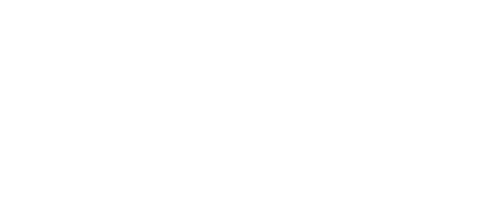
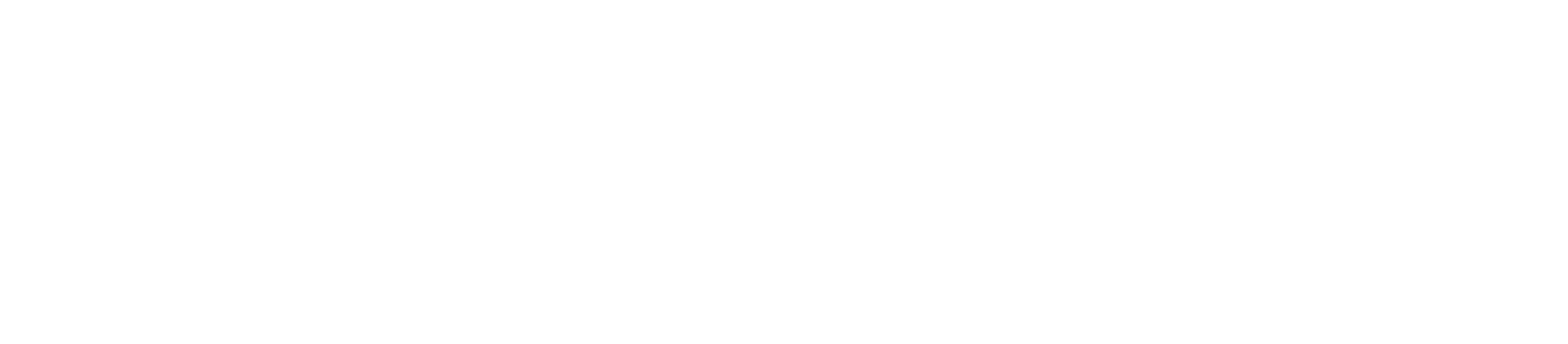
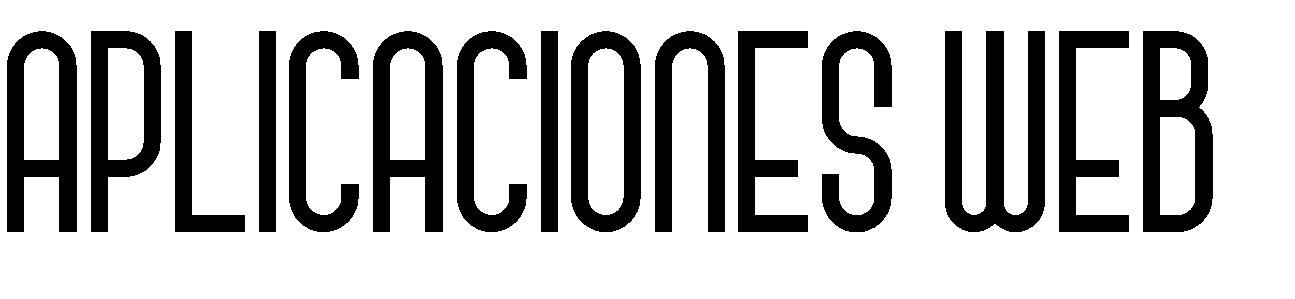
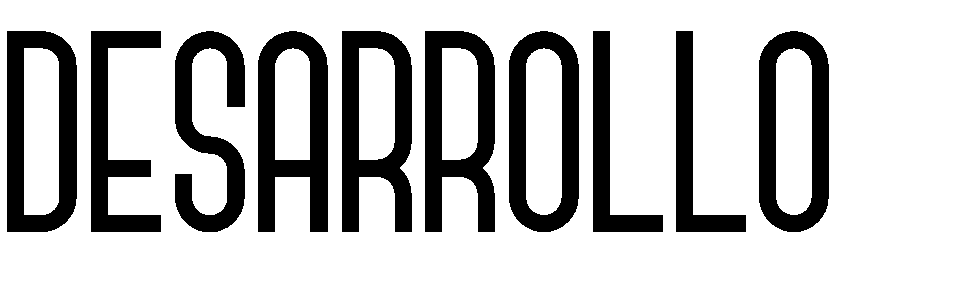
2º



1.

