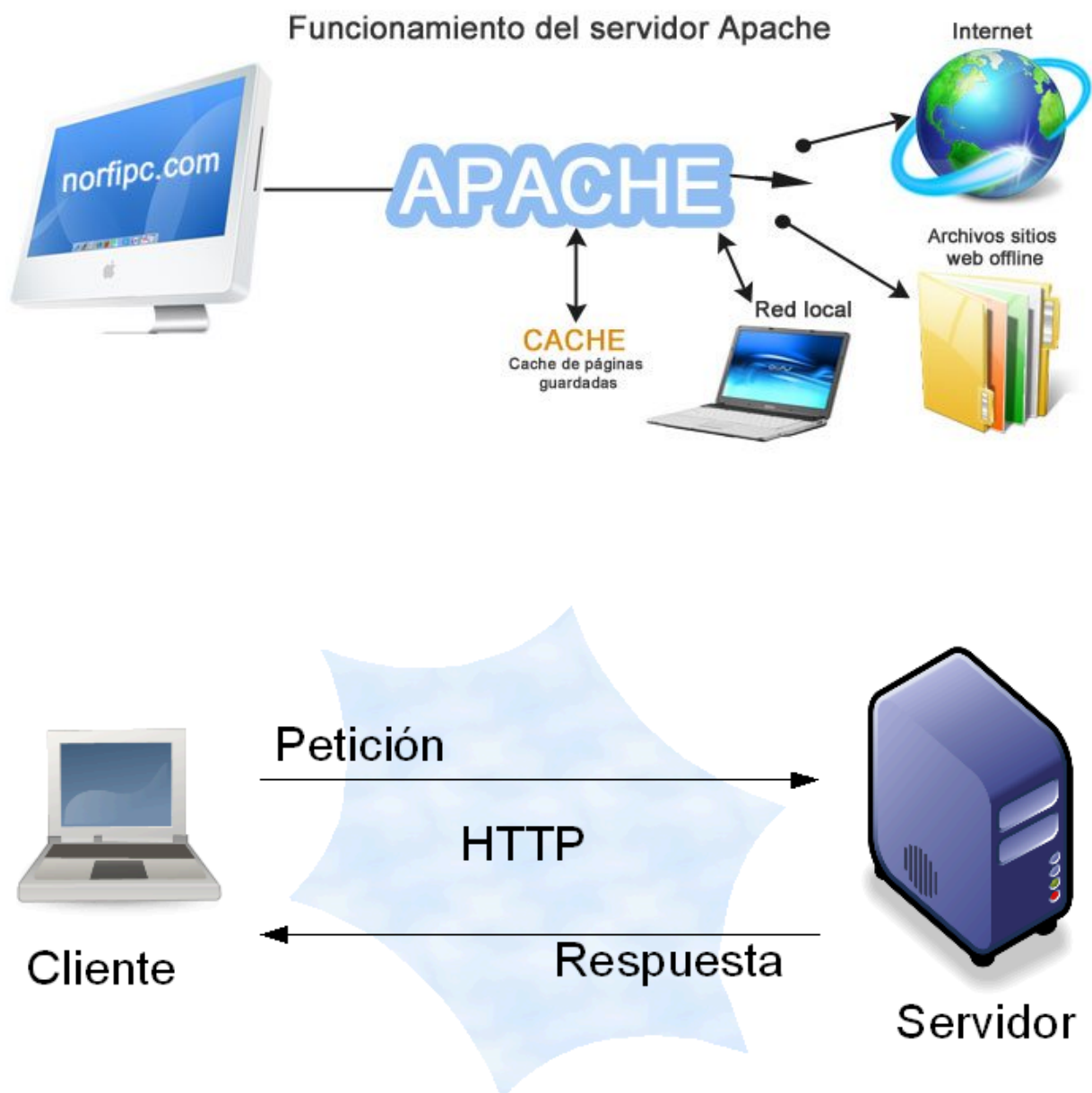


Configuración avanzada de servidores web y HTTPS



Mario Camacho Prieto
2º ASIR
Arquitectura en la Nube

ÍNDICE

- PARTE 1: Instalación y Configuración de Apache

-Preparación del Sistema

- Actualización de paquetes del sistema (sudo apt update && sudo apt upgrade).

-Instalación de Apache

- Instalación del servidor web Apache (sudo apt install apache2 -y).

-Configuración de Puertos de Apache

- Modificación del archivo ports.conf para cambiar el puerto de escucha a 8080.
- Modificación del archivo 000-default.conf para establecer el VirtualHost en el puerto 8080.

