

Actividad 1. Comandos

En la siguiente guía vamos a desarrollar el uso de diferentes comandos sobre un servidor y el comportamiento que tiene cada uno analizando su salida.

Toda la actividad se desarrolla en una máquina virtual creada con VMWare y ubuntu 20.04 LTS.

1.- ¿Cómo sabemos si tenemos conexión a internet?

Para saber si tenemos conexión a internet hay que sacar las direcciones ip que tiene nuestra máquina virtual y realizar un ping hacia ellas para ver si se reciben bien los paquetes o se pierden.

- **ifconfig**: hace una lista de las direcciones ip de nuestra máquina.

```
jaime@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.28 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::427c:f74:5d9e:9d43 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:0c:d1:2c txqueuelen 1000 (Ethernet)
    RX packets 5193 bytes 7427473 (7.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1374 bytes 102402 (102.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 215 bytes 18253 (18.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 215 bytes 18253 (18.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jaime@ubuntu:~$
```

- **ping**: Envía un ping a un servidor y comprueba si hay respuesta.

```
jaime@ubuntu:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.092 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.061 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.065 ms
^Z
[1]+  Stopped                  ping 127.0.0.1
jaime@ubuntu:~$ Hay respuesta del localhost
```

2.- ¿Cómo sabemos si nuestro servidor es accesible desde internet?

Para realizar esta parte investigaré sobre dos comandos

1.- **ufw**: Se utiliza para hacer la configuración del firewall. Puedes abrir y cerrar puertos con ello. En este caso no tengo ningún firewall configurado por ese motivo no muestra nada.

```
jaime@ubuntu:~$ ufw
ERROR: not enough args
jaime@ubuntu:~$
```

2.- **netstat**: muestra información sobre los subsistemas de red en Ubuntu. Como se ve en la imagen tiene establecida la conexión en la dirección de red que muestra.

```
jaime@ubuntu:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 192.168.0.28:bootpc     192.168.0.1:bootps     ESTABLISHED
udp        0      0 192.168.0.28:bootpc     192.168.0.1:bootps     ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type                   State         I-Node  Path
unix   2      [ ]                   DGRAM                 41320         /run/user/1000/systemd/notif
unix   2      [ ]                   DGRAM                 28800         /run/systemd/journal/syslog
unix  15      [ ]                   DGRAM                 28810         /run/systemd/journal/dev-log
unix   8      [ ]                   DGRAM                 28814         /run/systemd/journal/socket
unix   3      [ ]                   DGRAM                 28786         /run/systemd/notify
unix   3      [ ]                   STREAM                CONNECTED      50818
```

3.- ¿Cómo sabemos a quién pertenece una dirección web (url)?

Para saber a quien pertenece una dirección determinada utilizaremos estos comandos-

1.- **dig**: Nos permite hacer consultas a diferentes servidores de DNS para saber el nombre de una determinada IP.

```
jaime@ubuntu:~$ dig www.google.es

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.es
;; global options: +cmd Esto es informacion sobre la version de nuestro comando
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31293
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
Hemos recibido respuesta y sin errores
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 65494
;; QUESTION SECTION:
;www.google.es.                IN      A
La pregunta realizada

;; ANSWER SECTION:
www.google.es.                83      IN      A      142.250.185.3
La respuesta recibida

;; Query time: 31 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: lun feb 22 03:12:21 PST 2021
;; MSG SIZE rcvd: 58
```

2.- **nslookup**: Herramienta para interactuar con los servidores DNS. Esta herramienta es parecida a la anterior pero su uso está más extendido.

Se puede ejecutar desde nuestra terminal y desde la propia terminal de nslookup.

```
jaime@ubuntu:~$ nslookup goole.es
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   goole.es
Address: 185.53.178.70

jaime@ubuntu:~$ nslookup 8.8.8.8.
Server:      127.0.0.53
Address:     127.0.0.53#53
```

4.- ¿Cómo probamos que podemos acceder a un servidor?

El acceso a los servidores se puede probar de diferentes formas y comandos, aquí probaremos los dos siguientes.

1.- **curl**: Diseñado para verificar las conexiones a la url y gran herramienta para transferir datos. Compatible con la mayoría de los protocolos como: HTTP, HTTPS, FTP, SFTP...

```
jaime@ubuntu:~$ curl telefonica.es
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://www.telefonica.es/">here</a>.</p>
</body></html>
jaime@ubuntu:~$
```

2.- **wget**: usado para recuperar archivos de diferentes servidores web. Va seguido del nombre un archivo y el nombre del servidor donde queramos recuperar el archivo.

```
jaime@ubuntu:~$ wget
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try 'wget --help' for more options.
jaime@ubuntu:~$
```

5.- ¿Otros comandos?

Aquí detallaré una lista de los comandos necesarios para poder instalar las herramientas necesarias y otros comandos utilizados en la MV.

- 1.- **sudo apt update**: Actualiza todos los paquetes de nuestro ubuntu
- 2.- **sudo apt upgrade**: actualiza todas las aplicaciones que tengamos instaladas
- 3.- **sudo apt install net-tools**: Instala herramientas para controlar el subsistema de red de Ubuntu.
- 4.- **clear**: Limpia la consola.
- 5.- **sudo apt install curl**: para probar el comando propuesto en la actividad hemos tenido que instalarlo previamente con este comando.

*Archivo incluido en el repositorio de github:
<https://github.com/jjuarez1986/Despliegues.git>