

Actividad 1-Despliegue de Aplicaciones Web

Tarea individual-comandos pOWERshell

guía de uso

Sara Dozzi Segura |2º DAW| DESPLIEGUE DE APLICACIONES WEB

# Guía o “how-to”

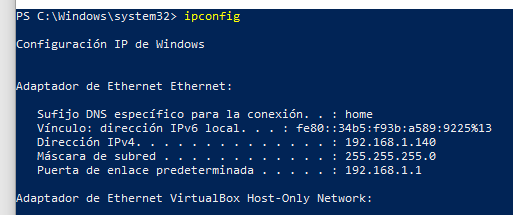
1. **Comandos para saber si tenemos conexión a internet**

**Ipconfig:** muestra información detallada sobre el adaptador de red, como su ip, MAC, gateway o puerta de enlace, dns.

Nos muestra información detallada acerca de la **configuración TCP/IP**de nuestro sistema y es posible **actualizar la configuración del DHCP** a través de este comando.

Se usa para saber si conecta correctamente a la red y que tipo de conexión tiene.

La información que proporciona permite saber si tenemos conexión además de comprobar con ping si la conexión es correcta.



**Ping:** acrónimo de **Packet Internet Groper, o buscador de paquetes en redes.**

**Verifica el estado de una conexión host local, con conexión tcp/ip y sirve para comprobar si una dirección IP es accesible desde la red.**

Permite conocer **el tiempo total que tarda un servidor en responder**una petición y así determinar si la conectividad está o no correcta.

**Su forma de ejecución sería** *PING dirección\_ip.*

Dispone de muchos parámetros, si lo mandamos sin ninguno, envía 4 solicitudes.

En su respuesta se incluye TTL(Time of life)= es el tiempo de vida del paquete enviado y su valor óptimo es 128.

Cuando hay errores suele indicar lo siguiente:

"Red de destino inaccesible" significa que no existe ninguna ruta al destino.

"Ha terminado el tiempo de espera para esta solicitud" indica errores en la conexión.

1. **Comandos para saber si nuestro servidor es accesible desde internet.**

**netstat:** el comando por defecto informa sobre conexiones de red activas.

Es una importante herramienta para controlar el tráfico de red entrante y saliente en busca de software dañino. Por lo que se recomienda revisar los puertos abiertos del sistema con el comando frecuentemente.

Los puertos abiertos y direcciones IP desconocidos son indicios de infección.

Esto responde a cómo saber si nuestro servidor es accesible y está activo a través de su puerto y en ejecución

el comando sigue el siguiente patrón:

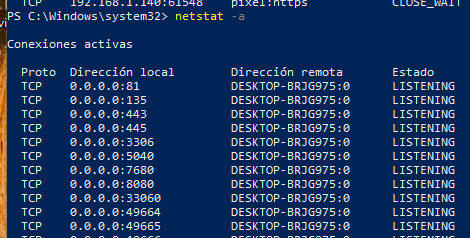
**netstat [-a] [-b] [-e] [-f] [-n] [-o] [-protocolo] [-r] [-s] [-t] [-x] [-y] [Intervalo]**

Algunos parámetros como **-a** indican los puertos activos.

**-o** indica el PID o id del proceso, lo que conduce a saber que aplicación está utilizando un puerto determinado, combinándolo con varios parámetros como por ejemplo:

**netstat -ano :**lista de conexiones activas con su id de proceso o PID**.**

**netstat -p IP**: lista de conexiones para el protocolo IPV4.



**netsh (network shell)**

Permite la administración y configuración de redes locales y remotas.

Por ejemplo para configurar el firewall de windows. Gestión de redes LAN y WLAN.

También sirve para diagnosticar problemas de red.

Entre los más importantes destacan:

**nesh + advfirewall:**

Forma parte del paquete cmdlet, y sirve para manejar funciones del firewall de Windows.

**Enable-NetFirewallRule:** sirve para habilitar reglas del firewall.

**nesh firewall set opmode enable/disable:** habilita o desactiva el firewall de Windows.

**netsh int ip reset**: en caso de configuración incorrecta restablece el protocolo TCP/IPv4.Es necesario reiniciar para que se efectúen los cambios.

1. **Cómo sabemos a quién pertenece una dirección web o URL.**

**nslookup:** permite administrar DNS o nombre de dominio en Windows. (Domain Name Server)

Nos permite determinar si un servidor está resolviendo de manera correcta los **nombres y las direcciones IP**.

En la siguiente imagen no hay servidor DNS identificado, y muestra una dirección



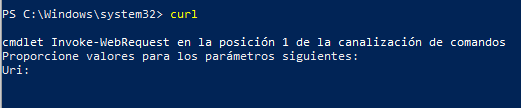
Si queremos conocer un dominio de una IP concreta como 8.8.8.8 que corresponde al DNS de google. O al revés, indicando el nombre de dominio, del que recibiremos su transcripción. Esto responde a la pregunta para saber a quién pertenece una URL.



1. **Comprobación para acceder a un servidor:**

**Invoke-WebRequest:** es el equivalente a curl de Linux, en Windows,

Si ejecutamos curl en PWSHELL aparece lo siguiente:



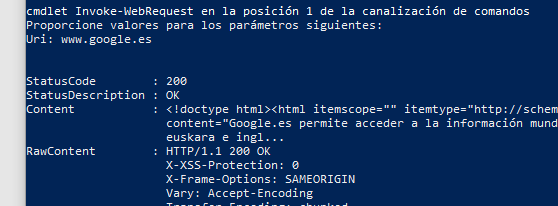
Nota: a partir de la version 3.0 de PW. se incluye CURL(Client URL) en Windows, siendo en realidad un alias de Invoke-webRequest.

