt het woord Administrator in de titelbalk getoond. Sinds Windows 11 heeft PowerShell een zwarte achtergrondkleur.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32>
```

PS staat voor PowerShell. Huidige locatie is hier C:\WINDOWS\system32.

De bestandslocatie van Windows PowerShell op een 64-bits pc:

- x86: %SystemRoot%\syswow64\WindowsPowerShell\v1.0
- x64: %SystemRoot%\System32\WindowsPowerShell\v1.0

De Windows-systeemmap syswow64 dient voor 32-bits systeembestanden.



Waarom 32- en 64-bits?

Beide versies zijn nodig. Stel dat een 32-bits besturings-systeem aangestuurd moet worden, is het belangrijk te werken met de 32-bits (x86)-versie. Het is zelfs noodzakelijk om na te gaan als een script niet werkt op een 32-bits systeem of dit niet ligt aan de gebruikte instructie. In een instructie worden cmdlets en commando's toegepast en het zou kunnen dat deze niet uit te voeren zijn via 32-bits.

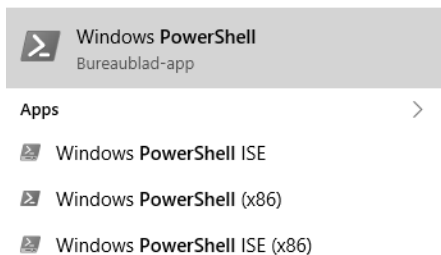
Gelukkig wordt 32-bits op moderne systemen steeds minder gebruikt en zijn er maar heel weinig instructies die niet zullen slagen om uitgevoerd te worden op een 32-bits systeem, maar het is wel belangrijk hier rekening mee te houden als deze situatie zich voordoet.

Een voorbeeld is het cloudmanagementplatform voor Mobile Device Management (MDM) van Microsoft: Intune. Ondernemingen kunnen Intune gebruiken om onder andere alle Windows-systemen te beheren en zo ook PowerShell-scripts te activeren die moeten worden uitgevoerd. Hiertoe wordt op ieder toestel Intune Management Extension geïnstalleerd, wat een 32-bits programma is. Ondanks dat dit een zo goed als nieuw platform is, worden scripts standaard als 32-bits uitgevoerd. Gelukkig bestaan er verschillende 64-bits workarounds (zolang Microsoft een en ander nog niet heeft opgelost).

Bron: z-nerd.com/blog/2020/03/31-intune-win32-apps-powershell-script-installer/
Bron: docs.microsoft.com/en-us/mem/intune/apps/intune-management-extension

Via het menu Start wordt in Windows standaard vier maal PowerShell (twee keer 64-bits en twee keer 32-bits) gevonden met twee kerncomponenten:

- Windows PowerShell – CLI – Command Line Interface (= Shell)
- Windows PowerShell ISE – GUI – Graphical User Interface (= IDE-editor)



Naast deze zijn er ook door andere firma's uitgebrachte PowerShell-editors, waarvan één eveneens door Microsoft gemaakt: Visual Studio Code. Meer zelfs: Microsoft dwingt iedereen over te stappen naar Visual Studio Code, omdat de actieve ondersteuning voor PowerShell ISE is weggefallen en Visual Studio Code cross-platform werkt.

Surf naar de website die aangegeven staat bij het openen van PowerShell om PowerShell Core te installeren. Op deze website vind je de nodige info om PowerShell Core te installeren, maar ook om oudere PowerShell-versies voor Windows te installeren.

Zo wordt geleerd dat PowerShell sinds Windows 7 standaard mee geïnstalleerd wordt via een WMF-installatiepakket bij de Windows-installatie. WMF staat voor Windows Management Framework en bevat allerlei bronnen die als systeembeheerder geraadpleegd kunnen worden, zoals PowerShell, WinRM voor CLI op afstand, WMI- en CIM-bevragingen om onder meer hardware aan te spreken enzovoort.

De installatie van PowerShell Core en het gebruik van Visual Studio Code komt pas later aan bod. Eerst wordt alles uitgevoerd in Windows PowerShell 5.1 en Windows PowerShell ISE.

Voer uit:

```
$psversiontable
```

Deze uitvoer vertelt welke versie en versienummer gebruikt wordt.



Opmerking

Onderzoek wijst uit dat beginners en experts in PowerShell ongeveer dezelfde kennis hebben in theorie, maar in praktijk kunnen experts veel sneller problemen oplossen.

Opmerkingen bij de oefeningen (virtualisatiehulp op volgende pagina)

Maak twee nieuwe VM's (client en server) in VirtualBox of VMWare. Hyper-V kan ook, maar wordt in dit boek niet uitgelegd. De Evaluation-versies worden gebruikt die Windows gratis aanbiedt:

- Windows 10 Enterprise (wordt nog gebruikt om virtualisatieproblemen te vermijden)
- Windows Server (zoals 2022) – Standard Evaluation Desktop Experience
- Nieuwere versies kunnen ook gebruikt worden.

Downloaden kan vanaf microsoft.com/en-us/evalcenter. Kies altijd het Engelstalige .ISO-bestand en zorg dat tijdens downloaden en de installatie de internetverbinding actief blijft!

