چگونه از پاورشل در ویندوز سرور 2019 برای اسکریپت‌نویسی استفاده کنیم؟

اسکریپت‌نویسی در پاورشل مشابه با فرآیند ساخت فایل‌های دسته‌ای است، اما به شکل قدرتمندتری انجام می‌شود. دستورات خط فرمان مفید هستند، اما محدودیت‌هایی دارند، در حالی که فرمان‌های پاورشل می‌توانند تغییراتی در سیستم‌عامل به وجود آورند. پاورشل اجازه می‌دهد به آیتم‌هایی که درون متغیرهای محیطی یا رجیستری قرار دارند اشاره کنیم و به سادگی فرمان‌هایی روی سامانه‌های راه دور اجرا کنیم و حتی از متغیرهای موجود در یک اسکریپت پاورشل استفاده کنیم، دقیقاً مشابه با کاری که در یک زبان برنامه‌نویسی انجام می‌دهید.

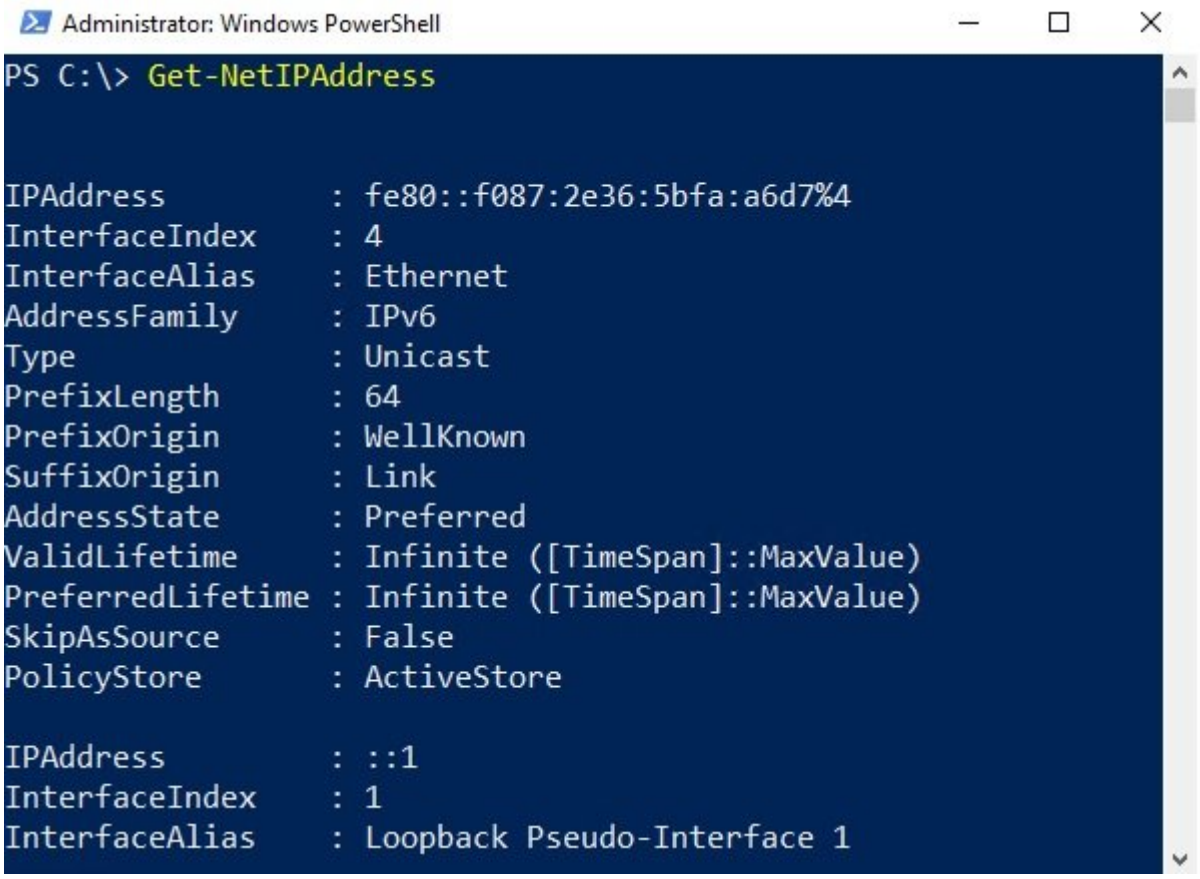
برای مطالعه قسمت قبل آموزش رایگان **ویندوز سرور 2019** اینجا کلیک کنید.

قالب‌بندی خروجی

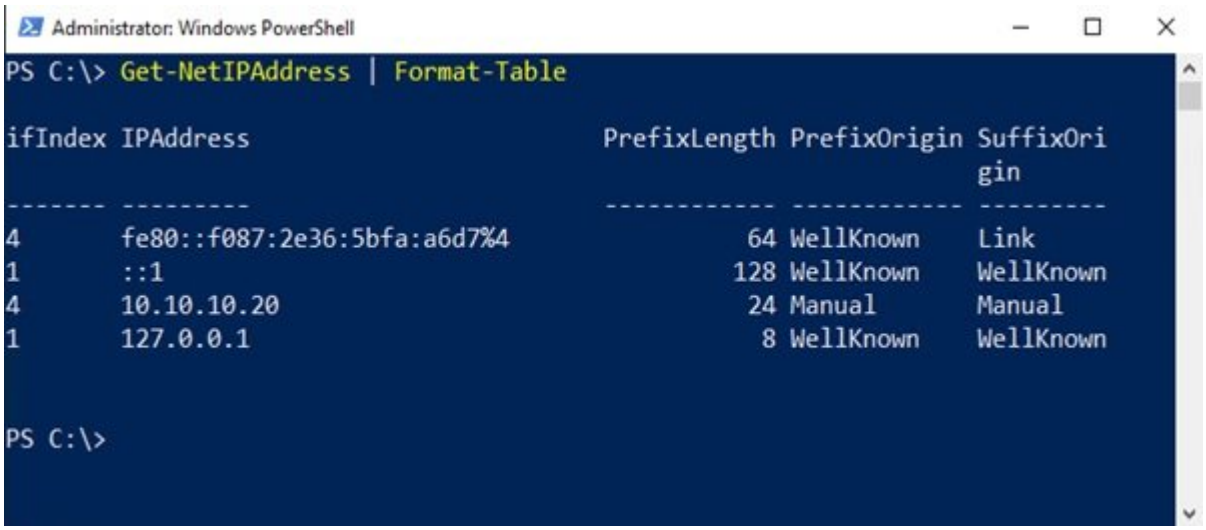
هنگام جست‌وجوی اطلاعات در پاورشل، اغلب با حجم زیادی از اطلاعات روبرو می‌شویم که مرتب کردن آن‌ها کار دشواری است. آیا به دنبال پیدا کردن فرمان مفیدی با استفاده از Get-Command هستید یا شاید به دنبال پیدا کردن نام مستعار خاصی با Get-Alias هستید. خروجی این فرمان‌های پاورشل به طرز حیرت‌انگیزی طولانی است، به همین دلیل از پارامترهای مختلفی به ویژه -Name برای خلاصه‌سازی اطلاعات استفاده می‌کنیم. البته پارامترهای دیگری نیز برای خلاصه‌سازی اطلاعات وجود دارد که در ادامه با برخی از آن‌ها آشنا می‌شویم.