En general se usa para comprobar la conectividad a una URL o para transferir datos.

Su uso más simple consiste en mostrar el contenido de una página, lo cual

Esto **explica el acceso un servidor** ya que permite realizar transferencias HTTP, HTTPS, FTP, telnet, etc

En el siguiente ejemplo muestra el código fuente de la página de inicio de google, tras indicar la URI nos muestra el contenido de la transacción HTTP realizada:



También se utiliza para descargar elementos o imágenes de una web:

Por ejemplo, especificando la URI, (con un script podría ser una variable $) y el destino donde queramos guardar la descarga:

Curl acompañado de los siguientes parámetros permiten guardar ficheros en diferentes ubicaciones.

Parametros de Curl:

-o permite especificar un nombre de archivo o ubicación diferente.

-O guardará el archivo en el directorio actual con el mismo nombre del archivo del equipo remoto.

Por último, con **curl**, se pueden realizar peticiones por get y post:

Ejemplo con GET: **curl** [**http://mydomain.com**](http://mydomain.com)

Ejemplo con POST: **curl –data “text=Hello” https://myDomain.com/firstPage.jsp**

Su equivalente en Powershell sería **Invoke-WebRequest,** cuyos alias son CURL, WGET y IWR.

Este comando envía solicitudes HTTP/HTTPS a una página o servicio web. Traduce la respuesta y obtiene imágenes y otros ficheros, y elementos html.

A partir de la versión 7.0 soporta configuración de servidor Proxy definida mediante variables de entorno.

**Invoke-WebRequest** -**Uri** http://dominio.com/ -OutFile C:\ImagenDescargada.jpg

Uri: donde indicamos la direccion URL.

OutFile: directorio donde se quiere descargar del equipo.

Podemos detener un proceso en un servidor, por ejemplo un notepad ejecutando lo siguiente:

**Invoke-Command -Session $sesion1 -ScriptBlock {Stop-Process -name  notepad}**

Se podría ejecutar el mismo comando en varios servidores:

**Invoke-Command -ComputerName Servidor01, Servidor02 {hostname}**

O mantener una sesión permanente en los servidores:

**Invoke-Command -Session $Sesion {hostname}**

También con este mismo comando se puede **iniciar sesión** como por ejemplo en la siguiente página:

$R = Invoke-WebRequest https://www.facebook.com/login.php-SessionVariable fb

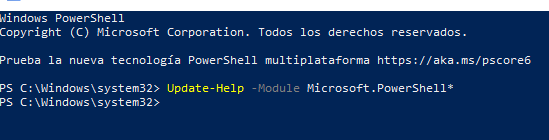


Con SessionVariable se crea una peticion http de sesión y se le asigna una variable.

1. **Otros comandos útiles:**

**-man**: comando para encontrar ayuda sobre el uso de otros comandos y herramientas de la consola.

-**Update help**: actualiza ficheros de ayuda para diferentes paquetes o algunos especificos de Powershell.



**-get-help Invoke-WebRequest** -examples:

Muestra ejemplos del comando especificado.

**-Ifconfig /all**: Muestra información más detallada sobre la configuración de red

**-?:** muestra ayuda también para los comandos.

**-netstat -ano:** comando muy útil para conocer las conexiones activas y "puertos de escucha" con sus respectivos puertos en forma númerica y PID (id del proceso), que conducen a la aplicación que utiliza dicho puerto.

get-help "comando" -detailed: para consultar comandos complejos que tienen muchas especificaciones, como **curl** por ejemplo.

NOTA: Los comandos propios de Linux, como por ejemplo CURL, se consultan con el comando de ayuda **man.**

**-tracert**:  **herramienta de diagnóstico** la cual nos permite hacer un monitoreo de los paquetes que son originados desde un host.

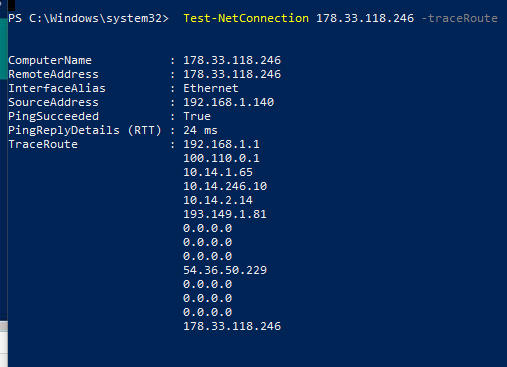
Nos permite ver la ruta que siguen los paquetes en la red, de una dirección ip determinada.

**-cmdlet Get-NetIPConfiguration: nos permite obtener información más detallada sobre los parámetros del adaptador de red:**

* **Tarjeta del adaptador de red.**
* Servidores **DNS**
* Direcciones **IP**
* **Interfaces** usadas.

**Si añadimos el parámetro -detailed, obtendremos aún más detalles sobre la red.**

cmdlet**Test-NetConnection: similar a ping,lo podemos acompañar del parámetro traceRoute:** Por ejemplo, para monitorear la ruta de "Solvetic":

****

**Se pueden ver los servidores por los que pasa la ruta (nodos), su destino Ip y su origen, y el estado de PING.**