

Powershell

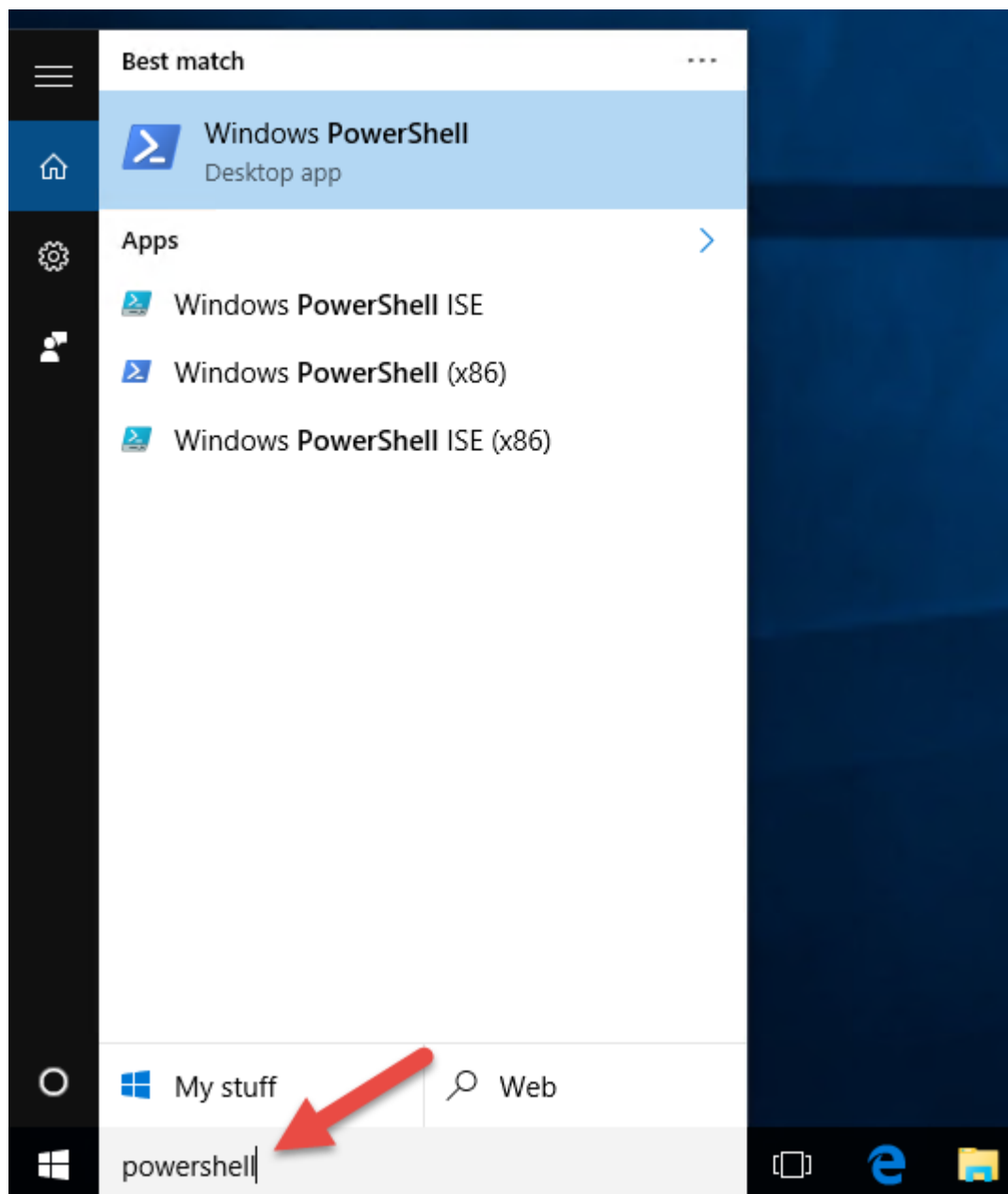
Voorwoord

Dit labo is grotendeels gebaseerd op de [officiële Powershell-docs](#), maar zeker niet alles omvattend. Bekijk ook deze documentatie voor extra info.