Página1

**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**

# ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: ufw, netstat

1. **¿Cómo sabemos a quién pertenece una dirección web (URL)? Pista: dig, nslookup**

# ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget

1. **¿Qué otros comandos te han hecho falta? Valoración: 10 puntos sobre 10.**

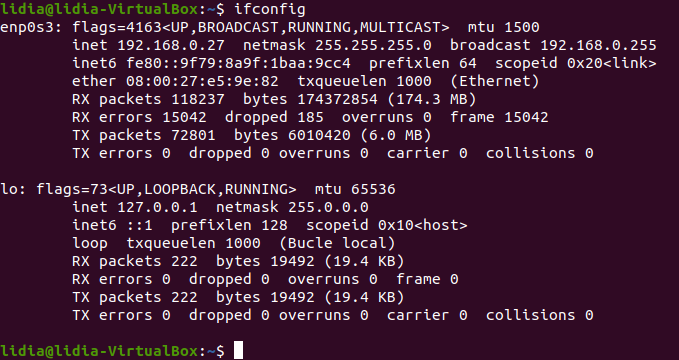
**GUŨA HOW-TO COMANDOS EN UBUNTU**

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.

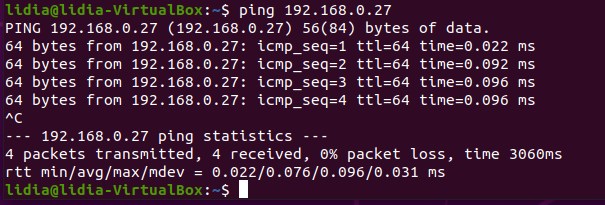
Con el **comando ifconfig,** listamos las direcciones IP de todos los dispositivos del equipo, ya que posteriormente con el comando ping, mandaremos una señal que deberá ser devuelta por el equipo para comprobar si se encuentra en línea o no.

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

Página2

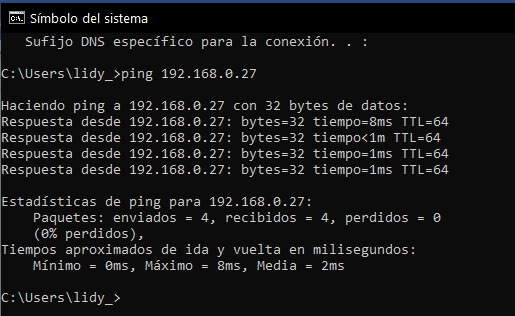


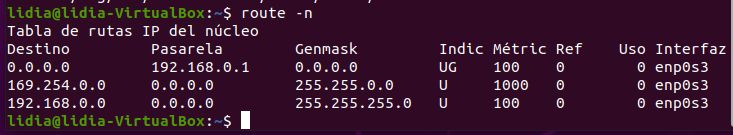
Como se puede ver en la imagen anterior, la IP del equipo es 192.168.0.27. El paso siguiente es usar el **comando ping,** este comando sirve para conocer si una dirección IP específica o host es accesible desde la red o no.

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.

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.

Página3



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.