قالب-جدول

هدف فرمان Format-Table بسیار ساده است: خروجی داده باید از یک فرمان دریافت شود در قالب جدول قرار گیرد. به‌طور کلی دستور فوق خواندن اطلاعات را ساده‌تر می‌کند. اجازه دهید به مثالی اشاره کنیم. ما چند بار از Get-NetIPAddress استفاده کرده‌ایم، اما همان‌گونه که متوجه شده‌اید خروجی آن چندان جالب نیست. اجرای این فرمان روی یک سرور مجازی که فقط یک کارت شبکه به آن اختصاص داده شده نزدیک به چهار صفحه اطلاعات ایجاد کرده که برخی از این اطلاعات حتا برای پیدا کردن آدرس‌های آی‌پی تخصیص داده شده به سرور مهم نیستند.



اگر بتوانیم از دستور Format-Table همراه با فرمان Get-NetIPAddress برای قالب‌بندی اطلاعات استفاده کنیم، داده‌هایی که تولید می‌شوند خوانایی بالایی دارند و اجازه می‌دهند پیدا کردن اطلاعاتی که به دنبال آن‌ها هستیم ساده‌تر شود. ترکیب نحوی به‌کارگیری دو فرمان فوق به شرح زیر است:



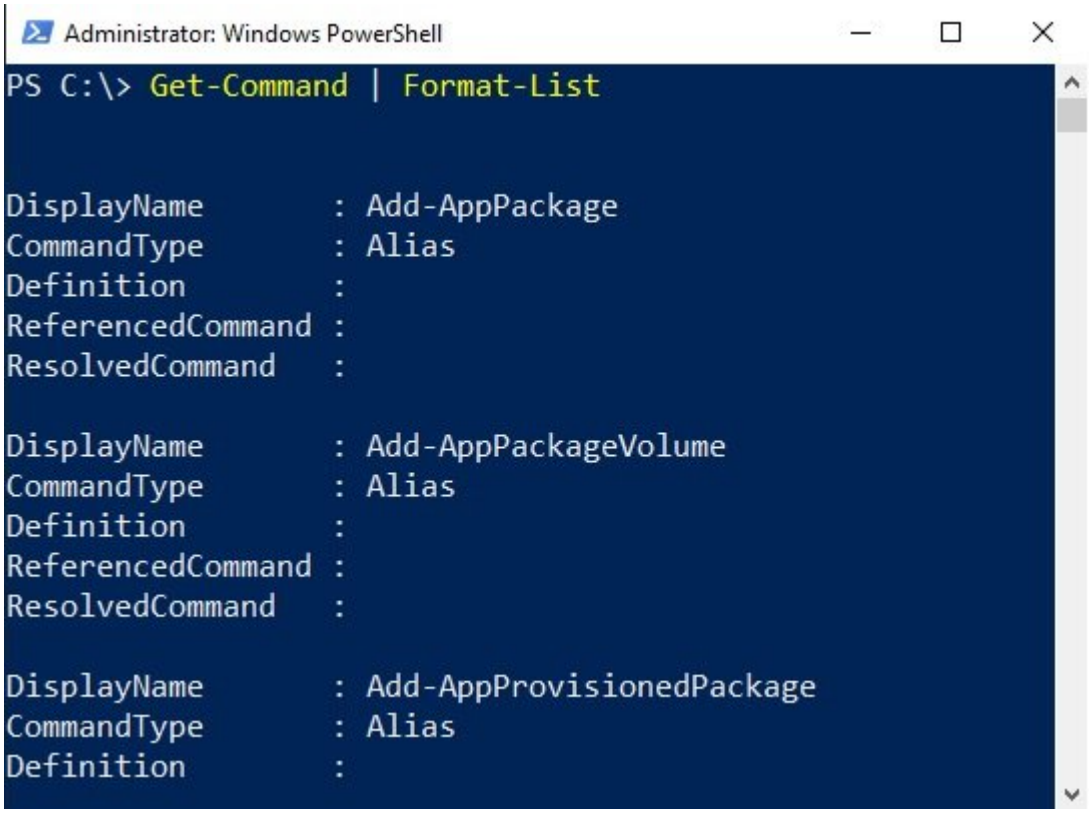
Get-NetIPAddress | Format-Table

دستور دیگری به‌نام Select-Object وجود دارد که عملکردی یکسان با فرمان Format-Table دارد. در حالی که به نظر می‌رسد Select-Object شناخته شده‌تری است، اما تجربه نشان داده که Format-Table قدرتمندتر است و عملکرد بهتری دارد.

