

Sesje – info

Tworzenie sesji

Nawiązywanie połączenia z wykorzystaniem sesji

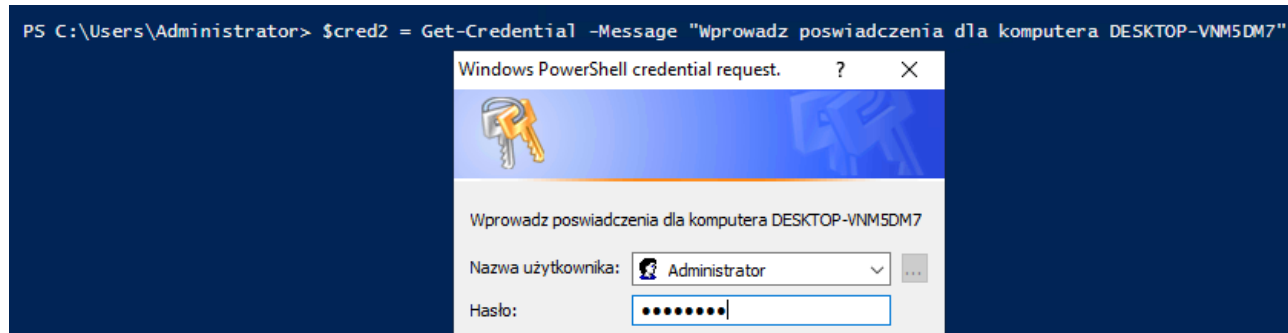
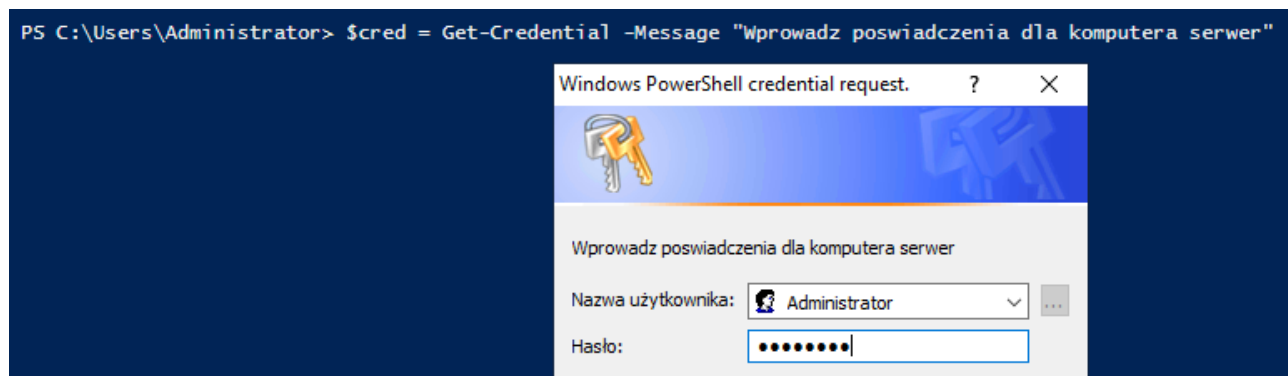
Zarządzanie sesjami

Konfigurowanie protokołu WSMAN

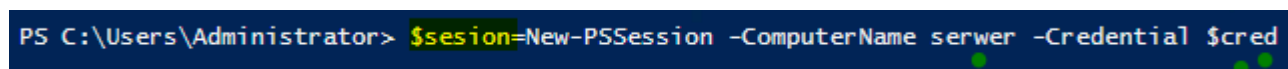
Do tej pory przykłady bazowały na poleceniu Invoke-Command

Za każdym razem, kiedy komenda ta jest uruchamiana PowerShell w tle wykonuje pracochłonne zadania związane z uwierzytelnieniem, przesłaniem poświadczeń, uruchomieniem zdalnej powłoki PowerShell w celu wykonania przesłanych poleceń, można wcześniej nawiązać połączenie i utrzymywać je w otwartym stanie dłużej, ilekroć zajdzie potrzeba uruchomienia kolejnego polecenia na systemie zdalnym można by było korzystać z już raz wcześniej utworzonego połączenia.

Aby przygotować takie stałe połączenie z komputerem zdalnym będziemy tworzyć tzw. obiekt sesji zaczniemy od pobrania poświadczeń pozwalających na korzystanie z komputerów **server** i **DESKTOP-VNM5DM7** zdalnie.



New-PSSession utworzy obiekt sesji, w wskazuje parametrach do jakiego komputera i poświadczenia



Wynik polecenia zapamiętany w zmiennej, dzięki temu wykonując kolejne komendy będziemy mieli, jak wskazać na to w której sesji to polecenie ma być wykonane.

W danej chwili można dysponować większą ilością sesji

Sprawdzam, czy sesja została utworzona i jednocześnie zawartość zmiennej `$session`

```
PS C:\Users\Administrator> $session
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
1	WinRM1	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available

Enter-PSSession pozwala na interaktywną pracę w systemie zdalnym

```
PS C:\Users\Administrator> Enter-PSSession -Session $session
```

```
[serwer]: PS C:\Users\Administrator\Documents>
```

Skorzystałem z gotowego obiektu `-Session $session`

Jestem na komputerze zdalnym

```
[serwer]: PS C:\Users\Administrator\Documents> Exit-PSSession
```

```
PS C:\Users\Administrator>
```

Rozłączyłem się

Sesja istnieje

Gdybym wcześniej zdefiniował zmienne w sesji to nich nie utracę, bo sesja istnieje a tylko ja się odłączyłem. Sprawdza czy sesja na pewno istnieje, polecenie wyświetla wszystkie obiekty sesji w danej chwili

```
PS C:\Users\Administrator> Get-PSSession
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
1	WinRM1	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available

Spróbujmy ją wykorzystać w poleceniu `Invoke-Command`

```
PS C:\Users\Administrator> Invoke-Command -Session $session -ScriptBlock { Get-WmiObject -Class Win32_operatingSystem }
```

```
SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 17763
RegisteredUser  : Użytkownik systemu Windows
SerialNumber    : 00429-00104-70082-AA484
Version        : 10.0.17763
PSComputerName  : serwer
```

Podajemy dwa argumenty sesje i polecenia do wykonania na systemie zdalnym.