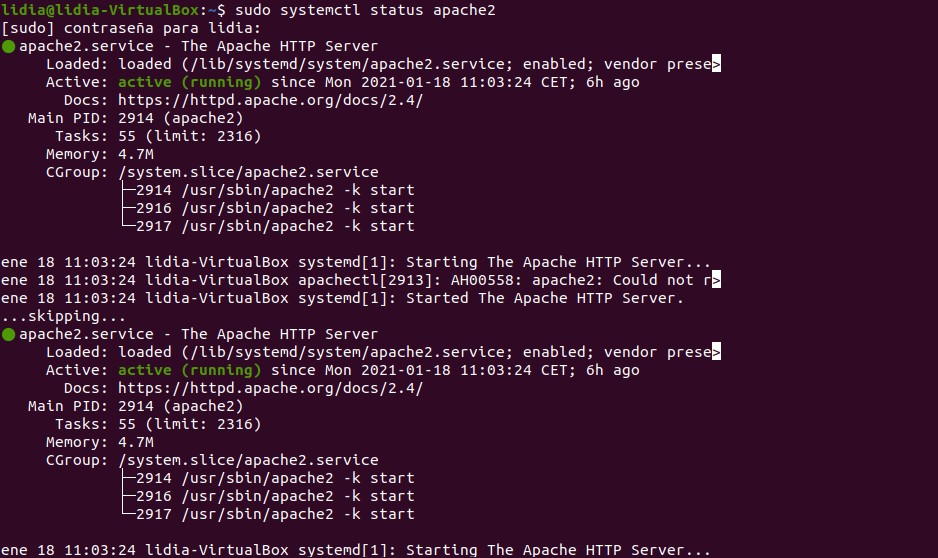
Comentar también que para la actividad hemos necesitado montar un servidor web, es por ello por lo que vamos a pasar a comentar como lo hemos realizado.

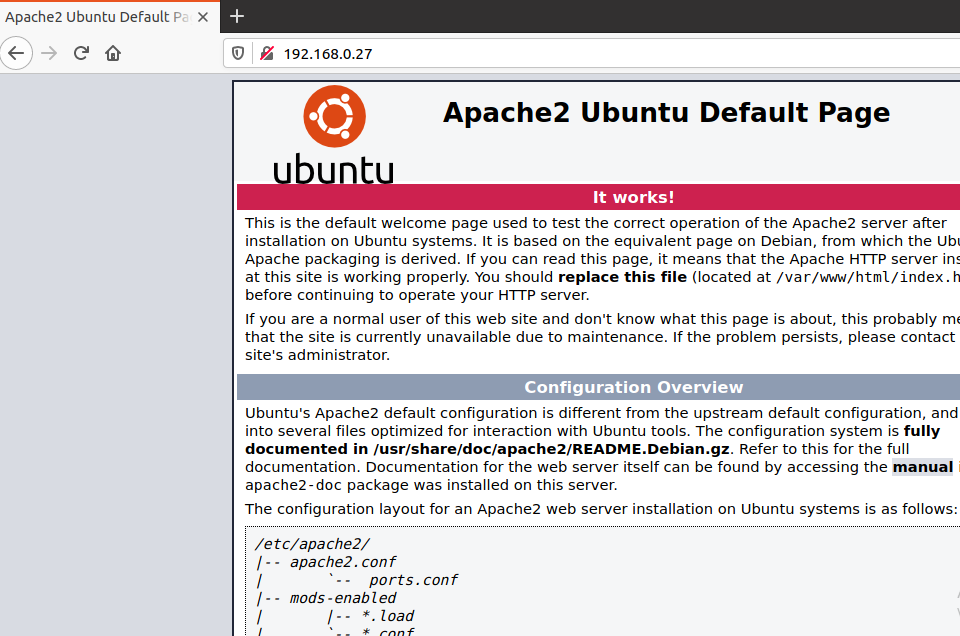
El servidor lo hemos instalado con el siguiente comando: **apt-get install apache2.**

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:

Página4



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 si nuestro servidor es accesible desde internet podemos utilizar los siguientes **comandos: ufw y netstat**.

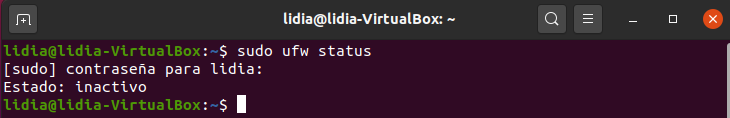
Página5

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.