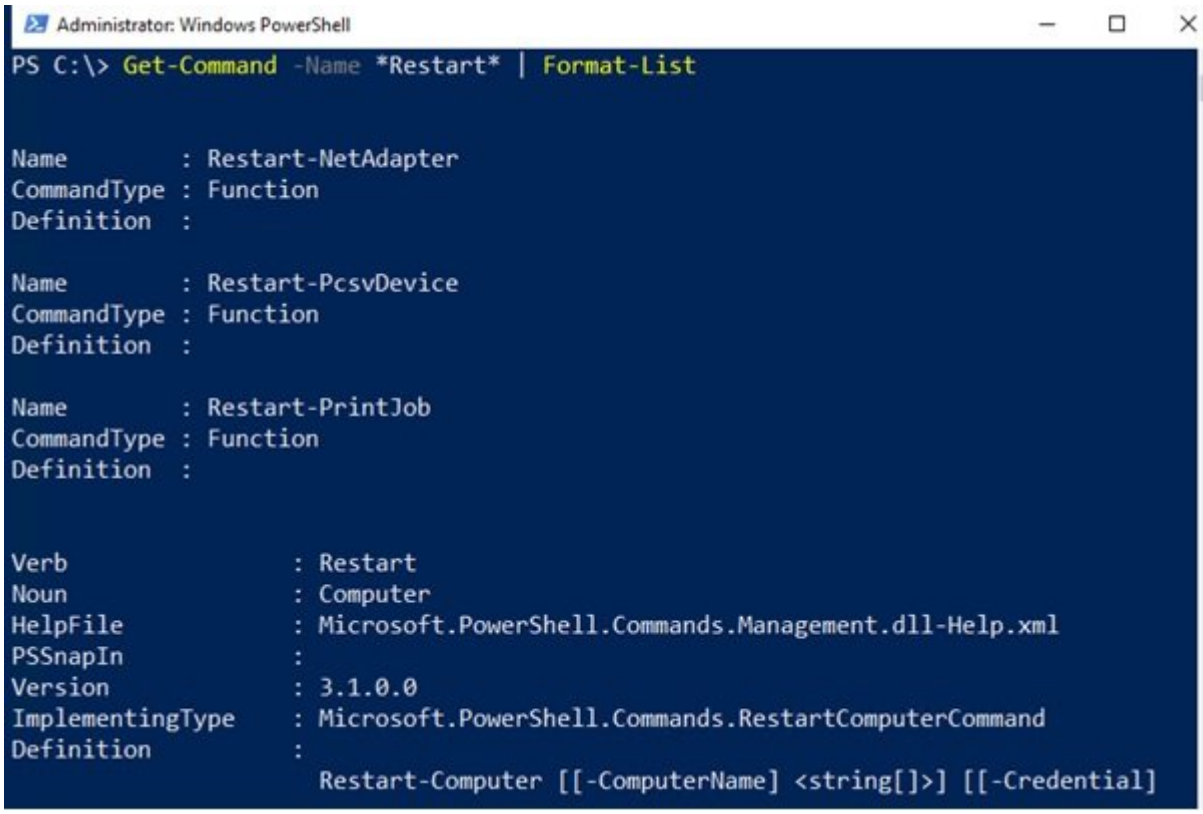
Format-List

مشابه با کاری که Format-Table انجام می‌دهد، شما می‌توانید از فرمان Format-List برای نمایش خروجی فرمان به صورت فهرستی از ویژگی‌ها استفاده کنید. می‌دانیم که فرمان Get-Command فهرستی از فرمان‌های موجود در پاورشل را در قالب جدولی ارائه می‌کند.

اگر بخواهیم خروجی را به شکل فهرست مشاهده کنیم و اطلاعات بیشتری در مورد هر فرمان به دست آوریم، باید از فرمان Get-Command همراه با Format-List به صورت زیر استفاده کنیم.



Get-Command | Format-List



خروجی این فرمان باعث می‌شود یک فهرست طولانی به دست آوریم و در حقیقت به مشکلی برسیم که چند دقیقه پیش به آن اشاره کردیم. برای حل این مشکل بهتر است نحوه نمایش اطلاعات را کمی خلاصه‌تر کنیم تا خوانایی افزایش پیدا کند. اجازه دهید جست‌وجویی در ارتباط با تمامی دستوراتی انجام دهیم که شامل کلمه Restart می‌شوند. برای این منظور از فرمان زیر استفاده می‌کنیم:

```
Get-Command -Name *Restart* | Format-List
```

محیط اسکریپت‌نویسی یکپارچه پاورشل

بیشتر سرپرستان سرورها با مفهوم ساخت فایل‌های دسته‌ای که در محیط خط فرمان اجرا می‌شوند آشنا هستند. دستوراتی که درون یک فایل قرار می‌گیرند و به صورت متوالی اجرا شوند. آیا مجبور هستید تا مجموعه‌ای از دستورات ایجاد کنید که باید روی سرورهای مختلف چند مرتبه اجرا شوند یا در آینده مجبور به اجرای چندباره برخی از دستورات هستید؟ نوشتن دستورات در یک فایل متنی و ذخیره کردن فایل با فرمت فایلی BAT باعث می‌شود تا یک فایل دسته‌ای قابل اجرا در اختیار داشته باشید که هر زمان نیاز باید با یک دستور ساده آن‌را فراخوانی کنید تا مجموعه‌ای از دستورات اجرا شوند.

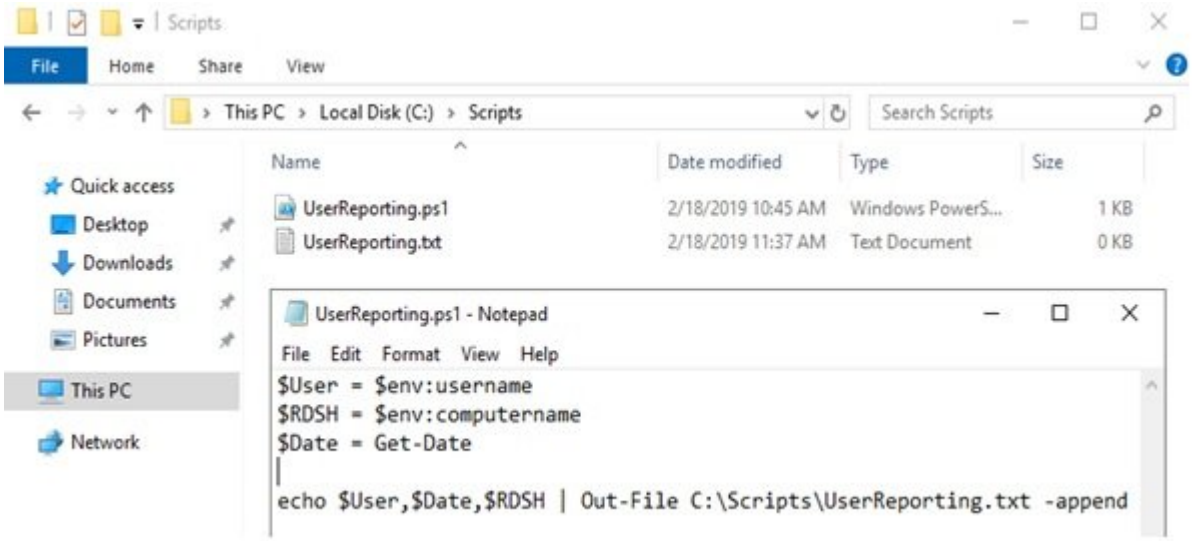
اسکریپت‌نویسی در پاورشل مشابه با فرآیند است که به آن اشاره کردیم، اما به شکل قدرتمندتری انجام می‌شود. دستورات خط فرمان مفید هستند، اما محدودیت‌هایی دارند، در حالی که فرمان‌های پاورشل می‌توانند تغییراتی در سیستم‌عامل به وجود آورند. پاورشل اجازه می‌دهد به آیتم‌هایی که درون متغیرهای محیطی یا رجیستری قرار دارند اشاره کنیم و به سادگی فرمان‌هایی روی سامانه‌های راه دور اجرا کنیم و حتی از متغیرهای موجود در یک اسکریپت پاورشل استفاده کنیم، دقیقاً مشابه با کاری که در یک زبان برنامه‌نویسی انجام می‌دهید. اجازه دهید نحوه انجام این‌کار را بررسی کنیم.

