

# 2º DESARROLLO APLICACIONES WEB



LIDIA MARTÍNEZ CAPITA

ACTIVIDAD 1.

Tarea Individual - Comandos

## ACTIVIDAD 1. TAREA INDIVIDUAL. COMANDOS

### REQUERIMIENTO 1

La administración de un servidor web y/o un servidor de aplicaciones requiere unos conocimientos básicos de comandos de consola que permite visualizar qué está pasando en nuestro servidor. Se pide practicar y crear una guía de uso para las siguientes problemáticas que nos podemos encontrar:

1. ¿Cómo sabemos si tenemos conexión a internet? Pista: `ifconfig`, `ping`
2. ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: `ufw`, `netstat`
3. ¿Cómo sabemos a quién pertenece una dirección web (URL)? Pista: `dig`, `nslookup`
4. ¿Cómo probamos que podemos acceder a un servidor? Pista: `curl`, `wget`
5. ¿Qué otros comandos te han hecho falta?

Valoración: 10 puntos sobre 10.

## GUÍA HOW-TO COMANDOS EN UBUNTU

Comentar que hemos utilizado una máquina virtual (VirtualBox, con el sistema operativo Ubuntu).

En primer lugar, vamos a responder a la primera pregunta, con respecto al modo de saber si tenemos conexión a internet. Es difícil encontrar un equipo Linux que no esté conectado a la red, sea servidor o estación de trabajo. Sin embargo, de vez en cuando se hace necesario diagnosticar fallos, intermitencias o lentitud en la red. Es por ello por lo que utilizamos ciertos comandos en el terminal para ello.

Con el **comando `ifconfig`**, listamos las direcciones IP de todos los dispositivos del equipo, ya que posteriormente con el comando `ping`, mandaremos una señal que deberá ser devuelta por el equipo para comprobar si se encuentra en línea o no.

Por ello inicialmente, una vez iniciada nuestra máquina virtual y abierto el terminal, ponemos el siguiente comando: `ifconfig`.

```
lidia@lidia-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.27 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::9f79:8a9f:1baa:9cc4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e5:9e:82 txqueuelen 1000 (Ethernet)
    RX packets 118237 bytes 174372854 (174.3 MB)
    RX errors 15042 dropped 185 overruns 0 frame 15042
    TX packets 72801 bytes 6010420 (6.0 MB)
    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 (Bucle local)
    RX packets 222 bytes 19492 (19.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 222 bytes 19492 (19.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lidia@lidia-VirtualBox:~$
```

Como se puede ver en la imagen anterior, la IP del equipo es 192.168.0.27. El paso siguiente es usar el **comando ping**, este comando sirve para conocer si una dirección IP específica o host es accesible desde la red o no.

Por lo tanto, en el terminal escribimos ping seguido de la dirección IP. Podemos comprobar la conexión a la red en la propia máquina como se ve en la imagen de abajo, si nos da respuesta y ningún error significa que es accesible.

```
lidia@lidia-VirtualBox:~$ ping 192.168.0.27
PING 192.168.0.27 (192.168.0.27) 56(84) bytes of data:
64 bytes from 192.168.0.27: icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from 192.168.0.27: icmp_seq=2 ttl=64 time=0.092 ms
64 bytes from 192.168.0.27: icmp_seq=3 ttl=64 time=0.096 ms
64 bytes from 192.168.0.27: icmp_seq=4 ttl=64 time=0.096 ms
^C
--- 192.168.0.27 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.022/0.076/0.096/0.031 ms
lidia@lidia-VirtualBox:~$
```

Además, es posible saber si dos equipos se pueden “ver”, para ello hemos probado a realizar ping en nuestra cmd de Windows con la IP del equipo Ubuntu.

```
C:\> Símbolo del sistema

Sufijo DNS específico para la conexión. . :

C:\Users\lidy_>ping 192.168.0.27

Haciendo ping a 192.168.0.27 con 32 bytes de datos:
Respuesta desde 192.168.0.27: bytes=32 tiempo=8ms TTL=64
Respuesta desde 192.168.0.27: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.0.27: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.27: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.0.27:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 8ms, Media = 2ms

C:\Users\lidy_>
```