De VM's bevatten elk 50 GB (dynamisch) aan opslagruimte en de NAT-NIC-instelling. De server bevat 4 GB RAM met twee virtuele processorkernen en de client 2 GB RAM. Op beide machines mogen de additions/VMWare Tools geïnstalleerd worden. Bij beide is de taal Engels (US). Time- & Currency en de keyboard-optie wordt volgens wat de gebruiker standaard gebruikt, ingesteld.

Bij de twee installaties wordt **Custom** gekozen en de volledige schijfruimte van 50 GB.

Het adminaccount op de server krijgt het wachtwoord Server* net na installatie en bij de blauwe balk van Network wordt **Yes** geselecteerd, zodat Network discovery in orde is. Na installatie wordt bij de client de gebruikersregio gekozen met de gepaste Keyboard-layout. Vervolgens: **domain join instead** linksonder waarbij gebruiker `ict` wordt aanmaakt met wachtwoord Server*. Beantwoord de drie vragen met willekeurig lange waarden voor de veiligheid.



Server*

Het wachtwoord Server* zal tijdens de gids vaak aan bod komen. Als opmerking hierbij kan gegeven worden dat * geen goed teken is om te gebruiken, omdat * ook als joker-teken dienstdoet, waardoor een onderliggend systeem dit teken verkeerd kan interpreteren. Maar hedendaagse systemen moeten hiermee om kunnen gaan en Server* heeft twee voordelen: het is snel te typen en makkelijk te onthouden als wachtwoord voor testscenario's.

Vervolgens wordt op alle vragen de tweede keuze gekozen (meestal **No**). De installaties zijn dan voltooid! Maar eerst: sluit beide machines af en maak een snapshot! Installeer hierna de VM-additions/VMWare Tools (typical) op beide VM's, sluit opnieuw af en maak snapshots.

- PowerShell en PowerShell ISE worden altijd gestart als administrator.
- Indien een pc-naam, gebruikersnaam enzovoort getypt moet worden, wordt verwacht dat dit een logisch (= bestaand) item is. In dit boek wordt dat dan onderstreept weergegeven, bijvoorbeeld: pcnaam. Het is ook mogelijk dat een stuk *cursief* staat, wat wijst op optionele invoer. Bijvoorbeeld: -credential werkgroep/gebruikersnaam
- Alle oefeningen kunnen op de client of de server worden uitgevoerd, tenzij anders vermeld.
- Alle bestanden worden bewaard in de mappen aangegeven tijdens de oefeningen.

- Als iets uitgevoerd moet worden in PowerShell Core, dan zal letterlijk PowerShell Core in de opdracht staan.

Oracle VirtualBox en VMWare: Oracle VirtualBox


Download deze software vanaf **virtualbox.org**. Virtualbox is de ideale gratis virtualisatiesoftware om testen uit te voeren via virtuele machines. Bij deze versie kan makkelijk gebruik worden gemaakt van snapshots, waardoor de toestand van een VM (virtuele machine) kan worden opgeslagen.

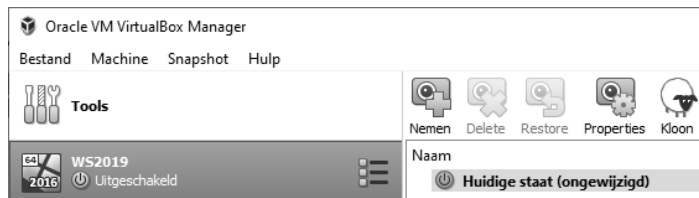
Na installatie van VirtualBox, wordt ook de VM VirtualBox Extension Pack geïnstalleerd (eveneens te downloaden vanaf **virtualbox.org**), wat nog meer opties toevoegt aan het reeds geïnstalleerde pakket. Bij een update of andere versie-installatie van VirtualBox moet de Extension Pack opnieuw gedownload en geïnstalleerd worden naar de gewijzigde versie.

Bij het eerste maal openen van VirtualBox: klik op **Bestand, Voorkeuren** en kies de standaardmap waar de virtuele machines opgeslagen moeten worden (bij voorkeur op een SSD).

Na het installeren van een besturingssysteem op een VM, kunnen de Guest Additions toegevoegd worden aan de VM. Deze bieden tal van extra mogelijkheden voor de VM (guest) als onderdeel van de fysieke pc (host).

Werken met snapshots (momentopnames van de virtuele machine) wordt als volgt gedaan:

Klik op de knop  en kies **Snapshots**. Vervolgens worden andere menuknoppen getoond.

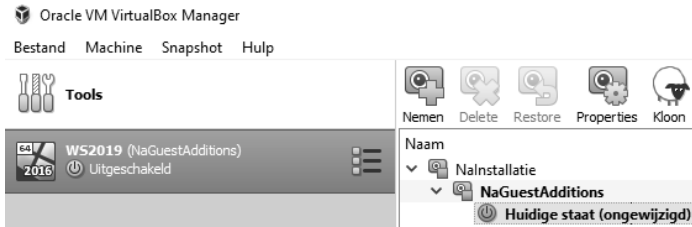


Huidige staat is geselecteerd. Klik op Nemen waarbij de snapshot NaInstallatie wordt genoemd.



De snapshot noemen we NaInstallatie.