فایل‌های PS1

ساخت یک فایل ساده .PS1 (یک فایل اسکریپتی پاورشل) تقریباً مشابه با ساخت یک فایل دسته‌ای (BAT) است. تمام کاری که باید انجام دهید این است که با استفاده از ویرایشگر مورد علاقه خود یک سند متنی را باز کنید، یک سری دستورات را تایپ کنید و سپس فایل را با فرمت FILENAME.PS1 ذخیره کنید. تا زمانی‌که محیط پاورشل اجازه اجرای اسکریپت‌ها را می‌دهد (مطابق با خط‌مشی‌های DEP که پیش از این به آن اشاره کردیم) می‌توانید با

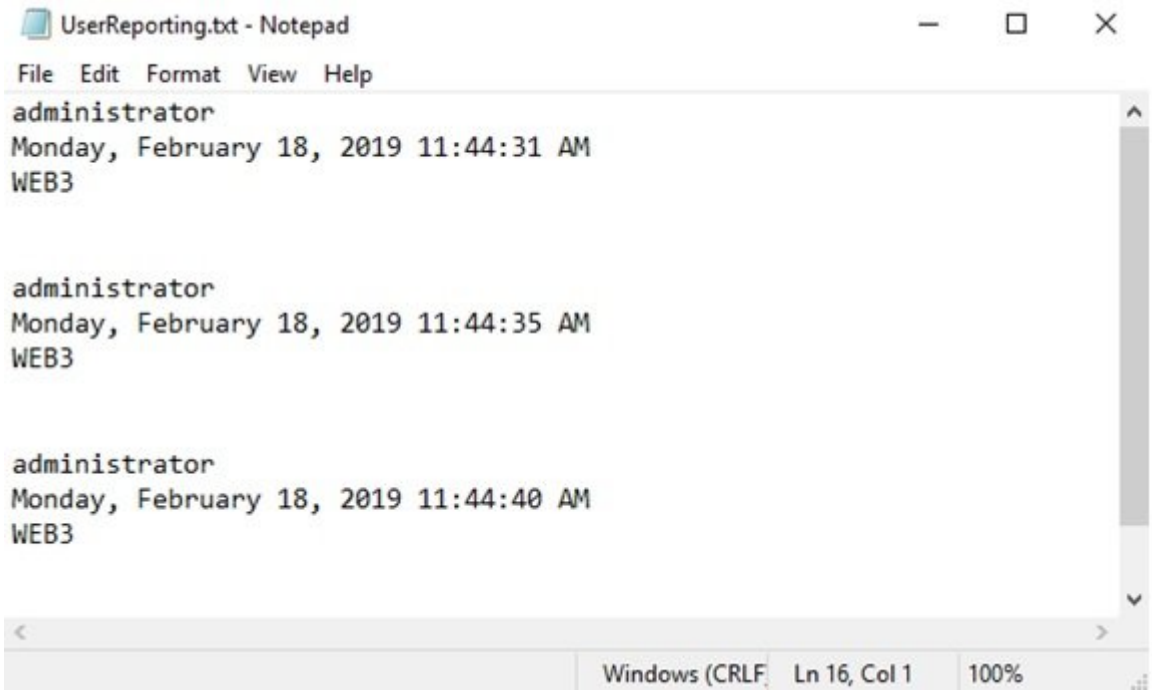
دوبار کلیک روی فایل PS1 آن‌را اجرا کیند یا فایل را همراه با پاورشل اجرا کنید. اجازه دهید به سراغ ساخت یک اسکریپت ساده برویم.

از آن‌جایی که فقط قصد ایجاد اسکریپت‌هایی داریم که تنها یک کار مشخص انجام دهند، اجازه دهید به یک مثال در دنیای واقعی فکر کنیم. من با سرورهای RDS کار می‌کنم و تصمیم دارم در مورد کاربرانی که به سرورها وارد می‌شوند گزارش قابل استنادی آماده کنم و حتا در صورت لزوم گزارش را در اختیار خود کاربران هم قرار دهم. یک روش ساده برای جمع‌آوری اطلاعات ساخت یک اسکریپت لاگین است که اطلاعات مربوط به نشست و زمان ورود کاربر را در فایلی ثبت کند. برای انجام این کار، نیاز به ایجاد یک اسکریپت دارم که بتوانم آن‌را پیکربندی کنم تا در زمان ورود به سیستم اجرا شود. برای اینکه اسکریپت کمی جالب‌تر و انعطاف‌پذیر باشد، قصد دارم از برخی متغیرها برای ایجاد نام‌کاربری، زمان و تاریخ فعلی و نام سرور RDS استفاده کنم. به این ترتیب، مجموعه‌ای از فایل‌های قدرتمندی در اختیار دارم که اجازه می‌دهد به سادگی متوجه شوم چه کاربری به چه سروری وارد شده است. من برای ساخت این اسکریپت از نوت‌پد ویندوز استفاده کنم. نرم‌افزار فوق را اجاره می‌کنم و دستورات زیر در آن وارد کرده و فایل را C:\Scripts\UserReporting.ps1 در مسیر ذخیره می‌کنم.

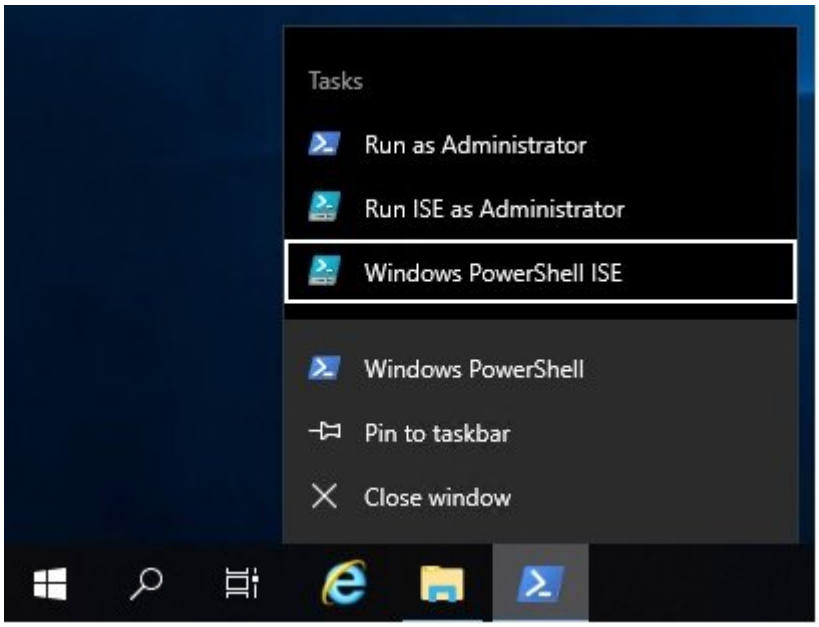



```
$User = $env:username $RDSH = $env:computername $Date = Get-Date echo
$User,$Date,$RDSH | Out-File C:\Scripts\UserReporting.txt -append
```