Además de ping, existe otro comando que nos permite ver la ruta que usa nuestro equipo Linux para conectarse a la red, en este caso, nuestro equipo sale por el router 192.168.0.1. Con esto podríamos responder a la pregunta 5 de la actividad. Por lo que según vayamos usando nuevos comandos lo comentaremos directamente en cada punto.

```
lidia@lidia-VirtualBox:~$ route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso Interfaz
0.0.0.0      192.168.0.1   0.0.0.0      UG    100    0        0 enp0s3
169.254.0.0  0.0.0.0       255.255.0.0  U     1000   0        0 enp0s3
192.168.0.0  0.0.0.0       255.255.255.0 U     100    0        0 enp0s3
lidia@lidia-VirtualBox:~$
```

Comentar también que para la actividad hemos necesitado montar un servidor web, es por ello por lo que vamos a pasar a comentar como lo hemos realizado.

El servidor lo hemos instalado con el siguiente comando: **apt-get install apache2**.

Vamos ahora a comprobar el estado de nuestro servidor web. Para ello introducimos el siguiente

comando: **sudo systemctl status apache2**

Si todo ha ido bien en la instalación nos aparecerá el siguiente mensaje:

```

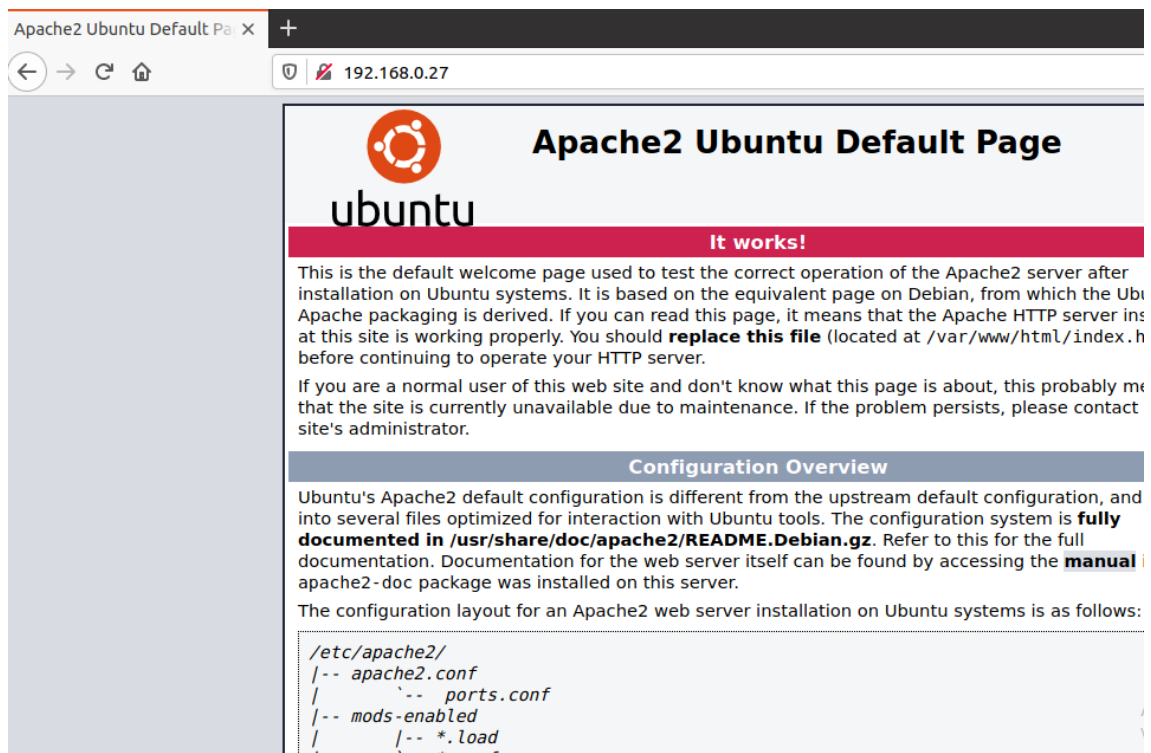
lidia@lidia-VirtualBox:~$ sudo systemctl status apache2
[sudo] contraseña para lidia:
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2021-01-18 11:03:24 CET; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2914 (apache2)
     Tasks: 55 (limit: 2316)
    Memory: 4.7M
     CGroup: /system.slice/apache2.service
            └─2914 /usr/sbin/apache2 -k start
              └─2916 /usr/sbin/apache2 -k start
                └─2917 /usr/sbin/apache2 -k start

ene 18 11:03:24 lidia-VirtualBox systemd[1]: Starting The Apache HTTP Server...
ene 18 11:03:24 lidia-VirtualBox apachectl[2913]: AH00558: apache2: Could not r
ene 18 11:03:24 lidia-VirtualBox systemd[1]: Started The Apache HTTP Server.
...skipping...
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2021-01-18 11:03:24 CET; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2914 (apache2)
     Tasks: 55 (limit: 2316)
    Memory: 4.7M
     CGroup: /system.slice/apache2.service
            └─2914 /usr/sbin/apache2 -k start
              └─2916 /usr/sbin/apache2 -k start
                └─2917 /usr/sbin/apache2 -k start

ene 18 11:03:24 lidia-VirtualBox systemd[1]: Starting The Apache HTTP Server...