Wat is Powershell?

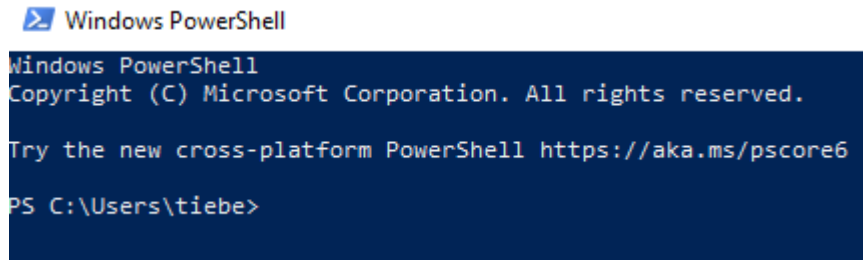
Powershell is een scriptingtaal ontwikkeld door Microsoft, oorspronkelijk voor het aansturen van Windows. Ondertussen is het uitgegroeid tot een enorm krachtige cross-platform-tool waarmee je je eigen machine, een remote machine, of zelfs een van de Microsoft Cloud-service zoals Microsoft 365, Azure, ...

Op iedere Windows-computer vind je Powershell standaard geïnstalleerd terug.



De belangrijkste app hier is Windows Powershell, dit geeft je een command-line interface, specifiek gebouwd voor Powershell. Een andere interessante app is de Windows Powershell ISE. Dit is eigenlijk een (eenvoudige) IDE, waarbij de command-line minder centraal staat, maar meer gebouwd is voor het schrijven van grotere scripts.

Wanneer je Powershell opent, krijg je de volgende interface te zien



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

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

PS C:\Users\tiebe>
```

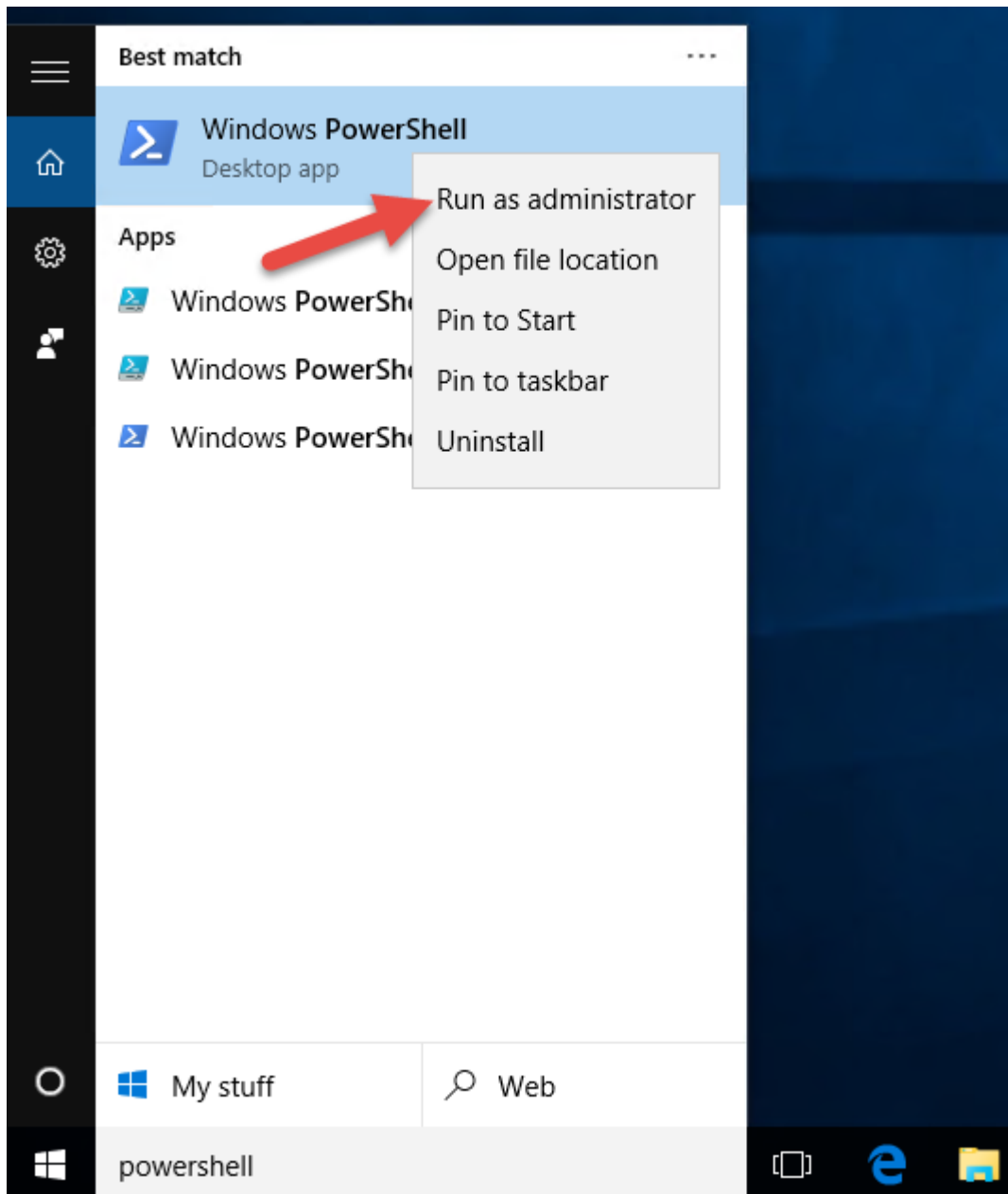
Achter dit prompt kan je commando's beginnen typen. Probeer bijvoorbeeld eens het volgende commando uit te voeren.