با نگاه کردن به اسکریپت فوق متوجه می‌شوید این اسکریپت چه کاری انجام می‌دهد. در اسکریپت فوق ابتدا سه متغیر تعریف کردم. به اسکریپت اعلام کردن که $User برابر با متغیر محیطی نام کاربری باشد. متغیر $RDSH نام سروری است که کاربر به آن وارد شده است. متغیر سوم $Date که تابع Get-Date را فراخوانی می‌کند تا تاریخ فعلی سیستم را به دست آورد و درون فایل ذخیره کند. پس از واکشی اطلاعات به درون متغیرهای پاورشل، اطلاعات درون یک فایل متنی در هارددیسک سرور ذخیره می‌شود. اگر این اسکریپت را چند بار اجرا کنیم و سپس فایل UserReporting.txt را باز کنیم، مشاهده می‌کنیم که هر بار اسکریپت اجرا می‌شود با موفقیت اطلاعات متغیرهای مشخص شده درون فایل قرار می‌گیرد.

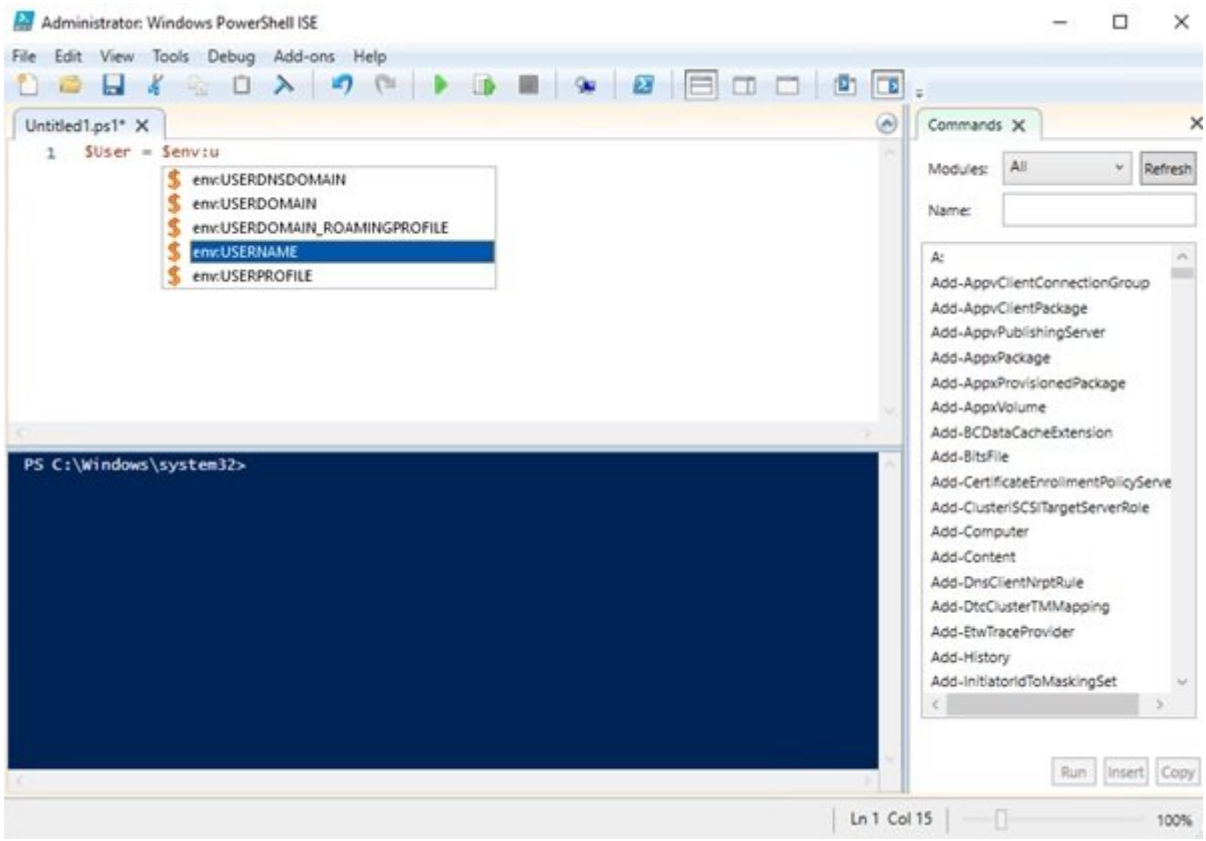


ویندوز سرور 2019 یک محیط اسکریپت‌نویسی یکپارچه (ISE) سرنام Integrated Scripting Environment در اختیار سرپرستان شبکه قرار می‌دهد. این برنامه به‌طور پیش‌فرض در ویندوز سرور 2019 نصب شده و یک پوسته اسکریپت‌نویسی ارائه می‌کند که برای اسکریپت‌نویسی پاورشل قابلیت‌های قدرتمندی در اختیارتان قرار می‌دهد. اگر فایل‌های اسکریپتی پاورشل با فرمت فایلی PS1 دارد، کافی است روی یکی از آن‌ها راست‌کلیک راست کنید و گزینه ویرایش را انتخاب کنید. همچنین با راست‌کلیک روی آیکون برنامه پاورشل گزینه PowerShell ISE را مشاهده می‌کنید.