```

Para probar el servidor web, debemos saber nuestra IP, pero como en el apartado anterior la hemos obtenido gracias al comando `ifconfig`, simplemente, lo que tenemos que hacer es abrir el navegador y poner nuestra IP. Si todo sale bien, como podemos comprobar en la imagen posterior es que ya tenemos instalado nuestro servidor.



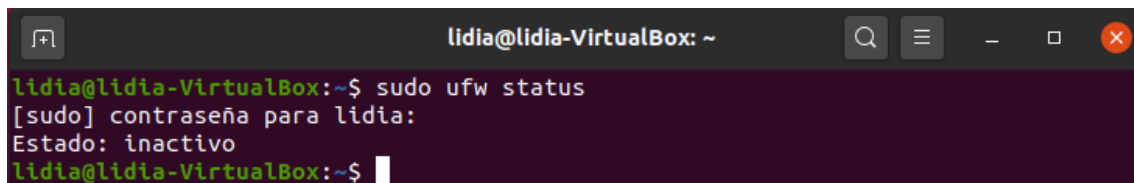
A continuación, para saber si nuestro servidor es accesible desde internet podemos utilizar los siguientes comandos: `ufw` y `netstat`.

Para ello tenemos que introducirnos en el tema del firewall. Un firewall que funcione correctamente es la parte más crucial de la seguridad completa del sistema Linux. Por defecto, la distribución de Ubuntu viene con una herramienta de configuración de firewall llamada UFW (Firewall sin complicaciones), es una herramienta de línea de comandos para configurar y administrar un firewall en las distribuciones de Ubuntu.

Por defecto, el firewall UFW niega todas las conexiones entrantes y solo permite todas las conexiones salientes del servidor. Esto significa que nadie puede acceder al servidor, a menos que abra específicamente el puerto, mientras que todos los servicios o aplicaciones en ejecución en nuestro servidor pueden acceder a la red externa.

Las políticas de firewall UFW predeterminadas se colocan en el **/etc/default/ufw**.

Por otra parte, UFW viene desactivado por defecto, para evitar posibles problemas de acceso al servidor antes de su configuración, por lo que el primer paso consistirá en definir qué tipos de conexión deseamos habilitar, para posteriormente activar el firewall. Podemos comprobar el estado del firewall con el **comando ufw status**. En un principio deberíamos ver que está inactivo. Se muestra a continuación en nuestra máquina virtual.