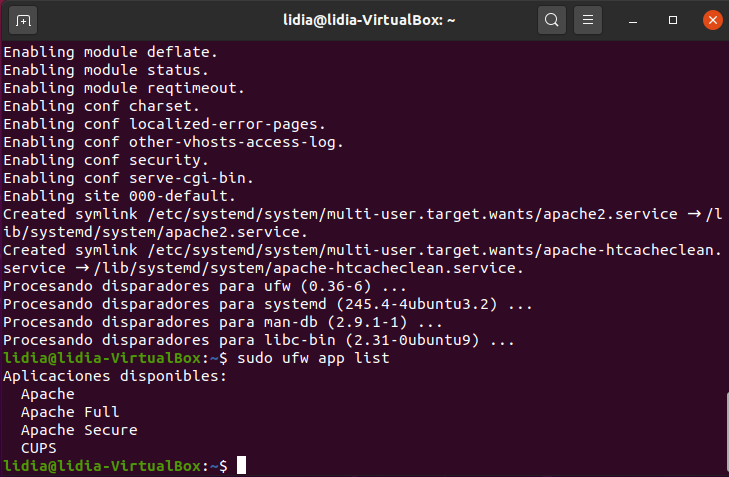
Las políticas de firewall UFW predeterminadas se colocan en el **/etc/default/ufw.**

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.



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.**

Página6

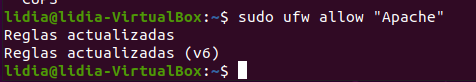


Por tanto, la salida de ese comando nos indicará qué aplicaciones localiza el firewall como posibles de configurar en nuestro sistema.

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

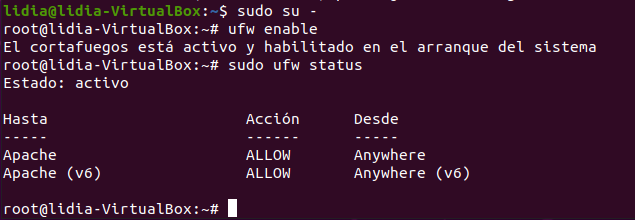
1. y HTTPS (443).

Esto lo conseguimos con el comando siguiente:



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.

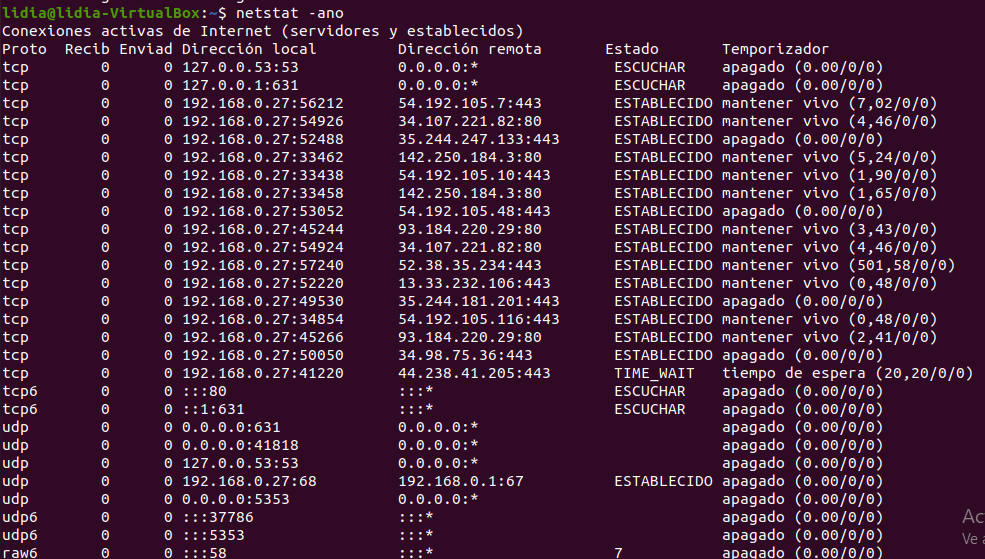
Página7



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:



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

Página8

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.