```
Get-Service -Name W32Time
```

Met de uitvoer van dit commando zal je meer info krijgen over de W32Time-service. Wil je deze service stoppen? Dat kan met het commando

```
Stop-Service -Name W32Time
```

Je zal echter een error krijgen, met de boodschap dat de service niet gestopt kan worden omdat de W32-Time-service niet geopend kan worden. Dit heeft als oorzaak dat we Powershell als een gewone user uitvoeren, en niet als Administrator. Sluit Powershell af en open het opnieuw door het te zoeken, met de rechtermuisknop te klikken en 'Uitvoeren als administrator' te kiezen.



Probeer nu bovenstaand commando nog eens uit te voeren, en je zal merken dat het nu wel lukt.

Opmerking

In Linux hebben we dit altijd opgelost met het commando `sudo`, wat ons toeliet om een enkel commando als een administrator uit te voeren. Er bestaan een aantal 'hacks' om iets gelijkaardig te maken in Powershell, maar tot op heden bestaat er nog geen officiële oplossing voor.

De Powershell-syntax

Powershell heeft een van de meest logisch opgebouwde syntaxes. Ieder commando (cmdlet) is opgebouwd uit twee elementen. In het Engels omschrijft Microsoft het als een verb-noun systeem. Het eerste deel van het commando (verb) beschrijft de actie die er gaat gebeuren. `New` gaat iets nieuw maken, `Get` gaat data ophalen, `Set` gaat data aanpassen, ...

Het tweede deel van het commando beschrijft waarop de actie wordt uitgevoerd. Probeer met deze informatie eens te achterhalen wat volgende commando's doen:

```
Get-Process  
Get-Command  
Remove-NetFirewallRule  
Stop-Computer
```

Opmerking

Ter info, de Linux-varianten van deze commando's zijn. We kunnen dus wel zeggen dat de syntax van Powershell iets eenvoudiger is.

```
ps aux  
compngen -ac | grep searchstr  
iptables -D INPUT 2  
shutdown now
```

Get-Help

Wanneer je niet zeker bent van hoe een Powershell-commando precies, kan je altijd gebruik maken van de Get-Help-functie.