Vanaf nu kan gekozen worden de volgende instructies mee te doen of te begrijpen wat gebeurt aan de hand van de afbeeldingen. Klik op **Nemen** voor een snapshot van de huidige staat.

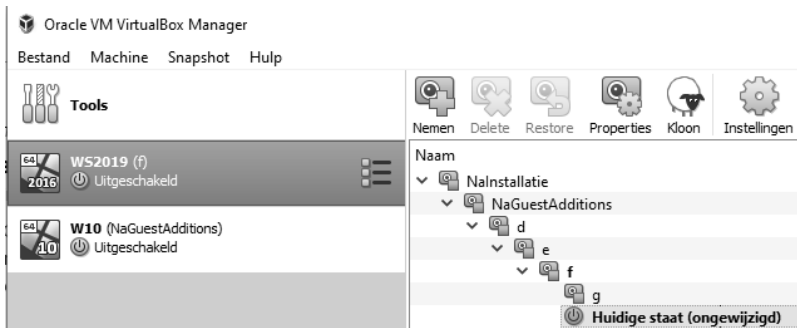


Na GuestAdditions te hebben geïnstalleerd, wordt opnieuw een snapshot van de huidige staat gemaakt. Voordat een nieuwe snapshot wordt gemaakt, is het verstandig de VM te herstarten.



Om nu alle mogelijkheden uit te leggen, is er per snapshot één tekstbestand extra op het VM-bureaublad geplaatst met als titel de letter die hetzelfde is als de snapshotnaam.

De huidige staat bevat dus momenteel een bureaublad met tekstbestanden d, e, f en g.txt. Klik met de rechtermuisknop op snapshot f en klik op **Restore**. Het volgende gebeurt:



Snapshot g bestaat nog, maar de huidige staat is aangepast naar de momentopname van snapshot f.

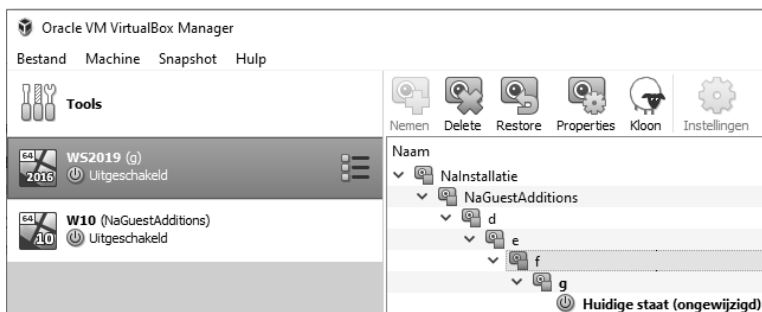
Deel 1 – Wegwijs in PowerShell

Snapshot g bestaat nog, maar de huidige staat is aangepast naar de momentopname van snapshot f. Na het starten vanaf de huidige staat, is tekstbestand g.txt verdwenen vanaf het VM-bureaublad. De VM wordt opnieuw afgesloten.

Er wordt nu hersteld vanaf snapshot g. Er wordt gevraagd of er nog eerst een snapshot genomen moet worden van de huidige staat (dit kan, maar is nu niet nodig).



Schakel dit vinkje uit.

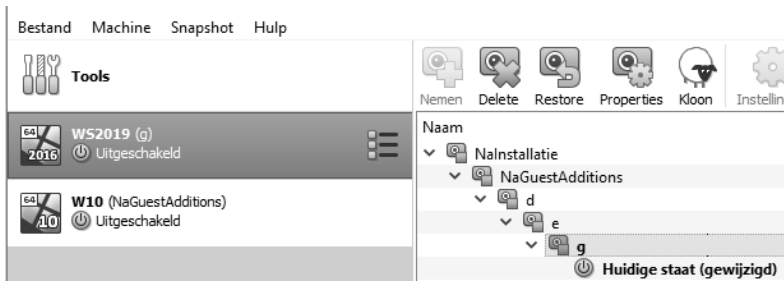


Bij het openen van de nieuwe huidige staat, staat g.txt opnieuw op het VM-bureaublad.

Nu wordt gekozen om snapshot f te verwijderen. Klik op **Verwijderen**. Eventueel kan het lang duren voordat het verwijderen is voltooid. In dit geval is dit niet zo, aangezien de snapshots amper iets bevatten.



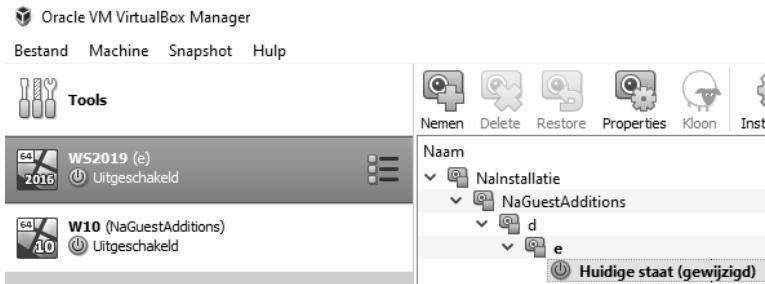
Klik op Verwijderen.