Tworząc sesje można podawać jej nazwy.

Poniżej tworze sesje o nazwie „Backup session”.

Wykonując `Get-PSSession` zobaczę nazwy sesji, dzięki czemu łatwiej panować nad sesjami.

```
PS C:\Users\Administrator> New-PSSession -ComputerName serwer -Credential $cred -Name "Backup session"
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
2	Backup session	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available

```
PS C:\Users\Administrator> Get-PSSession
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
1	WinRM1	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available
2	Backup session	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available

Kolejne polecenie to `Invoke-Command` korzystające z istniejącej sesji tylko tym razem nie zapamiętałem obiektu sesji w zmiennej, dlatego po parametrze `-Session` umieszczam `(Get-PSSession -Name "Backup session")` które odnajdzie sesję po znanej nazwie `"Backup session"`

Wyszukany obiekt zostanie wykorzystany do przesyłania poleceń do systemu zdalnego

```
Invoke-Command -Session (Get-PSSession -Name "Backup session") -ScriptBlock { Get-WmiObject -Class Win32_operatingSystem }
```

```
PS C:\Users\Administrator> Invoke-Command -Session
```

```
SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 17763
RegisteredUser  : Użytkownik systemu Windows
SerialNumber    : 00429-00104-70082-AA484
Version         : 10.0.17763
PSComputerName  : serwer
```

Kiedy sesja nie będzie potrzebna to można ją usunąć

```
$session | Remove-PSSession
```

Sprawdzam czy pierwsza sesja została usunięta

```
PS C:\Users\Administrator> Get-PSSession
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
2	Backup session	serwer	RemoteMachine	Opened	Microsoft.PowerShell	Available

Tak pierwsza sesja została usunięta

Jeżeli chcesz usunąć wszystkie sesje to

```
Get-PSSession | Remove-PSSession
```

Jak widać wszystkie sesje zamknięte

```
PS C:\Users\Administrator> Get-PSSession
PS C:\Users\Administrator> |
```

Możesz konfigurować działanie sesji na komputerze docelowym np.

Są tu ustawienia dotyczące ilości pamięci dostępnej w sesji, ogólnej ilości sesji remoutingu w danej chwili. Parametry można kontrolować

```
PS C:\Users\Administrator> dir WSMan:\localhost\Shell

WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Shell

Type      Name                               SourceOfValue Value
----      -
System.String AllowRemoteShellAccess            true
System.String IdleTimeout              7200000
System.String MaxConcurrentUsers        2147483647
System.String MaxShellRunTime            2147483647
System.String MaxProcessesPerShell      2147483647
System.String MaxMemoryPerShellMB        2147483647
System.String MaxShellsPerUser          2147483647
```

Utworze ponownie sesje

```
PS C:\Users\Administrator> $session=New-PSSession -ComputerName serwer -Credential $cred
PS C:\Users\Administrator> New-PSSession -ComputerName serwer -Credential $cred -Name "Backup session"

Id Name          ComputerName ComputerType State      ConfigurationName Availability
-- --          -
20 Backup session serwer        RemoteMachine Opened     Microsoft.PowerShell Available

PS C:\Users\Administrator> Get-PSSession

Id Name          ComputerName ComputerType State      ConfigurationName Availability
-- --          -
19 WinRM19      serwer        RemoteMachine Opened     Microsoft.PowerShell Available
20 Backup session serwer        RemoteMachine Opened     Microsoft.PowerShell Available
```

Przykład sesji rozłączonej

Disconnect-PSSession na chwile przerywa sesje

```
PS C:\Users\Administrator> Disconnect-PSSession -Session $session

Id Name          ComputerName ComputerType State      ConfigurationName Availability
-- --          -
19 WinRM19      serwer        RemoteMachine Disconnected Microsoft.PowerShell None

PS C:\Users\Administrator> Get-PSSession

Id Name          ComputerName ComputerType State      ConfigurationName Availability
-- --          -
19 WinRM19      serwer        RemoteMachine Disconnected Microsoft.PowerShell None
20 Backup session serwer        RemoteMachine Opened     Microsoft.PowerShell Available
```

Zwróć uwagę na status sesji 19 (jest rozłączona)

Sesja, od której się odłączyłeś może zostać na nowo podłączona.

```
PS C:\Users\Administrator> Receive-PSSession -Session $session
PS C:\Users\Administrator> Get-PSSession

Id Name          ComputerName ComputerType State      ConfigurationName Availability
-- --          -
19 WinRM19      serwer        RemoteMachine Opened     Microsoft.PowerShell Available
20 Backup session serwer        RemoteMachine Opened     Microsoft.PowerShell Available
```

Stan sesji to obie otwarte i obie dostępne

```
PS C:\Users\Administrator> Enter-PSSession -Session $session  
[serwer]: PS C:\Users\Administrator\Documents>
```

W przypadku zamykania narzędzi klienckich PowerShell na komputerze lokalnym sesje są automatycznie zamykane.

Sesja podlega rozłączeniu bez zamknięcia tylko w przypadku użycia `Disconnect-PSSession`

Tworzenie sesji

Nawiązywanie połączenia z wykorzystaniem sesji

Zarządzanie sesjami

Konfigurowanie protokołu WSMAN