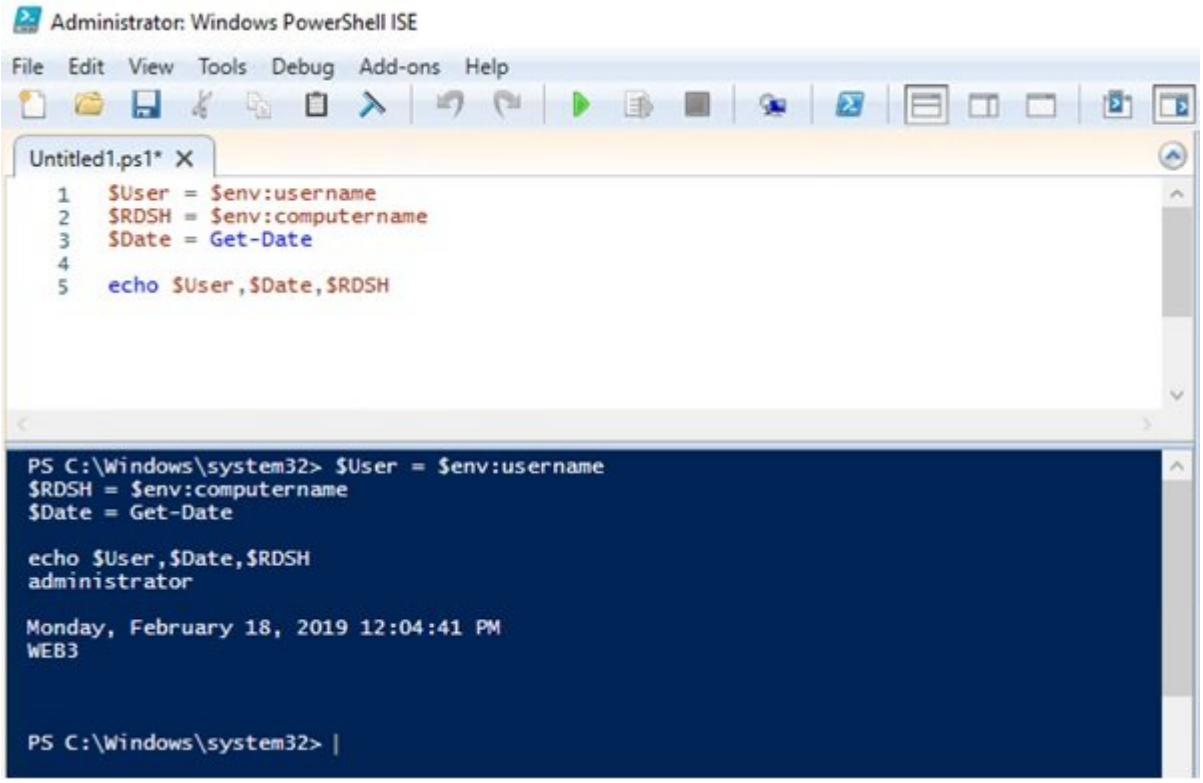


پس از اجرای برنامه اگر شروع به تایپ اطلاعات اسکریپتی کنیم که دقایقی قبل با نوت‌پد آن‌ها را نوشتیم، مشاهده می‌کنیم در زمان تایپ پنجره‌ها و گزینه‌های پیشنهادی ارائه می‌شوند که در انتخاب فرمان‌ها یا متغیرهایی که قصد استفاده از آن‌ها را داریم به ما کمک می‌کند. مطابق تکنیکی که صفحه‌کلیدهای خودکار در تلفن‌های هوشمند بر مبنای آن کار می‌کنند، ISE در مورد آنچه شروع به تایپ می‌کنید، پیشنهاداتی ارائه می‌دهد، بنابراین نیازی نیست که حتما نام یک فرمان یا پارامتر را به‌طور کامل حفظ کنید. کافی است حرفی که فرمان با آن آغاز می‌شود را تایپ کنید تا فهرستی از گزینه‌های در دسترس را مشاهده کنید. همچنین فهرستی در سمت راست ظاهر می‌شود که دسترسی به تمامی دستورات را فراهم می‌کند و حتا قابلیت جست‌وجو هم دارد. یک ویژگی عالی که اجازه می‌دهد اسکریپت‌ها در

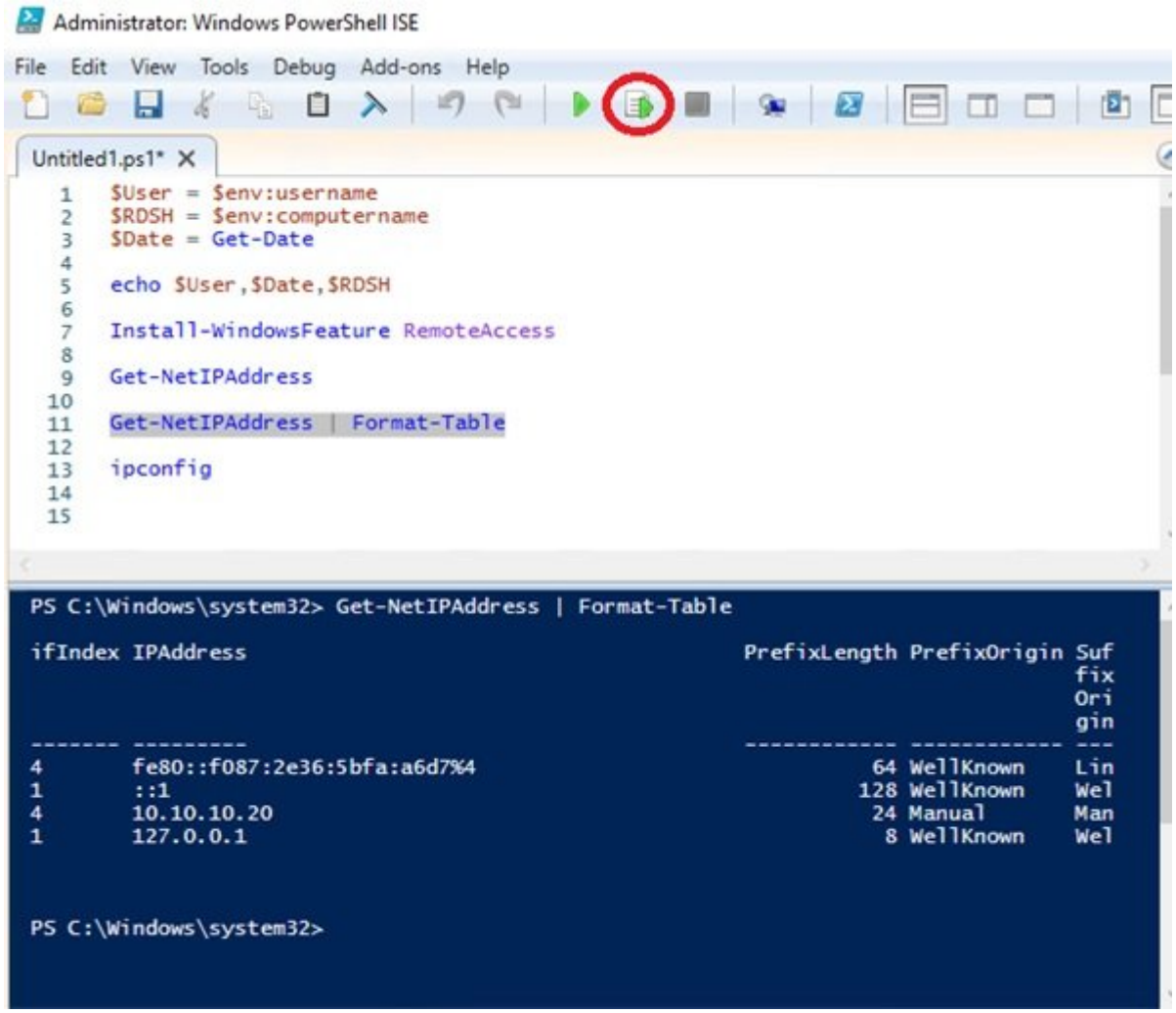
زمان کوتاهی نوشته شوند.