A terminal window titled 'lidia@lidia-VirtualBox: ~' with standard window controls. The terminal shows the command 'sudo ufw status' being executed. The output is: '[sudo] contraseña para lidia: Estado: inactivo'. The prompt returns to 'lidia@lidia-VirtualBox:~\$' with a cursor.

```
lidia@lidia-VirtualBox:~$ sudo ufw status
[sudo] contraseña para lidia:
Estado: inactivo
lidia@lidia-VirtualBox:~$
```

Una vez que sabemos que está inactivo y antes de activar el firewall, es importante configurar las conexiones permitidas, habilitando al menos el acceso por SSH, para que no perdamos el acceso al servidor. Para saber qué tipo de conexiones podemos activar o desactivar en el servidor, ejecutamos el comando siguiente: **sudo ufw app list**.

```
lidia@lidia-VirtualBox: ~  
Enabling module deflate.  
Enabling module status.  
Enabling module reqtimeout.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.  
Procesando disparadores para ufw (0.36-6) ...  
Procesando disparadores para systemd (245.4-4ubuntu3.2) ...  
Procesando disparadores para man-db (2.9.1-1) ...  
Procesando disparadores para libc-bin (2.31-0ubuntu9) ...  
lidia@lidia-VirtualBox:~$ sudo ufw app list  
Aplicaciones disponibles:  
  Apache  
  Apache Full  
  Apache Secure  
  CUPS  
lidia@lidia-VirtualBox:~$
```

Por tanto, la salida de ese comando nos indicará qué aplicaciones localiza el firewall como posibles de configurar en nuestro sistema.

Si por ejemplo queremos permitir el acceso al servidor por SSH debemos habilitar las conexiones con la aplicación Apache. Esta aplicación nos permite el acceso por HTTP por los puertos comunes de HTTP (80) y HTTPS (443).

Esto lo conseguimos con el comando siguiente:

```
lidia@lidia-VirtualBox:~$ sudo ufw allow "Apache"  
Reglas actualizadas  
Reglas actualizadas (v6)  
lidia@lidia-VirtualBox:~$
```

Finalmente, una vez realizada la configuración, podemos activar ya el firewall, con el **comando ufw enable**, también podemos comprobar la configuración de UFW con el **comando ufw status**. Ahora deberíamos ver que el firewall se encuentra activo y además podremos obtener un listado de aplicaciones habilitadas.



```

lidia@lidia-VirtualBox:~$ sudo su -
root@lidia-VirtualBox:~# ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
root@lidia-VirtualBox:~# sudo ufw status
Estado: activo

Hasta          Acción          Desde
-----
Apache         ALLOW           Anywhere
Apache (v6)    ALLOW           Anywhere (v6)

root@lidia-VirtualBox:~#

```

Adicionalmente también se pueden permitir o denegar conexiones en todos los puertos desde una dirección IP específica.

En segundo lugar, con el **comando netstat** podemos saber que está pasando en nuestra red. El caso que nos interesa, a pesar de que este comando tiene varias salidas diferentes para varios usos, es el de mostrar las conexiones de red.