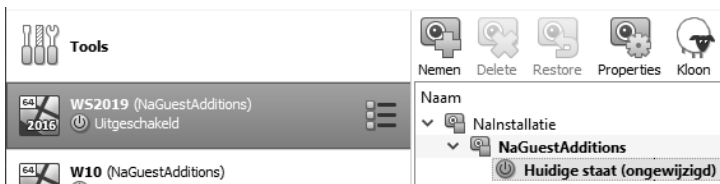
Snapshot *f* is nu verdwenen.

De huidige staat blijft hetzelfde: alle .txt-bestanden staan nog op het VM-bureaublad.

Stel dat nu snapshot *g* wordt verwijderd, dan blijft de huidige staat hetzelfde met alle .txt-bestanden op het VM-bureaublad, tot wanneer een restore naar een eerdere toestand heeft plaatsgevonden.



Nu is het tijd om terug te keren naar de oorspronkelijke staat: herstel vanaf NaGuestAdditions en verwijder daarna de overgebleven snapshots *d* en *e*, indien dit mee uitgevoerd werd.



De VM opnieuw starten kan enkel en alleen vanaf de huidige staat! Bij opstart zal gemerkt worden dat geen van de tekstbestanden nog aanwezig is.

Het kan handig zijn om eerst de volledige VM-map te back-uppen, voor terug te keren naar een bepaalde situatie. Ga hiervoor in Verkenner naar de locatie van de VM.

VirtualBox heeft soms volgende eigenaardigheden:

- na een tijd beginnen de VM's steeds trager te werken;
- na een update van VirtualBox werken de VM's die al bestonden hallucinant traag.

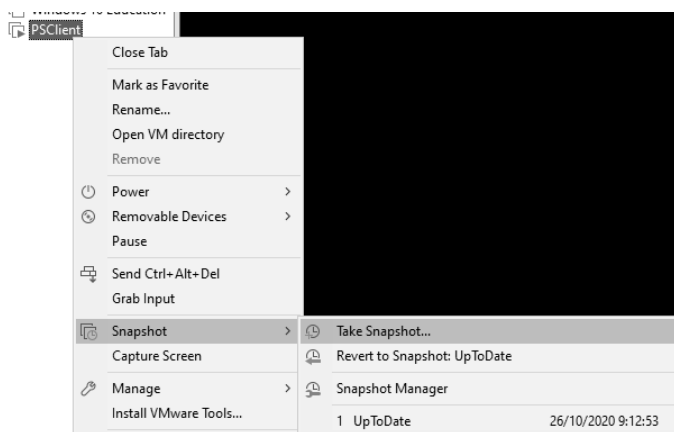
Hopelijk zijn deze problemen met de nieuwere versies van VirtualBox verleden tijd. Eventueel kan het volgende worden toegepast: verwijder VirtualBox zonder de VM's en installeer de meest recente versie opnieuw. Daarna kun je een nieuwe VM maken en de virtuele vaste schijf van de oude VM hieraan koppelen.

Oracle VirtualBox en VMWare: VMWare

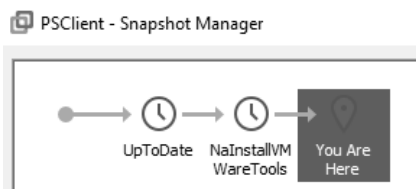
Naast VirtualBox is ook VMWare bekende virtualisatiesoftware. De gratis versie hiervan heet VMWare Workstation Player maar heeft geen snapshotfunctie aan boord. Om virtuele apparaten zoals een vaste schijf toe te voegen, moet de gebruiker administrator zijn.

Daarom wordt voorgesteld om te werken met VMWare indien VMWare Workstation Pro gebruikt kan worden. Met deze versie is het wel mogelijk om met snapshots te werken. Het kan namelijk goed mogelijk zijn dat teruggekeerd moet worden naar een eerdere toestand omdat een PowerShell-instructie de staat van het besturingssysteem heeft beschadigd.

Voor VMWare komt het gebruik van snapshots op dezelfde manier neer. Klik met de rechtermuisknop op de naam van de virtuele machine en kies **Snapshot**.

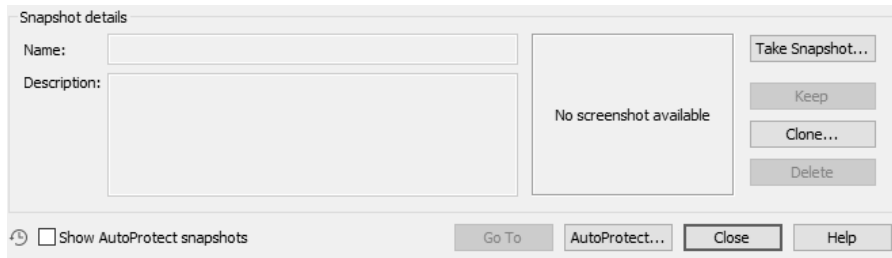


Hier kan een snapshot genomen worden of kan via de Snapshot Manager teruggekeerd worden naar een eerdere snapshot.



Snapshot Manager.

Onderaan staan enkele knoppen die zeer interessant zijn.



Opties voor snapshots.

- **Clone** Letterlijk een kopie maken vanaf een toestand.
- **Go To** Als geklikt wordt op een eerdere snapshot, is dit de knop om terug te keren naar die snapshottoestand.