صفحه کوچک آبی پاورشل که نیمی از پنجره ISE را اشغال کرده کاربردی است. معمولا وقتی برخی از دستورات را تایپ می‌کنید ISE با کدگذاری فرمان‌ها و پارامترها کمک می‌کند به راحتی متوجه شوید دستورات چه کاری انجام می‌دهند. با کلیک روی دکمه فلش سبز در نوار وظیفه که دارای برچسب Run Script است یا فشار کلید F5 اسکریپتی که نوشته‌اید اجرا می‌شود. حتی اگر اسکریپت خود را ذخیره نکرده‌اید، ISE دستوراتی که نوشته‌اید را اجرا می‌کند و خروجی را در پنجره پاورشل نشان می‌دهد. قابلیت فوق کمک می‌کند تا اسکریپت‌هایی که نوشته‌اید را آزمایش کنید یا تغییراتی که در یک اسکریپت اعمال کرده‌اید را آزمایش، بدون این‌که فایل را ذخیره کرده و سپس از طریق یک پنجره پاورشل به‌طور مجزا آن‌را اجرا کنید.



محیط اسکریپت‌نویسی فوق قابلیت برجسته‌تر دیگری نیز دارد که اجازه می‌دهد تنها بخش‌های خاصی از اسکریپت خود را انتخاب کنید و فقط آن دستورات را اجرا کنید. یک فایل اسکریپتی PS1 ممکن است مملو از دستورات رایج پاورشل باشد که شما برای انجام کارهای روزانه خود به آن‌ها نیاز خواهید داشت. در این حالت زمانی که تنها نیازمند اجرای بخشی از دستورات هستید کافی است دستورات موردنظر را انتخاب کنید و سپس گزینه Run Selection را انتخاب کرده یا کلید (F8) را فشار دهید تا دستورات انتخابی اجرا شوند. با انتخاب دستورات درون متن قبل از اجرای اسکریپت در محیط ISE تنها فرمان‌های انتخاب شده اجرا می‌شوند. در تصویر زیر مشاهده کنید که تعداد فرمان‌های موجود در فایل من زیاد هستند، اما تنها دستوراتی که انتخاب شده‌اند اجرا می‌شوند.



مدیریت از راه دور سرور

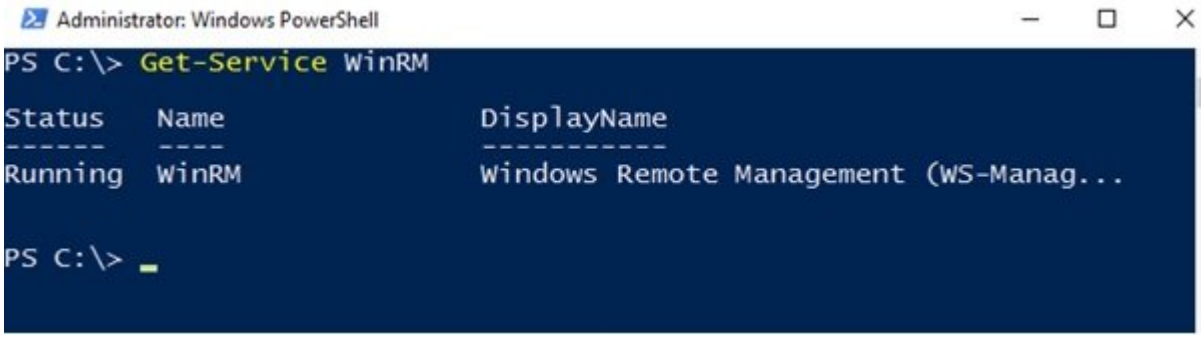
اکنون که کمی با نمونه محلی پاورشل کار کردیم و چند روش ساده برای شروع اسکریپت‌نویسی را بررسی کردیم، بهتر است نگاه دقیق‌تری به پاورشل داشته باشیم. در گذشته به شما گفتیم که این امکان وجود دارد تا از Server Manager به عنوان یک نقطه مرکزی برای مدیریت از راه دور سرورها استفاده کرد و همچنین می‌دانیم که ابزارهای موجود در Server Manager در اکثر موارد تنها با یک کلیک ساده کار چند دستور پاورشل را انجام می‌دهند. اما گاهی اوقات مدیریت سرورهای از راه دور با استفاده از پاورشل کار ساده‌تری است، بدون آن‌که نیازی به لاگین کردن به سیستم ضرورتی داشته باشد. برای انجام این‌کار در اولین گام باید مطمئن شویم که سرورها آماده پذیرش اتصالات از راه دور پاورشل هستند.

آماده‌سازی سرور از راه دور

برای انجام این‌کار مولفه‌ها و گزینه‌هایی وجود دارند که برای روی سرور فعال شده و در حال اجرا باشند تا بتوانید از طریق پاورشل و از روی سیستم‌های مختلف به سرورها دسترسی داشته باشید. اگر همه سرورها مبتنی بر ویندوز سرور هستند، Remoting PowerShell به‌طور پیش‌فرض روی آن‌ها فعال است. با این حال ، اگر سعی کردید از PowerShell remoting استفاده کنید، اما این امکان فراهم نبود، مهم است که علت عدم کارکرد را بررسی کنید. به این ترتیب، می‌توانید در صورت بروز مشکلات، فرآیند عیب‌یابی را انجام دهید و به صورت دستی قابلیت دسترسی از راه دور را فعال کنید. در برخی از سیستم‌ها ممکن است به دلیل خط‌مشی‌های خاصی برخی از مولفه‌ها غیرفعال شده باشند و در نتیجه امکان برقراری اتصال از راه دور مبتنی بر پاورشل وجود نداشته باشد، در این حالت باید برخی از آیتم‌های موجود در سیستم را بررسی کنید.