Es decir, el uso de netstat nos sirve para poder determinar nuestras conexiones tanto internas (localhost) como externas. Una salida típica de este comando introduciendo el parámetro “-ano” sería:

```

lidia@lidia-VirtualBox:~$ netstat -ano
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Envíad Dirección local      Dirección remota      Estado      Temporizador
tcp    0      0 127.0.0.53:53         0.0.0.0:*              ESCUCHAR    apagado (0.00/0/0)
tcp    0      0 127.0.0.1:631         0.0.0.0:*              ESCUCHAR    apagado (0.00/0/0)
tcp    0      0 192.168.0.27:56212    54.192.105.7:443      ESTABLECIDO mantener vivo (7,02/0/0)
tcp    0      0 192.168.0.27:54926    34.107.221.82:80      ESTABLECIDO mantener vivo (4,46/0/0)
tcp    0      0 192.168.0.27:52488    35.244.247.133:443    ESTABLECIDO apagado (0.00/0/0)
tcp    0      0 192.168.0.27:33462    142.250.184.3:80      ESTABLECIDO mantener vivo (5,24/0/0)
tcp    0      0 192.168.0.27:33438    54.192.105.10:443     ESTABLECIDO mantener vivo (1,90/0/0)
tcp    0      0 192.168.0.27:33458    142.250.184.3:80      ESTABLECIDO mantener vivo (1,65/0/0)
tcp    0      0 192.168.0.27:53052    54.192.105.48:443     ESTABLECIDO apagado (0.00/0/0)
tcp    0      0 192.168.0.27:45244    93.184.220.29:80      ESTABLECIDO mantener vivo (3,43/0/0)
tcp    0      0 192.168.0.27:54924    34.107.221.82:80      ESTABLECIDO mantener vivo (4,46/0/0)
tcp    0      0 192.168.0.27:57240    52.38.35.234:443      ESTABLECIDO mantener vivo (501,58/0/0)
tcp    0      0 192.168.0.27:52220    13.33.232.106:443     ESTABLECIDO mantener vivo (0,48/0/0)
tcp    0      0 192.168.0.27:49530    35.244.181.201:443    ESTABLECIDO apagado (0.00/0/0)
tcp    0      0 192.168.0.27:34854    54.192.105.116:443    ESTABLECIDO mantener vivo (0,48/0/0)
tcp    0      0 192.168.0.27:45266    93.184.220.29:80      ESTABLECIDO mantener vivo (2,41/0/0)
tcp    0      0 192.168.0.27:50050    34.98.75.36:443       ESTABLECIDO apagado (0.00/0/0)
tcp    0      0 192.168.0.27:41220    44.238.41.205:443     TIME WAIT    tiempo de espera (20,20/0/0)
tcp6   0      0 :::80                 :::*                   ESCUCHAR    apagado (0.00/0/0)
tcp6   0      0 :::1:631              :::*                   ESCUCHAR    apagado (0.00/0/0)
udp    0      0 0.0.0.0:631           0.0.0.0:*              apagado (0.00/0/0)
udp    0      0 0.0.0.0:41818         0.0.0.0:*              apagado (0.00/0/0)
udp    0      0 127.0.0.53:53         0.0.0.0:*              apagado (0.00/0/0)
udp    0      0 192.168.0.27:68       192.168.0.1:67        ESTABLECIDO apagado (0.00/0/0)
udp    0      0 0.0.0.0:5353          0.0.0.0:*              apagado (0.00/0/0)
udp6   0      0 :::37786              :::*                   apagado (0.00/0/0)
udp6   0      0 :::5353               :::*                   apagado (0.00/0/0)
raw6   0      0 :::58                 :::*                   7               apagado (0.00/0/0)

```

Este comando como vemos nos ofrece información en varias columnas: en la columna de más a la izquierda vemos la columna Proto, es decir el protocolo establecido para establecer la comunicación.

Aquí fundamentalmente veremos tres tipos de protocolos: ICMP, UDP y TCP.

Dirección Local: aquí nos aparece el número de IP local que establece una comunicación de salida, podemos poner un símil. Esto es lo mismo que los móviles: todos los móviles tienen un número asignado que es utilizado para establecer comunicaciones con él. Cuando nosotros llamamos a alguien tenemos



comunicaciones salientes, y si nos llaman tenemos comunicaciones entrantes.

La columna de estado nos indica en qué estado se encuentra la comunicación entre procesos, y veremos diferentes tipos (en nuestro ejemplo no salen todos):

LISTENING significa que detrás de ese puerto hay un proceso esperando que alguien hable con él, es decir, preparado para aceptar comunicaciones. En este caso podemos ver que el puerto 80 del servidor (apache) está listo para escuchar.

IMED\_WAIT, estamos esperando a que el servidor acepte nuestra petición de cerrar comunicación.

ESTABLISHED significa que el proceso que está detrás de ese puerto ya está hablando con algo o alguien; la columna de dirección remota nos indica con quién habla.

Como conclusión, podemos comentar que cuando un puerto se abre y recibe el estado de “Listen” y espera a que se detecte una conexión, el problema principal de estos puertos abiertos es que, de esta manera, se les da la oportunidad a terceros de introducir malware en el sistema.

Por ello, es recomendable comprobar regularmente los puertos abiertos del sistema, tarea en la que destaca el comando netstat.

En referencia a la tercera pregunta, para saber a quién pertenece una dirección web, cada dominio de internet tiene asociada una cantidad de datos públicos que se pueden consultar, como, por ejemplo quién es el propietario, cómo contactar con él o en qué máquina y país está alojado.