- **AutoProtect** Dit is een functie die dagelijks het zelf ingestelde aantal snapshots automatisch aanmaakt. Een soort van back-up, die standaard niet wordt geactiveerd.

Bij het maken van een VM moeten enkele keuzes worden gemaakt:

- Eerst wordt de Windows-versie gekozen voor installatie. Staat hier geen 64-bits bij, dan moet waarschijnlijk nog de hardwarevirtualisatie aangevinkt worden in de BIOS- of UEFI-instellingen van de CPU bij het booten van de host (het fysieke toestel).
- Bij de nieuwste Windows-versies staat Hyper-V standaard uit, maar controleer dit ook even op het fysieke toestel: **Configuratiescherm** (kleine pictogrammen), **Programma's en onderdelen**, **Windows-onderdelen in- of uitschakelen** of **Instellingen, Apps, Optionele onderdelen, Meer Windows-functies** (onderaan). Open alle plusjes bij Hyper-V en kijk na of er geen enkele is aangevinkt. Mocht VMware op de pc geïnstalleerd zijn, dan is dit normaal geen enkel probleem om VirtualBox te gebruiken en omgekeerd.
- Daarna moet de keuze in RAM gemaakt worden.
- Vervolgens moeten de keuzes betreffende de schijf worden gemaakt. Kies voor dynamische toewijzing. Dit betekent dat de schijfruimte op de fysieke schijf enkel dat zal innemen wat op de virtuele schijf is gebruikt. Stel dat dit 16 GB is na installatie van Windows op de VM, dan wordt er op de fysieke schijf maar 16 GB gebruikt. Via de andere optie wordt meteen gekozen om 50 GB van de fysieke schijf te benutten.

Klik na het maken van de VM op de naam ervan en kies Instellingen, waarbij volgende zaken worden gecontroleerd (de meeste instellingen kunnen ook gewijzigd worden terwijl de VM in gebruik is):

- Instellen RAM.
- Processor: eventueel meer kernen geven, dit kan opgezocht worden via Windows Taakbeheer, Prestaties en bekijk hoeveel kernen er gebruikt worden (op de tab Processor). Deel het aantal kernen door 2 om hierop af te stemmen. Bijvoorbeeld vier kernen betekent maximaal twee virtuele CPU's uitdelen aan de VM.

- Voeg het .iso-bestand toe om te koppelen zodat het besturingssysteem wordt geïnstalleerd.
- Netwerk inschakelen op NAT of NAT-netwerk: deze optie zorgt ervoor dat de virtuele netwerkkaart een IP-adres zal krijgen van de zogeheten virtuele NAT-router, waardoor de VM zich niet in hetzelfde netwerk zal bevinden als de fysieke pc. Veiligheidsbewustzijn! Via een bridge zal de VM zich wel in het fysieke netwerk bevinden.

Na dit te lezen en uit te voeren, staan de virtuele machines klaar om mee aan de slag te gaan! Na installatie van Windows Server wordt gevraagd om Ctrl+Alt+Del in te drukken:

- In VirtualBox is dit op te lossen door enkel de rechter-Ctrl-toets+Del in te drukken. De rechter-Ctrl-toets is de hosttoets. Deze moet bijvoorbeeld ook ingedrukt worden als de cursor niet buiten de VM geraakt of om volledig beeld te geven: Ctrl+F.
- In VMWare is dit op te lossen door in de menubalk **VM** te kiezen en vervolgens **Send Ctrl+Alt+Del**. Om te schakelen tussen host- en guest-OS wordt gebruikgemaakt van linker-Ctrl+Alt.

Het is aan te raden om vanaf nu alle oefeningen in de virtuele machines te maken, zodat het fysieke toestel geen schade lijdt. Hiertoe kun je de bestanden met instructieregels uit de bestandsmap gebruiken. Bij VirtualBox kies je op elke VM **Apparaten, Drag-and-drop, Bidirectioneel** in het menu. Dit maakt het slepen van bestanden en mappen mogelijk. Kopiëren en plakken werkt vaak niet tussen fysieke en virtuele machine; vandaar slepen.

1 Aan de slag

Start PowerShell niet als administrator op de client en server. De huidige locaties zijn:

- Client: PS C:\users\ict>
- Server: PS C:\Windows\system32> of PS C:\Users\Administrator>

Typ `exit` in beide CLI-schermen en start PowerShell opnieuw als administrator. De locaties zijn nu:

- Client: PS C:\Windows\system32>
- Server: PS C:\Users\Administrator>

Typ `exit` in de CLI van de server en sluit de server volledig af, omdat de server pas later nodig is. De instructies die op de client worden uitgevoerd, zijn ook van toepassing op de server.

Voer vanaf nu alles uit op de client. Voer de cmdlet `get-childitem` uit. Deze cmdlet toont alle onderliggende mappen en bestanden.

Typ `powershell`, waardoor een nieuwe Windows PowerShell-sessie start in hetzelfde PowerShellvenster. Dit kan handig zijn om een volledige nieuwe sessie te starten.

Typ tweemaal `exit`, waardoor beide sessies in Windows PowerShell afsluiten.

