

UNIVERSIDAD TECNOLOGICA DE PUEBLA

TECNICO SUPERIOR UNIVERSITARIO EN TECNOLOGIAS DE LA INFORMACION

AREA: INFRAESTRUCTURA DE REDES DIGITALES

MATERIA: ADMINISTRACION DE SERVIDORES II

PRODUCTO NUMERO 4 REPORTE DE CONFIGURACION DE SERVICIOS EN LA NUBE

MAESTRO: JOSE DE JESUS GARCIA SANCHEZ

ALUMNOS:

CONTRERAS GUTIERREZ MAURICIO ALEJANDRO

MONDRAGON ALVAREZ JENNIFER

SAINOS BONILLA CHRISTIAN ALEXIS

VALERA FAJARDO ALAN ERNESTO

MATRICULAS:

UTP0156338

UTP0157138

UTP0156344

UTP0143047

5º “G”

FECHA DE ENTREGA: 21/03/2024

DEFINICION DE POWER SHELL

PowerShell es una solución de automatización de tareas multiplataforma formada por un shell de línea de comandos, un lenguaje de scripting y un marco de administración de configuración. PowerShell funciona en Windows 10, Linux y macOS.

PowerShell es un entorno de línea de comandos y un lenguaje de scripting desarrollado por Microsoft. Se basa en el marco .NET y está diseñado para la automatización de tareas administrativas y la gestión del sistema en entornos Windows. PowerShell permite a los usuarios interactuar con el sistema operativo y las aplicaciones mediante la ejecución de comandos y scripts para realizar una amplia variedad de tareas, desde la administración del sistema hasta la configuración de servicios y la gestión de recursos. Ofrece una amplia gama de características avanzadas, como el acceso directo a objetos del sistema y la capacidad de ejecutar comandos de otros lenguajes y entornos de desarrollo, lo que lo convierte en una herramienta poderosa para administradores de sistemas y desarrolladores en entornos Windows.

CARACTERISTICAS DE POWER SHELL

Lenguaje de scripting orientado a objetos: PowerShell está diseñado para trabajar con objetos, lo que facilita el manejo y la manipulación de datos de forma estructurada.

Acceso a todos los aspectos del sistema operativo: PowerShell ofrece acceso a una amplia gama de funciones y características del sistema operativo Windows, lo que permite a los usuarios administrar casi todos los aspectos del sistema.

Integración con el marco .NET: PowerShell está construido sobre el marco .NET, lo que le permite acceder a todas las bibliotecas y clases de .NET, lo que brinda una gran flexibilidad y potencia para el desarrollo de scripts y la automatización de tareas.

Interfaz de línea de comandos intuitiva: La sintaxis de PowerShell es fácil de aprender y entender, lo que facilita la escritura y ejecución de comandos, especialmente para aquellos familiarizados con otros lenguajes de scripting o programación.

Automatización de tareas repetitivas: PowerShell es especialmente útil para la automatización de tareas administrativas repetitivas, lo que permite a los administradores de sistemas ahorrar tiempo y reducir errores mediante la creación de scripts para realizar estas tareas de manera eficiente.

Compatibilidad con remoting: PowerShell permite la administración remota de sistemas, lo que significa que los usuarios pueden ejecutar comandos y scripts en computadoras remotas desde una ubicación central, facilitando la administración de infraestructuras distribuidas.

Extensibilidad: PowerShell es altamente extensible, lo que significa que los usuarios pueden crear y compartir módulos, scripts y funciones personalizadas para ampliar su funcionalidad y adaptarlo a sus necesidades específicas.

INSTALACION DE POWER SHELL

COMANDOS BASICOS DE POWER SHELL

|  |  |  |
| --- | --- | --- |
| COMANDO | FUNCIONALIDAD | EJEMPLO |
| Get-ChildItem | Muestra los elementos (archivos y carpetas) en una ubicación específica del sistema de archivos. | Get-ChildItem C:\Users muestra los elementos dentro del directorio "C:\Users" |
| Set-Location: | ambia el directorio de trabajo actual en la sesión de PowerShell. | ambia el directorio de trabajo actual en la sesión de PowerShell. |
| New-Item | Crea un nuevo archivo o carpeta en el sistema de archivos. | New-Item -ItemType File -Path C:\Temp\archivo.txt crea un nuevo archivo llamado "archivo.txt" en la carpeta "C:\Temp". |
| Copy-Item: | Copia un archivo o carpeta de una ubicación a otra. | Copy-Item C:\archivo.txt D:\Backup copia el archivo "archivo.txt" de "C:" a la carpeta "D:\Backup". |
| Move-Item: | Mueve un archivo o carpeta de una ubicación a otra. | Move-Item C:\archivo.txt D:\Documentos mueve el archivo "archivo.txt" de "C:" a la carpeta "D:\Documentos". |
| Remove-Item | Elimina un archivo o carpeta. | Remove-Item C:\archivo.txt elimina el archivo "archivo.txt" de "C:". |
| Get-Process: | Obtiene una lista de los procesos en ejecución en el sistema. | Get-Process muestra una lista de todos los procesos en ejecución. |
| Stop-Process | Detiene un proceso en ejecución | Stop-Process -Name notepad detiene todos los procesos llamados "notepad" en el sistema. |
| Get-Service | Obtiene una lista de los servicios en el sistema. | Get-Service muestra una lista de todos los servicios en el sistema. |
| Start-Service | Inicia un servicio detenido en el sistema. | Start-Service -Name Spooler inicia el servicio "Spooler" en el sistema |
| Get-Help | Proporciona ayuda y documentación sobre cmdlets y otros elementos de PowerShell. | Get-Help Get-Service muestra la ayuda para el cmdlet Get-Service. |
| Set-ExecutionPolicy | Establece la directiva de ejecución de scripts en PowerShell para controlar la ejecución de scripts. | Set-ExecutionPolicy RemoteSigned establece la política de ejecución para permitir la ejecución de scripts locales firmados y scripts remotos sin firma. |
| Get-Content | Obtiene el contenido de un archivo de texto. | Get-Content C:\archivo.txt muestra el contenido del archivo "archivo.txt". |
| Select-Object: | Selecciona propiedades específicas de un objeto y las muestra en la salida. | Get-Process | Select-Object Name, Id muestra solo los nombres y los identificadores de los procesos en ejecución. |
| Sort-Object: | Ordena los objetos devueltos según una propiedad específica. | Get-Service | Sort-Object Status ordena la lista de servicios según su estado. |
| Format-Table | Formatea la salida de los objetos en forma de tabla. | Get-Process | Format-Table -Property Name, CPU muestra una tabla con los nombres de los procesos y el uso de CPU. |
| ForEach-Object: | Ejecuta un bloque de comandos para cada objeto en una secuencia de entrada. | Get-ChildItem | ForEach-Object { Write-Output $\_.Name } muestra el nombre de cada archivo y carpeta en el directorio actual. |
| Out-File | Envía la salida de un comando a un archivo. | Get-Process | Out-File -FilePath C:\procesos.txt guarda la lista de procesos en el archivo "procesos.txt". |

SINTAXIS DE POWERSHELL

ARGUMENTOS Y PARÁMETROS DE POWERSHELL

CREACIÓN DE OBJETOS PERSONALIZADOS

.