Es por ello que **comando dig** también nos puede servir para dicha tarea. Podemos ver más abajo que nos muestra una gran cantidad de información y por tanto es una herramienta para para consultar servidores DNS. Se utiliza a menudo para diagnosticar problemas con la resolución de nombres (traducción de nombres de host a direcciones IP) debido a su flexibilidad, facilidad de uso y claridad en la salida.

Por ejemplo, vamos a verlo con la web de [www.apple.com](http://www.apple.com)

```
lidia@lidia-VirtualBox:~$ dig www.apple.com

;<<> DiG 9.16.1-Ubuntu <<> www.apple.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 47258
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.apple.com.                IN      A

;; ANSWER SECTION:
www.apple.com. 1595 IN      CNAME   www.apple.com.edgekey.net.
www.apple.com.edgekey.net. 6492 IN      CNAME   www.apple.com.edgekey.net.globalredir.akadns.net.
www.apple.com.edgekey.net.globalredir.akadns.net. 2015 IN      CNAME   e6858.dsce9.akamaiedge.net.
e6858.dsce9.akamaiedge.net. 6 IN      A      104.83.215.89

;; Query time: 24 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: vie ene 22 19:59:03 CET 2021
;; MSG SIZE rcvd: 193
```

De ello podemos sacar la siguiente información.

En la primera línea se muestra la versión de dig y la dirección consultada.

En la siguiente línea aparece Global options, que muestra las opciones que se han aplicado a todas las consultas de dominio. En nuestro ejemplo, era solo la opción predeterminada +cmd (comando).

La siguiente línea que aparece es status donde se muestra Noerror, es decir, que no hubo errores y la solicitud se resolvió correctamente.

El ID 47258 es un ID aleatorio que une la solicitud y la respuesta.

La opción Query por su parte, que en este caso es igual a 1, quiere decir el número de consultas en esta sesión y answer es por tanto el número de respuestas en esta consulta, que es 4.

La siguiente instrucción que de la que se ha hablado en este apartado ha sido **nslookup** otra opción para realizar esta tarea.

Con este comando también podemos localizar la dirección IP de un host o dominio.

```
lidia@lidia-VirtualBox:~$ nslookup www.apple.com
Server:      127.0.0.53
Address:     127.0.0.53#53

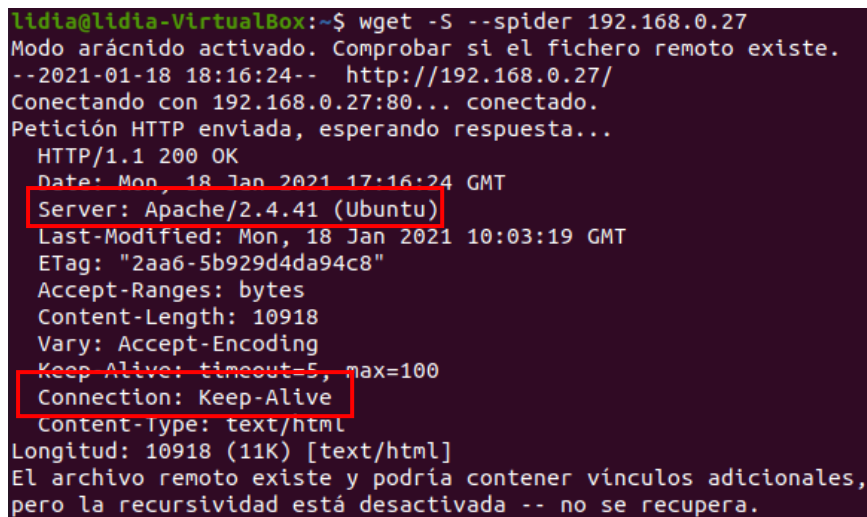
Non-authoritative answer:
www.apple.com canonical name = www.apple.com.edgekey.net.
www.apple.com.edgekey.net canonical name = www.apple.com.edgekey.net.globalredir.akadns.net.
www.apple.com.edgekey.net.globalredir.akadns.net canonical name = e6858.dsce9.akamaiedge.net.
Name:   e6858.dsce9.akamaiedge.net
Address: 104.83.215.89
Name:   e6858.dsce9.akamaiedge.net
Address: 2a02:26f0:b1:18c::1aca
Name:   e6858.dsce9.akamaiedge.net
Address: 2a02:26f0:b1:189::1aca
```

Con respecto a la última pregunta de cómo podemos probar que podemos acceder a un servidor, existen diferentes instrucciones o comandos para verificar si un sitio web está activo o inactivo desde la terminal de Ubuntu.