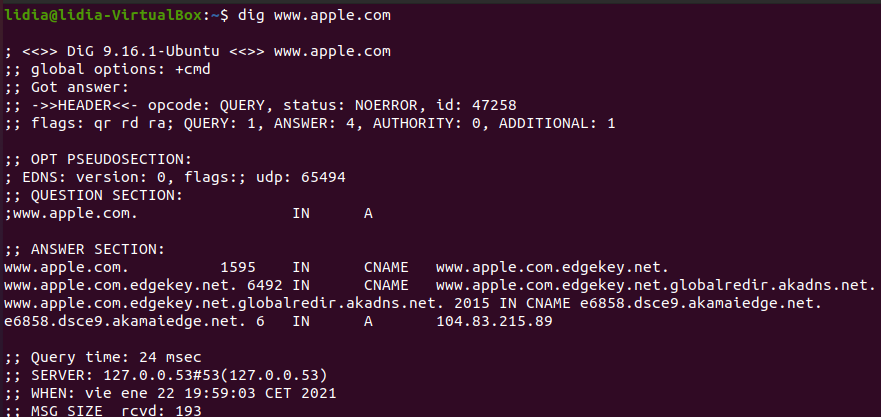
Por ello, es recomendable comprobar regularmente los puertos abiertos del sistema, tarea en la que destaca el comando netstat.

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.

Página9

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



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).

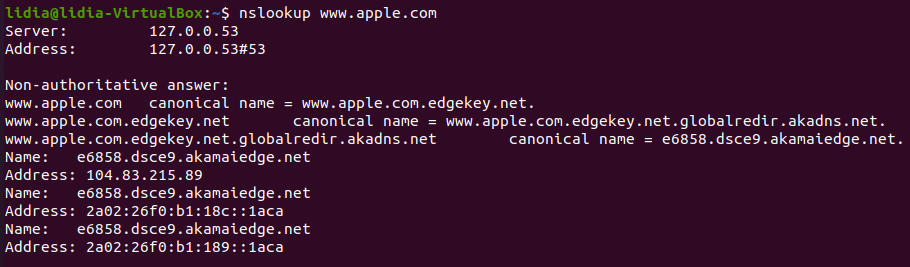
La siguiente línea que aparece es status donde se muestra Noerror, es decir, que no hubo errores y la solicitud se resolvió correctamente.

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.



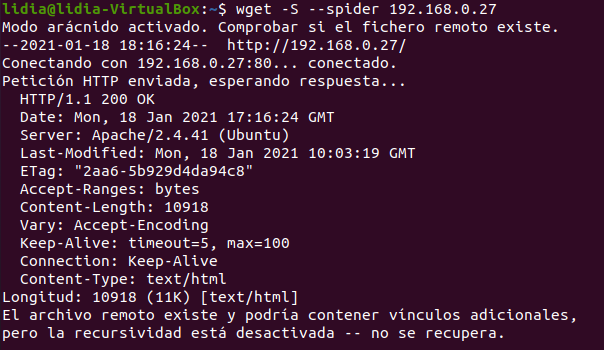
Página10

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.

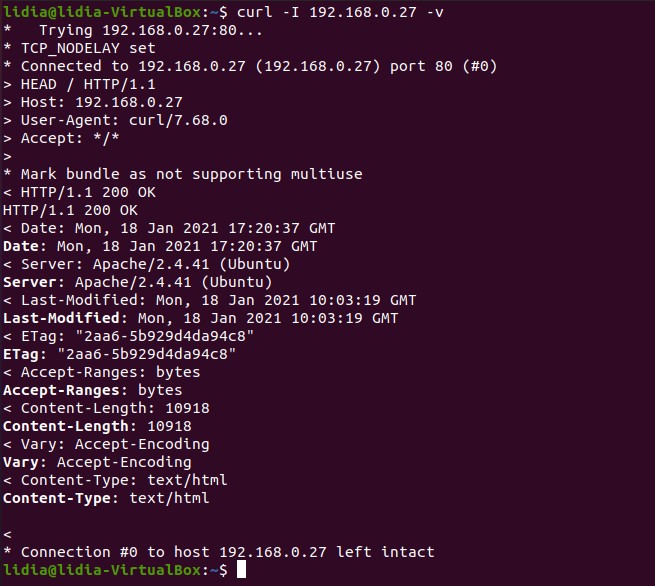


Como podemos observar en la imagen ha sido posible establecer la conexión.

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.

Página11



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 especifico.
  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.