-Instalación de PHP

- Instalación de PHP y el módulo de Apache (sudo apt install php libapache2-mod-php -y).
- Reinicio del servicio Apache para aplicar cambios (sudo systemctl restart apache2).

-Verificación de Apache (Puerto 8080)

- Comprobación del estado del servicio (sudo systemctl status apache2).
- Instalación de net-tools para verificar puertos.
- Confirmación de que Apache está escuchando en el puerto 8080 (sudo netstat -tulpn | grep 8080).
- Creación de un archivo info.php para pruebas.
- Verificación del funcionamiento de PHP con curl http://localhost:8080/info.php.

- PARTE 2: Instalación y Configuración de Nginx

-Instalación de Nginx

- Instalación del servidor web Nginx (sudo apt install nginx -y).

-Configuración de Puertos de Nginx

- Modificación del archivo de configuración default de Nginx para cambiar el puerto de escucha a 8081.

-Contenido Web

- Creación de una página index.html personalizada para Nginx.

-Verificación de Nginx (Puerto 8081)

- Reinicio del servicio Nginx (`sudo systemctl restart nginx`).
- Comprobación del estado del servicio (`sudo systemctl status nginx`).
- Confirmación del puerto 8081 (`sudo netstat -tulpn | grep nginx`).
- Verificación del contenido HTML con `curl http://localhost:8081`.

- PARTE 3: Instalación y Configuración de Caddy

-Instalación de Prerrequisitos

- Instalación de paquetes necesarios (`debian-keyring, apt-transport-https, curl`).

-Configuración del Repositorio de Caddy

- Importación de la clave GPG del repositorio.
- Adición de la fuente del repositorio de Caddy (`caddy-stable.list`).

-Instalación de Caddy

- Actualización de paquetes e instalación de Caddy (`sudo apt update && sudo apt install caddy -y`).

-Configuración de Caddy

- Creación del directorio web (`sudo mkdir -p /var/www/caddy`).
- Creación de un archivo `README.md` como página de bienvenida.
- Descarga de una imagen de prueba.
- Edición del `Caddyfile` para configurar el servidor:
 - Establecimiento del puerto de escucha en 8082.
 - Definición del directorio raíz (`root /var/www/caddy`).
 - Habilitación del explorador de archivos (`file_server browse`).

-Verificación de Caddy (Puerto 8082)

- Reinicio del servicio Caddy (`sudo systemctl restart caddy`).
- Comprobación del estado del servicio (`sudo systemctl status caddy`).
- Confirmación del puerto 8082 (`sudo netstat -tulpn | grep caddy`).
- Verificación con `curl http://localhost:8082/` y `curl http://localhost:8082/README.md`.

- PARTE 4: Configuración de HTTPS con Certbot en Apache

-Instalación de Certbot

- Instalación de Certbot y el plugin de Apache (`sudo apt install certbot python3-certbot-apache -y`).

-Generación de Certificado (Autofirmado)

- Creación de un certificado autofirmado con openssl.

-Configuración de SSL en Apache

- Habilitación del módulo SSL de Apache (sudo a2enmod ssl).
- Configuración de un VirtualHost para HTTPS en el puerto 8443.
- Habilitación del sitio SSL (sudo a2ensite default-ssl.conf).

-Verificación de HTTPS (Puerto 8443)

- Reinicio del servicio Apache (sudo systemctl restart apache2).
- Prueba de conexión HTTPS con curl -i -k https://localhost:8443.

- PARTE 5: Verificación Final de los 3 Servidores

-Verificación Individual de Servicios

- Apache: Comprobación de estado (status) y puertos (netstat) (443 y 8443).
- Nginx: Comprobación de estado (status) y puertos (netstat) (80 y 8081).
- Caddy: Comprobación de estado (status) y puertos (netstat) (2019 y 8082).

-Verificación General

- Revisión de todos los puertos de los servidores (8081, 8082, 8443) con un solo comando netstat.
- Prueba final de acceso a uno de los puertos (curl http://localhost:8081).

PARTE 1: INSTALACIÓN Y CONFIGURACIÓN DE APACHE

He cogido y puesto este comando para actualizar la lista de paquetes y mejorar el sistema a las últimas versiones pero no he puesto el “-y” para poder confirmarlo, ya que no se porque pero me daba el código erróneo.

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

localhost@localhost-VirtualBox:~$ sudo apt update && sudo apt upgrade
[sudo] contraseña para localhost:
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 78 paquetes. Ejecute «apt list --upgradable» para verlos.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
```

Con esto lo que voy a hacer es instalar el servidor web de Apache en el sistema de la MV.