2 Eerste instructie uitvoeren

Met instructie wordt het uitvoeren van een (reeks) opdracht(en) bedoeld vanaf een PowerShell-sessie. Binnen een opdracht kunnen één of meerdere commando's gebruikt worden. Zoals reeds bekend, wordt in PowerShell de naam cmdlet gebruikt in plaats van commando indien het gaat om een PowerShell-commando of -functie.

Open PowerShell op de client als administrator en voer uit:

```
rename-computer -newname "psclientInitialen"
```

Merk op dat gevraagd wordt om de pc opnieuw op te starten. Voer uit:

```
restart-computer
```

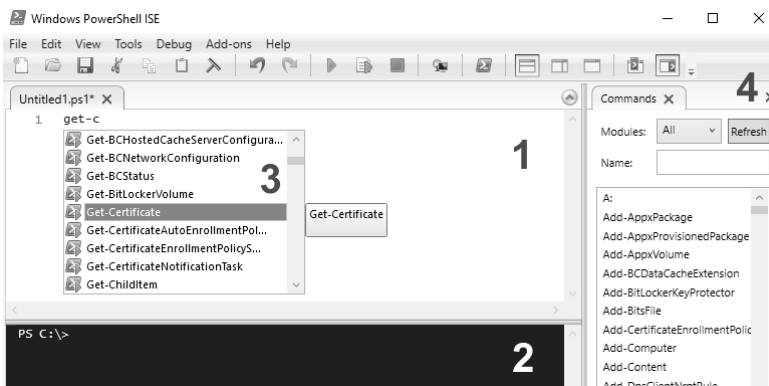
Controleer na de herstart zowel grafisch als in PowerShell of de naamswijziging is doorgevoerd. Start PowerShell en typ de globale variabele:

```
$env:computername
```

3 Gebruikmaken van Windows PowerShell ISE

Meestal wordt de CLI geopend voor het gebruik van makkelijke cmdlets of het uitvoeren van werkende scripts. In de ontwikkeling van Windows PowerShell werd de vraag steeds groter naar een IDE (Integrated Development Environment) om zelf gebruiksvriendelijke scripts te creëren. Dit werd mogelijk via de implementatie van Windows PowerShell ISE (Integrated Scripting Environment).

Windows PowerShell ISE bestaat hoofdzakelijk uit een scriptpaneel (1) en een consolepaneel (2), die onder meer gebruikmaken van IntelliSense (3; automatisch suggesties aanbieden wat bij de code kan horen) en een commandview (4), waardoor cmdlets snel en correct gevormd kunnen worden. De weergave kan aangepast worden via het menu **View** in de menubalk.

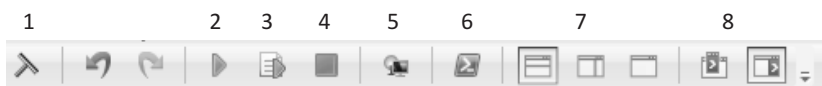


Windows PowerShell ISE.

Daarnaast toont de ISE ook kleuren om een cmdlet, functie, variabele of ander type tekst aan te geven. Er kan meteen gestart worden met typen in ofwel het script- of het consolepaneel. Het verschil ligt hem in het gebruik van het aantal regels.

In het scriptpaneel kunnen enkele regels code onder elkaar geplaatst worden zonder uit te voeren. Druk, wanneer gereed, op F5 of op de knop **Run Script** en de output verschijnt in het consolepaneel.

Indien de uitvoer per regel moet worden getoond, wordt gebruikgemaakt van het consolepaneel, waarbij een regel getypt wordt en steeds bevestigd wordt met Enter. Dit komt op hetzelfde neer als gebruik van de CLI, met als voordeel het gebruik van IntelliSense.



Werkbalk.

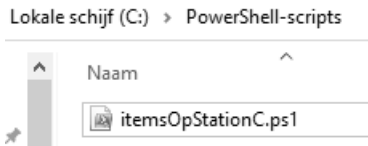
Knoppen op de werkbalk:

- 1 Maak consolepaneel leeg
- 2 Run Script (F5)
- 3 Run Selection (F8): voer geselecteerde regels uit
- 4 Stop Operation (Ctrl+spatie)
- 5 New Remote PowerShell Tab: activeer remote sessie met andere client
- 6 Start PowerShell.exe: Start CLI
- 7 Start command view in nieuw venster
- 8 Toon command view rechts in hetzelfde venster

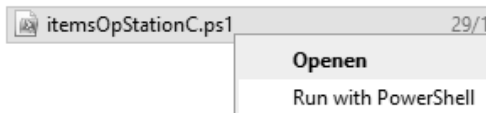
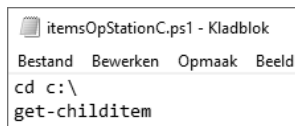
Oefening

Start Windows PowerShell ISE en zorg dat het script- en consolepaneel en de command view zichtbaar zijn. Eventueel wordt op **View** geklikt om bepaalde vensters te tonen en/of rechtsboven om het scriptvenster `Script` open te klappen.

- 1 Typ in het scriptpaneel de tekst `cd c:\` en druk op Enter.
- 2 Typ de volgende regel: `get-childitem`.
- 3 Voer uit via F5 of de knop **Run Script** en bekijk de uitvoer in het consolepaneel.