Uno de ellos es el **comando wget**, una herramienta que nos permite la descarga de contenidos desde servidores web de una forma simple y que recupera archivos usando HTTP, HTTPS y los protocolos de Internet más utilizados.

Vamos a ver un ejemplo.

Si ejecutamos el **comando wget -S --spider 192.168.0.27** (dirección IP, hemos añadido más complementos como la -S ya que lo que queremos es mostrar la respuesta del servidor), junto con spider para que no descargue nada seguido de la IP.



```
lidia@lidia-VirtualBox:~$ wget -S --spider 192.168.0.27
Modo arácnido activado. Comprobar si el fichero remoto existe.
--2021-01-18 18:16:24-- http://192.168.0.27/
Conectando con 192.168.0.27:80... conectado.
Petición HTTP enviada, esperando respuesta...
HTTP/1.1 200 OK
Date: Mon, 18 Jan 2021 17:16:24 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Mon, 18 Jan 2021 10:03:19 GMT
ETag: "2aa6-5b929d4da94c8"
Accept-Ranges: bytes
Content-Length: 10918
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
Longitud: 10918 (11K) [text/html]
El archivo remoto existe y podría contener vínculos adicionales,
pero la recursividad está desactivada -- no se recupera.
```

Como podemos observar en la imagen ha sido posible establecer la conexión.

De la misma manera, es posible utilizar el **comando curl** que sirve para conectarnos con servidores para trabajar con ellos. Este comando está diseñado para funcionar sin interacción del usuario.

Con las opciones siguientes obtendremos información acerca de la dirección IPv6 utilizada, el puerto y los certificados aplicados. Vemos que está conectado al servidor a través del puerto 80.

```
lidia@lidia-VirtualBox:~$ curl -I 192.168.0.27 -v
* Trying 192.168.0.27:80...
* TCP_NODELAY set
* Connected to 192.168.0.27 (192.168.0.27) port 80 (#0)
> HEAD / HTTP/1.1
> Host: 192.168.0.27
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
HTTP/1.1 200 OK
< Date: Mon, 18 Jan 2021 17:20:37 GMT
Date: Mon, 18 Jan 2021 17:20:37 GMT
< Server: Apache/2.4.41 (Ubuntu)
Server: Apache/2.4.41 (Ubuntu)
< Last-Modified: Mon, 18 Jan 2021 10:03:19 GMT
Last-Modified: Mon, 18 Jan 2021 10:03:19 GMT
< ETag: "2aa6-5b929d4da94c8"
ETag: "2aa6-5b929d4da94c8"
< Accept-Ranges: bytes
Accept-Ranges: bytes
< Content-Length: 10918
Content-Length: 10918
< Vary: Accept-Encoding
Vary: Accept-Encoding
< Content-Type: text/html
Content-Type: text/html

<
* Connection #0 to host 192.168.0.27 left intact
lidia@lidia-VirtualBox:~$
```

Finalmente, se añade un pequeño resumen con la definición y uso de cada comando.

## RESUMEN COMANDOS

1. COMANDO IFCONFIG: listamos las direcciones IP de todos los dispositivos del equipo.
2. COMANDO PING: comprobar la conexión con un equipo específico.
3. COMANDO UFW: configurar y administrar un firewall.
4. COMANDO NETSTAT: listar las conexiones activas de un equipo, tanto entrantes como salientes.
5. COMANDO DIG: realizar consulta a los servidores DNS para solicitar información sobre direcciones de host.
6. COMANDO NSLOOKUP: encontrar la dirección IP de un equipo determinado o también realizar una búsqueda DNS inversa
7. COMANDO CURL verificar la conectividad a una URL o conectarse a un servidor para trabajar con él.
8. COMANDO WGET: recuperar contenido y archivos de varios servidores web.