```
localhost@localhost-VirtualBox:~$ sudo apt install apache2 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 liba
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
```

Con este comando “sudo nano /etc/apache2/ports.conf” vamos a abrir el archivo de configuración de puertos para cambiar el listen 80 a listen 8080 (se hace para evitar conflictos si el puerto 80 ya está en uso por otro servicio o para permitir la ejecución del servidor como un usuario sin privilegios).

```
localhost@localhost-VirtualBox:~$ sudo nano /etc/apache2/ports.conf

GNU nano 7.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Este comando sirve para definir cómo el servidor web maneja las solicitudes entrantes en el puerto 80 para el sitio predeterminado, especificando la raíz del documento, el correo electrónico del administrador del servidor y las ubicaciones de los archivos de registro.

```
localhost@localhost-VirtualBox:~$ sudo nano /etc/apache2/sites-available/000-default.conf

GNU nano 7.2 /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

^C Ayuda      ^O Guardar    ^W Buscar    ^K Cortar    ^T Ejecutar   ^C Ubicación
Mostrar aplicaciones Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^_ Ir a línea
```

Este comando se utiliza para instalar el módulo de PHP para Apache (libapache2-mod-php), junto con todas sus dependencias, lo que permite que el servidor web Apache pueda interpretar y ejecutar código PHP.

```
localhost@localhost-VirtualBox:~$ sudo apt install php libapache2-mod-php -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 liblvm2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
 libapache2-mod-php8.3 libsodium23 php-common php8.3 php8.3-cli php8.3-common php8.3-opcache php8.3-readline
Paquetes sugeridos:
 php-pear
Se instalarán los siguientes paquetes NUEVOS:
 libapache2-mod-php libapache2-mod-php8.3 libsodium23 php php-common php8.3 php8.3-cli php8.3-common php8.3-opcache
 php8.3-readline
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 8 no actualizados.
Se necesita descargar 5.084 kB de archivos.
Se utilizarán 22,8 MB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 php-common all 2:93ubuntu2 [13,9 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-common amd64 8.3.6-0ubuntu0.24.04.5 [740 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-opcache amd64 8.3.6-0ubuntu0.24.04.5 [371 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-readline amd64 8.3.6-0ubuntu0.24.04.5 [13,5 kB]
]
```

El comando “sudo systemctl restart apache2” se utiliza para reiniciar completamente el servicio del servidor web Apache (apache2). Esto se hace generalmente después de modificar archivos de configuración para que los cambios surtan efecto, o para solucionar problemas menores de funcionamiento del servidor.

```
localhost@localhost-VirtualBox:~$ sudo systemctl restart apache2
[sudo] contraseña para localhost:
localhost@localhost-VirtualBox:~$
```

Este tipo de comando todavía no me indica en qué puerto se está ejecutando, ya que lo digo luego pero sí me está indicando que está funcionando correctamente.

```
localhost@localhost-VirtualBox:~$ sudo systemctl status apache2
Warning: The unit file, source configuration file or drop-ins of apache2.service changed
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-10-16 14:13:16 CEST; 2min 46s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3550 (apache2)
    Tasks: 6 (limit: 4601)
  Memory: 10.6M (peak: 11.4M)
     CPU: 55ms
    CGroup: /system.slice/apache2.service
            └─3550 /usr/sbin/apache2 -k start
              └─3552 /usr/sbin/apache2 -k start
                └─3553 /usr/sbin/apache2 -k start
                  └─3554 /usr/sbin/apache2 -k start
                    └─3555 /usr/sbin/apache2 -k start
                      └─3556 /usr/sbin/apache2 -k start

oct 16 14:13:16 localhost-VirtualBox systemd[1]: Starting apache2.service - The Apache HT
oct 16 14:13:16 localhost-VirtualBox apachectl[3549]: AH00558: apache2: Could not reliabl
oct 16 14:13:16 localhost-VirtualBox systemd[1]: Started apache2.service - The Apache HT
Mostrar aplicaciones (ID)
```

Y con este comando sí que indicó que esto se está ejecutando en el puerto 8080.

```
localhost@localhost-VirtualBox:~$ sudo apt install net-tools -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 15 no actualizados.
Se necesita descargar 204 kB de archivos.
Se utilizarán 811 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 net-tools amd64 2.10-0
Descargados 204 kB en 0s (1.067 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 151261 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../net-tools_2.10-0.1ubuntu4.4_amd64.deb ...
Desempaquetando net-tools (2.10-0.1ubuntu4.4) ...
Configurando net-tools (2.10-0.1ubuntu4.4) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
localhost@localhost-VirtualBox:~$ sudo netstat -tulpn | grep 8080
tcp6      0      0 :::8080          :::*              ESCUCHAR      3550/apac
```