- 4 Typ `exit` in het consolepaneel, waarbij gevraagd wordt om het script op te slaan als een .ps1-bestand. Klik op **Ja**, maak een nieuwe map `PowerShell-scripts` op het C:\-station en sla het bestand op met de naam `itemsOpStationC.ps1`.
- 5 Ga naar de map `PowerShell-scripts` en dubbelklik op `itemsOpStationC.ps1`, waarbij Kladblok opent. Tip: kijk ook even na of de .ps1-extensie zichtbaar is. Zo niet, maak ze dan zichtbaar (**Beeld, Bestandsnaamextensies of View, Extensions**).



PowerShell is standaard beveiligd tegen het openen van bestanden met PowerShell-extensie. En dat is maar goed ook, want het woord Power verwijst naar krachtig en PowerShell kan inderdaad met één tot enkele cmdlets het besturingssysteem flink beschadigen. In plaats van PowerShell wordt Kladblok geopend, waarbij twee regels worden getoond. Merk op dat de regel met `exit` niet getoond wordt. Klik met de rechtermuisknop op `itemsOpStationC.ps1` en voer uit in PowerShell.



Indien een melding wordt getoond betreffende Execution-Policy, is dit vrij normaal. Druk op A om het script uit te voeren. Op deze Execution-Policy wordt later teruggekomen, aangezien dit voor PowerShell zeer belangrijk is. Typ `set-executionpolicy remotesigned` bij problemen.

Het script wordt in een mum van tijd in PowerShell uitgevoerd, maar de uitvoer wordt niet getoond. Dat is niet fijn. Dubbelklik opnieuw op het bestand, voeg in Kladblok als laatste regel het woord `pause` toe en sla op met `Ctrl+S`. Klik opnieuw met de rechtermuisknop op het bestand en voer uit in PowerShell. Sluit af door op `Enter` te drukken. Het eerste PowerShell-script is een feit! Tip: vervang `pause` door `start-sleep 5`.

```
Directory: C:\
Mode                LastWriteTime         Length Name
----                -
d-----          13/12/2020    18:35     DRIVERS
d-----           9/09/2020    15:52 IPCamRecord
d-----          16/07/2020     8:36     MSI
d-----           7/12/2019    10:14     PerfLogs
d-----          29/12/2020    14:28 PowerShell-scripts
d-r-----         13/12/2020    13:15 Program Files
d-r-----         18/11/2020    20:14 Program Files (x86)
d-----          22/12/2020    14:39     scripts
d-r-----         15/07/2020    18:39     Users
d-----           6/12/2020    12:28 VMPowerShell
d-----           9/09/2020    15:45 VMS_CAPTURE
d-----           9/12/2020    13:44     Windows
Press Enter to continue...: _
```

Uitvoer.

4 Visual Studio Code

Standaard wordt de IDE-omgeving Windows PowerShell ISE geïnstalleerd via een Windows-installatie. Dit gebeurt natuurlijk niet op andere besturingssystemen, waardoor een IDE voor PowerShell Core ontbreekt.

Via de installatie van Visual Studio Code wordt het mogelijk om PowerShell Extensions te activeren, waardoor Visual Studio Code gebruikt kan worden als IDE voor PowerShell op andere besturingssystemen. Belangrijk om te weten:

- Gebruik PowerShell ISE wanneer code op Windows uitgevoerd moet worden en deze nog voldoet voor PowerShell versie 5.1, maar weet dat deze ISE-versie geen optimale ondersteuning meer heeft, waardoor bekende bugs eventueel niet meer worden opgelost.
- Gebruik PowerShell ISE op Windows als een niet kant-en-klaar script speciaal voor Windows moet worden uitgevoerd, zodat fouten duidelijker aangetoond worden.
- Gebruik Visual Studio Code in alle andere gevallen.

De installatie van Visual Studio Code volgt in deel 2 van dit boek.

5 Typische PowerShell-opmerkingen

Vaak zal gebruik worden gemaakt van instructies die te maken hebben met services en processen, omdat die veel kunnen aantonen zonder de pc of de virtuele machine te beschadigen. Er zijn nog enkele aanvullende zaken die werken met PowerShell vergemakkelijken:

- PowerShell is niet hoofdlettergevoelig. De cmdlets en meer die gebruikt worden, zullen vaak met of zonder hoofdletter in de opgegeven instructies staan.
- PowerShell houdt rekening met wat de gebruiker al kent. Linux-gebruikers werken veel in een CLI (Command Line Interface) en maken daarvoor onder andere gebruik van UNIX-commando's. Windows-gebruikers die vroeger veel met CMD (Opdrachtprompt) werkten, gebruiken ook verschillende commando's.

Al deze commando's zijn in PowerShell geïmplementeerd, waardoor heel wat cmdlets alternatieven (aliassen) hebben.

- De cmdlet `get-childitem` kan bijvoorbeeld gebruikt worden door `dir` (commando in Windows voor directory) of `ls` (UNIX- en Linux-commando voor list) te typen.

Oefening

- 1 Start PowerShell en ga naar locatie C:\: `cd c:\` of `cd \` indien de C-partitie al actief is.
- 2 Voer achtereenvolgens de commando's `ls` en `dir` uit.
Indien correct uitgevoerd, wordt tweemaal dezelfde output getoond.