سرویس WinRM

یک مولفه‌های مهم مدیریت از راه دور سرویس WinRM است. مطمئن شوید این سرویس در حال اجرا است. اگر به دلایل امنیتی سرویس فوق را غیرفعال کرده‌اید برای استفاده از Remoting PowerShell مجبور هستید سرویس فوق را فعال کنید. بهتر است از services.msc برای بررسی فعال یا غیر فعال بودن سرویس فوق استفاده کنید. البته این امکان وجود دارد که از دستور زیر در محیط پاورشل برای بررسی این موضوع استفاده کنید.




```
Get-Service WinRM
```

Enable-PSRemoting

به‌طور معمول، کار دیگری که برای روی یک سرور راه دور انجام دهید، اجرای یک فرمان ساده است. البته، برای انجام این‌کار باید به شبکه دسترسی داشته باشید برای آن‌که به یک سرور اجازه دهید دستورات و ارتباطات راه دور پاورشل را قبول کند، باید فرمان زیر را اجرا کنید.

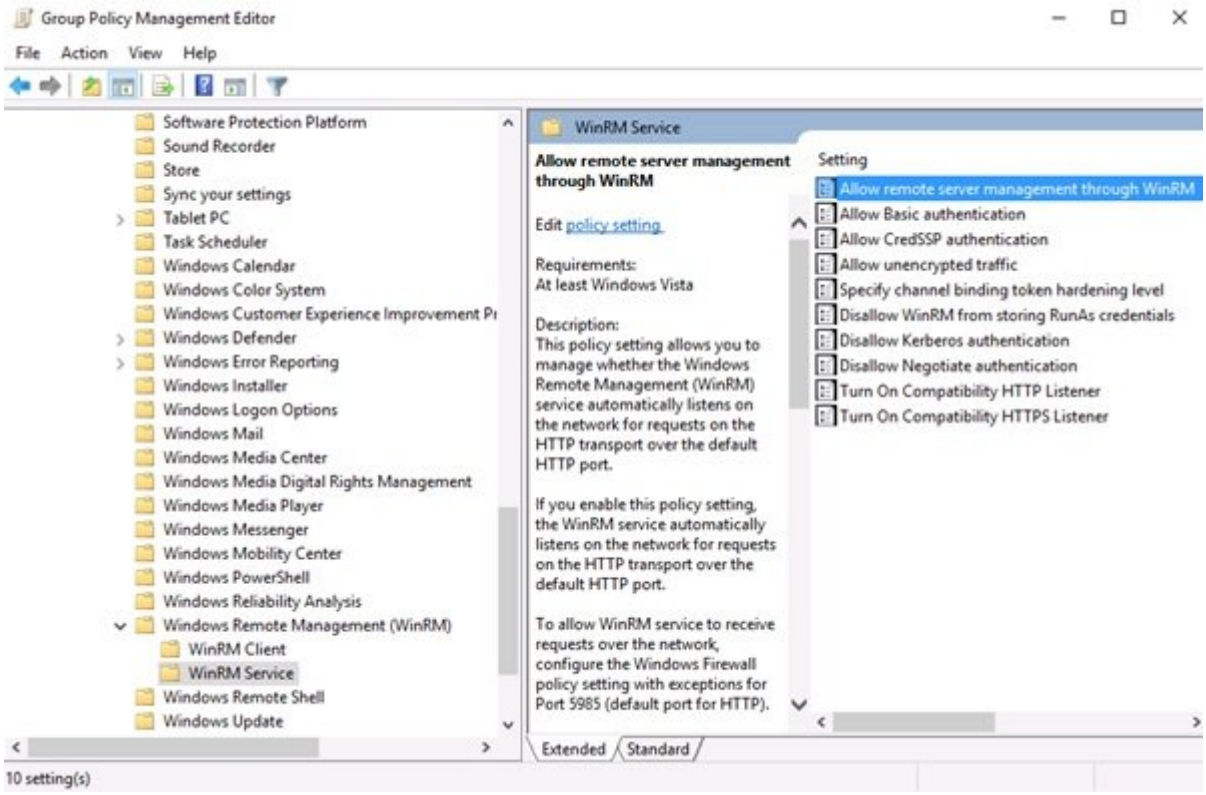
```
Enable-PSRemoting -Force
```

پارامتر -Force در انتهای فرمان Enable-PSRemoting باعث می‌شود که فرمان بدون درخواست تأیید از شما اجرا شود. Enable-PSRemoting چند کار مختلف در پس‌زمینه انجام می‌دهد. ابتدا تلاش می‌کند سرویس WinRM را راه‌اندازی کند. البته بهتر است خودتان سرویس WinRM را همان‌گونه که اشاره شد فعال کنید. در ادامه به ارتباطات راه دور گوش می‌کند و یک قاعده دیوارآتش روی سیستم ایجاد می‌کند تا این ترافیک بتواند با موفقیت از آن عبور کند.

اگر قصد استفاده از Remoting PowerShell در مقیاس بزرگ را دارید، بهتر است به جای مراجعه دستی از Group Policy استفاده کنید تا تغییرات به‌طور خودکار روی سرورها انجام شود. یک GPO جدید ایجاد کنید، پیوندها و فیلترهای مربوطه را تنظیم کنید تا تغییرات فقط سرورهایی که قرار است به شکل مرکزی مدیریت شوند، اعمال شود. برای انجام این‌کار به مسیر زیر بروید.

Computer Configuration | Policies | Administrative Templates | Windows Components | Windows Remote Management (WinRM) | WinRM Service

و سپس گزینه Set Allow remote server management through WinRM to Enabled را فعال کنید.



در شماره آینده به سراغ مبحث پاورشل در **ویندوز سرور 2019** خواهیم رفت.

برای مطالعه تمام بخش‌های آموزش **ویندوز سرور 2019** روی لینک زیر کلیک کنید:

آموزش رایگان ویندوز سرور 2019

تاریخ انتشار:
25 آذر 1398

نشانی منبع:
https://www.shabakeh-mag.com/networking-technology/16387/%DA%86%DA%AF%D9%88%D9%86%D9%87-%D8%A7%D8%B2-%D9%BE%D8%A7%D9%88%D8%B1%D8%B4%D9%84-%D8%AF%D8%B1-%D9%88%DB%8C%D9%86%D8%AF%D9%88%D8%B2-%D8%B3%D8%B1%D9%88%D8%B1-2019-%D8%A8%D8%B1%D8%A7%DB%8C-%D8%A7%D8%B3%DA%A9%D8%B1%DB%8C%D9%BE%D8%AA%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C-%D8%A7%D8%B3%D8%AA%D9%81%D8%A7%D8%AF%D9%87-%DA%A9%D9%86%DB%8C%D9%85%D8%9F