Crea un archivo para poder indicar que puedo mostrar información del php que he instalado.

```
localhost@localhost-VirtualBox:~$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
```

Se utiliza para solicitar y mostrar el contenido de una página web específica. En este caso, intenta acceder al archivo info.php en el servidor web local (localhost) que está escuchando en el puerto 8080. La salida muestra la advertencia sobre usar la versión Snap de curl y luego comienza a imprimir el código HTML de la página solicitada, lo que indica que la conexión al servidor fue efectiva.

```
localhost@localhost-VirtualBox:~$ curl http://localhost:8080/info.php

Caution: You are using the Snap version of curl.
Due to Snap's sandbox nature, this version has some limitations.
For example, it may not be able to access hidden folders in your home directory
or other restricted areas of the os.
This means you may encounter errors when using snap curl to download files.
For those case, you might want to use the native curl package.
For details, see: https://github.com/boukendesho/curl-snap/issues/1

To stop seeing this message, run the following command:
curl.snap-acked

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
<head>
```

2. INSTALACIÓN Y CONFIGURACIÓN DE NGINX

El comando `$\mathbf{\text{sudo apt install nginx -y}}` se utiliza para instalar el servidor web Nginx en el sistema. `$\mathbf{\text{apt}}` es el gestor de paquetes en sistemas basados en Debian/Ubuntu, y `sudo` se usa para tener permisos de administrador. El flag `$\mathbf{\text{-y}}` acepta automáticamente todas las solicitudes de instalación, y la salida muestra que el sistema ha comenzado el proceso de lectura de paquetes y verificación de dependencias antes de proceder con la descarga e instalación.

```
localhost@localhost-VirtualBox:~$ sudo apt install nginx -y
[sudo] contraseña para localhost:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  liblvm2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  nginx-common
Paquetes sugeridos:
  fcgiwrap nginx-doc
Se instalarán los siguientes paquetes NUEVOS:
  nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 11 no actualizados.
Se necesita descargar 564 kB de archivos.
Se utilizarán 1.596 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.
Des:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ub
Descargados 564 kB en 0s (1.991 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete nginx-common previamente no seleccionado.
(Leyendo la base de datos ... 151316 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Desempaquetando nginx-common (1.24.0-2ubuntu7.5) ...
```

Aquí cambio el puerto de listen 80 al 8081, ya que sirve para especificar el puerto y la dirección IP en los que el servidor web Nginx debe escuchar las conexiones entrantes y esto se cambia porque evita conflictos entre puertos, requiere permisos de administrador y para dejar el puerto 80 para el proxy principal.

```
GNU nano 7.2 /etc/nginx/sites-available/default *
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8081 default_server;
    listen [::]:8081 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
```

Con este comando lo que conseguimos es crear una página HTML identificable y determinada para el puerto Nginx.

```
localhost@localhost-VirtualBox:~$ echo "<h1>Servidor Nginx</h1>Funcionando en puerto 8081</p>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1>Funcionando en puerto 8081</p>
```

Reinicio Nginx para poder aplicar los cambios de configuración.

```
localhost@localhost-VirtualBox:~$ sudo systemctl restart nginx
```

Ahora compruebo que el Nginx se está ejecutando y funcionando correctamente en el puerto 8081.

```
localhost@localhost-VirtualBox:~$ sudo systemctl status nginx
[sudo] contraseña para localhost:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-10-16 14:51:26 CEST; 4 days ago
     Docs: man:nginx(8)
   Process: 6003 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 6005 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 6036 (nginx)
    Tasks: 5 (limit: 4601)
   Memory: 3.7M (peak: 8.2M)
      CPU: 30ms
   CGroup: /system.slice/nginx.service
           └─6036 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─6039 "nginx: worker process"
                 └─6040 "nginx: worker process"
                    └─6041 "nginx: worker process"
                       └─6042 "nginx: worker process"

oct 16 14:51:26 localhost-VirtualBox systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server:
oct 16 14:51:26 localhost-VirtualBox systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server:
```

Verifico el estado del servidor Nginx, mostrando qué puerto de red está utilizando actualmente para escuchar las conexiones entrantes (puerto 8081). Y en la captura de abajo muestro el contenido que me envía mi servidor de web local que está escuchando en el puerto 8081.

```
mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep nginx
tcp        0      0 0.0.0.0:80      0.0.0.0:*        LISTEN      *
tcp6       0      0 :::8081        :::*              LISTEN      *

mario@mario-VirtualBox:~$ curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2022-03-22
  See: https://launchpad.net/bugs/1966004
-->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Apache2 Ubuntu Default Page: It works</title>
  <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }
</style>
<div style="text-align: center;>
  <img alt="Ubuntu logo" data-bbox="484 444 515 464" style="height: 40px; width: 40px; margin: 0 auto 0 auto;"/>
  <p>Ubuntu</p>
  <p>It works!</p>
</div>
```