- 3 Typ `get-ch` en druk daarna op de tabtoets. `Get-Childitem` wordt snel getoond. *Tab-completion* is een toevoeging om sneller binnen PowerShell te werken.
Daarnaast kan als alias voor de cmdlet `get-childitem`, `gci` gebruikt worden, wat `get-childitem` afgekort is en in de lijst van aliassen is opgenomen: `get-alias gci`.

6 Houd rekening met iedereen

Start PowerShell ISE en voeg volgende code toe:

```
cd hkcu:\Software
new-item NSleutel
new-itemproperty -path NSleutel -name test -value "testwaarde"
```

Een toepasselijke naam voor dit script zou kunnen zijn: [AanmaakRegistersleutel.ps1](#).

Sla op als .ps1-bestand met deze gekozen naam en bewaar in de map "PowerShell-scripts".

Bovenstaand script zou voor de gebruiker duidelijker zijn met commentaar. Commentaar kan in PowerShell worden toegevoegd via # voor een enkele regel en <# #> voor meerdere regels.

Voeg bovenaan het volgende commentaar toe aan het script en daaronder een aparte commentaarregel:

```
<# -----
Auteur: naam
Datum: datum
Functie: registersleutel en -waarde toevoegen aan HKCU – Software
----- #>

cd hkcu:\Software
new-item NSleutel #Toevoegen registersleutel
new-itemproperty -path NSleutel -Name test -value "Testwaarde"

#waarde toevoegen aan registersleutel
```

Sla het script opnieuw op en voer uit. Voor iedere gebruiker is het nu duidelijk wat het script doet. Open Regedit en ga op zoek naar de gemaakte sleutel. Verwijder deze handmatig.

7 Escapeteken

In PowerShell wordt ` (*backtick*) gebruikt als escapeteken. Zoek deze op het toetsenbord.

Indien het teken niet gevonden wordt, kan een kopie worden gemaakt van dit teken op een website waar het gevonden wordt via de zoekterm `escape character powershell` in Google.

Een escapeteken kan later in een instructie handig zijn. Enkele voorbeelden:

- `t = tab
- `n = nieuwe regel

Bijvoorbeeld: `"Nieuwe`nregel" > c:\powershell-scripts\escape.txt`. Open het bestand en merk op dat Nieuwe op de eerste regel staat en regel op de tweede. Toch kan het zijn dat

bepaalde tekstprogramma's dit nog op één regel weergeven. Om dit op te lossen moet ``r` toegevoegd worden net voor ``n`.

Oefening

Start PowerShell en klik met de rechtermuisknop in de titelbalk. Besteed ongeveer twee minuten aan de verschillende instellingen van de shell. De shell kan bijvoorbeeld aantrekkelijker worden gemaakt. Het is bijvoorbeeld belangrijk om een goed lettertype te kiezen om het verschil te zien tussen een accent ' en het escapeteken: `.

Het escapeteken kan ook gebruikt worden om een coderegel af te breken naar een volgende regel, wat het lezen van de code soms vergemakkelijkt. Dit wordt in de voorbeelden soms toegepast, om aan te geven dat een instructie op één regel hoort.

Pas geen kleuren aan, tenzij dit echt gewenst is. Het is namelijk niet makkelijk om terug te keren naar de standaardeigenschappen van PowerShell.

Niet alle instellingen voor het uiterlijk van de PowerShell-CLI kunnen grafisch aangepast worden; een voorbeeld is de titel wijzigen van de gestarte sessie. Voer daarvoor de volgende instructie in:

```
$host.ui.RawUI.WindowTitle = "Administrator: Nieuwe sessie van Naam Voornaam"
```

8 Cmdlets

Waarom `get-childitem` gebruiken als het ook korter kan met `dir`? Jeffrey Snover, de maker van PowerShell, wilde commando's implementeren die duidelijk waren in het gebruik van het systeem. Een cmdlet wordt steeds gevormd via *verb-noun*, waarbij het werkwoord (*verb*) algemeen vertelt wat het commando zal doen en het zelfstandig naamwoord (*noun*) daarna het commando specificeert. Vaak voorkomende verbs zijn `add`, `clear`, `convertFrom`, `convertTo`, `disable`, `enable`, `export`, `get`, `set`, `new`, `remove`, `import`, `start`, `stop`, `test`, `write`. Ze allemaal zien kan via `get-verb`.

`invoke` is een andere veelgebruikte en speciale verb, die vooral wordt gebruikt bij sessies op afstand om een instructie uit te voeren. Denk bij `invoke` aan de Nederlandse vertaling "beroep doen op".

Enkele basis-cmdlets zijn `compare-object`, `sort-object`, `select-object`, `convertFrom-SecureString`, `Get-Alias`, `New-Alias`, `Get-Command`, `Get-Credential`, `Set-Date`, `Get-ExecutionPolicy`, `Get-Host`, `Set-Location`, `Stop-Process`, `Remove-PSDrive`, `Get-Service`, `Set-Variable`, `Read-Host`.

Uit bovenstaande voorbeelden kun je al snel afleiden waarvoor de verb of de cmdlet staat.

Merk op dat cmdlets altijd in het enkelvoud eindigen: bijvoorbeeld `get-service` en niet `get-services`.