

Introduction to PowerShell

Windows PowerShell

- PowerShell is relatively new and the entire windows server 2012 operating system is built around it.
 - we will see PowerShell throughout the operating system, how the different functionalities of this server OS can be done with PowerShell
- It can be regarded as the new command line interface (command prompt), and it can be used as a scripting engine
- In windows server 2012, the entire GUI is built on top of it, i.e. we can do all the things we do in the GUI using PowerShell

Windows PowerShell

- PowerShell as a whole is huge (it can take a whole course)
- You access windows PowerShell by clicking its icon on the windows taskbar



- Or we can open windows PowerShell from the server manager, Tools menu
- Or use the start screen to open it as any other program

Windows PowerShell

- PowerShell is used for command line administration and also used for scripting
- It is more powerful than the cmd
- In PowerShell, we use commandlets (Cmdlet)
 - E.g. get-location,
 - It is verb-noun combination.
 - We use command verbs like get, set, new etc
 - Nouns like location, item, object, service, date etc

Windows PowerShell

- We can use the help feature how to use a command
 - E.g. “get-help get” lists all possible commands with “get”
- You can clear the screen using “clear-host” or “clear” command (like the cls in cmd)
- The old cmd line command also work here (called aliases)
- You can for example change the location of the PowerShell prompt by typing “set-location” then the place

Windows PowerShell

- E.g. set-location desktop
- We can create a folder there as follows:
 - New-item -name “my folder” -type directory
- To create a word file in it, do the following:
 - Set-location “my folder”
 - For names with blanks, put in quotation
 - New-item -name “test file.docx” -type file

Filtering with pipeline

- Type “get-service” lists all the services
- To filter only the services that are running, we use pipeline | as follows:

Get-Service | where-object {\$_.status -eq “Running”}

- The \$_. represents “any” (wild card).
- You can also use “name” on place of “status” to see only one service
- We can nest pipeline deeper as follows:

Get-Service | where-object {\$_.status -eq “Running”} | sort-object displayname

Filtering with pipeline

- We can save the output to a text file
- E.g. type `ipconfig`
- To save the output, type the following:
`Ipconfig | out-file "ip.txt"`
- To see if you have created it, type `"get-childitem"`
- To append text to the above file, do the following
`get-date | out-file "ip.txt" -append`
- Check the file for the change

Batch File

- To save a batch file (which is a script that can be run to do something), we create file with .ps1 extension (power shell extension).
- Type the following:
 - `new-item -name "date.ps1" -type file`
 - `edit date.ps1`
- Open the text file using any text editor and just type the PowerShell script, for this case as a simple example, just type the following:
 - `Get-date`
- Then save and exit: use the File menu save, then exit

Batch File

- To run the script, type `.\date.ps1`
- If you get error because running scripts is disabled by default (on workstations), type `set-executionpolicy remotesigned`
- You see the result of the script

PowerShell ISE

- For users who are familiar with the GUI, there is a better PowerShell environment, called windows PowerShell ISE (integrated scripting environment)
 - To open it, right click on the PowerShell icon at the windows taskbar, select PowerShell ISE or you can find it using the Tools menu at the server manager
- The PowerShell ISE assists us by auto populating certain command when typing We use the tab key to select the command it suggests
- Type Get-Command -Noun service
 - This gives us all the commands that have to do with “service”
 - We can do all those things related to service in powershell

PowerShell ISE

- To see all the processes running on the system, type `get-process` in PowerShell
 - This is also the output we get using the task manager
 - To see the task manager, you can right click on the task bar, and select task manager, click on More Details to see the details.
 - Then click on view menu, and uncheck Group by type
 - This list is equivalent to the PowerShell list by `get-process`.

PowerShell Vs Command Prompt

- Open command prompt (by typing cmd on the run dialog box, or any other way)
- To see all the services that are running on windows, type “net start” on command prompt
- The equivalent command on PowerShell is *Get-Service | where-object {\$_.status - eq “Running”}*
- Which one is easier?

PowerShell Vs Command Prompt

- From a typing perspective, net start is easier, but from a memorization perspective PowerShell command is easier.
 - Net start is something you must know, it doesn't logically make sense till you learn it
 - PowerShell command is more intuitive.
- PowerShell is so much more flexible, it is capable of doing all the things we do using the GUI