```
Get-Help Get-Command
```

Bovenstaand commando toont in het kort alle verschillende parameters die je kan meegeven aan het **Get-Command**-commando. Als je nog niet helemaal zeker bent, kan je de parameter **-Online** meegeven. Dit opent automatisch de online documentatie.

Variabelen

Het concept variabelen zullen jullie ondertussen al wel kennen, en in Powershell is dit niet anders. Powershell is dynamical typed, dus je hoeft je geen zorgen te maken over het instantiëren van variabelen. Een variabele toekennen kan bijvoorbeeld simpelweg door

```
$srvc = Get-Service -Name wuauserv
```

Hierdoor wordt de info over wuauserv (Windows Update) in de variabele **\$srvc** gestoken. Wanneer je deze variabele simpelweg ingeeft in de shell en op enter duwt, krijg je de info terug te zien. Je kan deze variabele ook in combinatie met andere commando's gebruiken:

```
Stop-Service $srvc  
Start-Service $srvc
```

Oefening 1

Je maakt een array (variabele) met daarin enkele ip-adressen.

```
$ips = @('1.1.1.1', '8.8.8.8', '8.8.4.4', '84.198.169.35')
```

Met een ForEach-loop ga je door deze lijst en probeer je een ping naar ieder IP-adres te sturen (**Test-Connection**). Wanneer de ping succesvol is, schrijf je 'Success for IP x.x.x.x', als deze niet slaagt schrijf je 'Failure for IP x.x.x.x'. x.x.x.x vervang je uiteraard door het IP-adres dat je niet probeerde te pingen. Voor het schrijven van data naar de terminal gebruik je best **Write-Host**.

Nuttige links

[Powershell Arrays](#) Lees hier zeker ook de titel 'ForEach loop'

[Powershell if-else](#)

Oneliners & Pipelines

Een van de favoriete vormen van zelfverheerlijking voor een Systeembeheerder is het schrijven van compacte, onleesbare, ingewikkelde, en vaak nutteloze oneliners. De kunst is om zoveel mogelijk commando's aan elkaar te plakken, en data door te geven via een '|', een pipeline. Bekijk het volgende voorbeeld:

```
Get-Service |  
Where-Object CanPauseAndContinue -eq $true |  
Select-Object -Property *
```

In dit voorbeeld gaat in het eerste deel van het commando alle services opgevraagd worden. In het tweede deel wordt er gefilterd, zodat alleen nog de objecten met Property CanPauseAndContinue=true overblijven, en tenslotte wordt in het derde deel van alle objecten alle properties getoond.

Powershell remoting

Je kan via Powershell, zoals in het begin gezegd ook remote PC's aansturen. Hiervoor moet je eerst en vooral op alle computers die je Remote wil aansturen het commando **Enable-PSRemoting** voor gebruiken. Hierna kan je via **Enter-PSSession -Computername COSCIDC1** remote commando's uitvoeren op de andere computer.

Probeer dit eens op je labo-omgeving, en probeer zo de services op de remote PC op te vragen.

Oefeningen

Voor we aan de oefeningen beginnen, bekijk eerst nog eens de Powershell-oefeningen aan het einde van labo 1, en eventueel de uitleg die erbij hoort. Zorg dat je snapt wat hier allemaal gebeurt, en stel vragen waar

nodig.

Probeer ook eens de tweede oefening hier te maken (met het inlezen van een CSV-bestand). Probeer zelf eens op te zoeken hoe je deze moet inlezen (tip: **Import-CSV**)

Oefening 2

Stel voor alle gebruikers in je AD-domein de property 'Company' in op 'Cosci'.

Oefening 3

(Dit werkt enkel op de Domain-Controller in de labo-omgeving) Vraag een overzicht op van alle DNS-records onder de zone 'cosci.be'

Oefening 4

Probeer in een oneliner met Powershell remoting en de **Copy-Item**-cmdlet een bestand van je lokale machine naar een remote machine te kopiëren. Kijk hier vooral goed naar de parameters die je met **Copy-Item** kan meegeven.

Oefening 5

Vraag via de Windows Event Log alle events met een error op. EXTRA: Vraag nu ook eens alle events op van een remote-computer

Oefening 6

Geef een overzicht van alle GPO's (Group Policy Objects) op het domein cosci.be.