3. INSTALACIÓN Y CONFIGURACIÓN DE CADDY

Verifico la autenticidad de los paquetes de Debian, habilita la descarga segura de paquetes a través de HTTPS, e instala la herramienta curl para transferir datos por URL.

```
mario@mario-VirtualBox:~$ sudo apt install -y debian-keyring
g debian-archive-keyring apt-transport-https curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
curl ya está en su versión más reciente (8.5.0-2ubuntu10.6)
.
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https debian-archive-keyring
  debian-keyring
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0
no actualizados.
Se necesita descargar 31,5 MB de archivos.
Se utilizarán 33,4 MB de espacio de disco adicional después
de esta operación.
```

Este comando intentaba descargar una clave pública GPG (<https://dl.cloudsmith.io/>...) usando curl y luego importarla de forma segura al sistema (sudo gpg --dearmor -o ...) para poder verificar la autenticidad de los paquetes del repositorio de Caddy.

```
mario@mario-VirtualBox:~$ curl -IsLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o
/usr/share/keyrings/caddy-stable-archive-keyring.gpg
gpg: missing argument for option "-o"
Orden «url» no encontrada. Quizá quiso decir:
  la orden «erl» del paquete snap «erlang (25.3)»
  la orden «curl» del paquete snap «curl (8.16.0)»
  la orden «urh» del paquete snap «urh (2.9.3)»
  la orden «surl» del paquete snap «surl (0.8.0)»
  la orden «yurl» del paquete snap «yurl (v0.6.3)»
  la orden «ul» del paquete deb «bsdextrautils (2.39.3-9ubuntu6.3)»
  la orden «erl» del paquete deb «erlang-base (1:25.3.2.8+dfsg-1ubuntu4.5)»
  la orden «uil» del paquete deb «uil (2.3.8-3)»
  la orden «curl» del paquete deb «curl (8.5.0-2ubuntu10.6)»
»
  la orden «ur» del paquete deb «libur-perl (0.470+ds-2)»
  la orden «zurl» del paquete deb «zurl (1.12.0-1)»
Consulte «snap info <nombre del snap>» para ver más versiones.
```

Este comando utiliza curl para descargar la definición del repositorio del servidor web Caddy y la canaliza a tee para añadir esa dirección como una nueva fuente de paquetes APT (/etc/apt/sources.list.d/).

```
mario@mario-VirtualBox:~$ curl -sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
# Source: Caddy
# Site: https://github.com/caddyserver/caddy
# Repository: Caddy / stable
# Description: Fast, multi-platform web server with automatic HTTPS

deb [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main

deb-src [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main
bash: /etc/apt/sources.list.d/caddy-stable.list: No existe el archivo o el directorio
```

Actualizo la lista de paquetes disponibles (apt update) y, si tiene éxito, instala el servidor web Caddy (apt install caddy -y) con privilegios de administrador.

```
mario@mario-VirtualBox:~$ sudo apt update && sudo apt install caddy -y
Obj:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libllvm19
```


Creo un directorio (mkdir) para alojar los archivos web de Caddy, asegurándose de crear todos los directorios padres necesarios (-p) si no existen.

```
mario@mario-VirtualBox:~$ sudo mkdir -p /var/www/caddy
```

Esta serie de comandos tiene como objetivo crear un archivo de texto llamado README.md en el directorio web de Caddy (/var/www/caddy/). Cada línea usa echo para imprimir una cadena de texto y la redirige a tee -a para añadir (-a) el contenido a ese archivo, creando así la página de bienvenida con formato Markdown. El mensaje de "Permiso denegado" indica que en ese punto el usuario no tenía los derechos necesarios para escribir en el archivo, a pesar de usar sudo.

```
mario@mario-VirtualBox:~$ echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
# Bienvenido a Caddy
```

```
mario@mario-VirtualBox:~$ echo "" | sudo tee -a /var/www/caddy/README.md
```

```
mario@mario-VirtualBox:~$ echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
bash: /var/www/caddy/README.md: Permiso denegado
```

```
mario@mario-VirtualBox:~$ echo "" | sudo tee -a /var/www/caddy/README.md
```

```
mario@mario-VirtualBox:~$
```

```
mario@mario-VirtualBox:~$ echo "## Características" | sudo tee -a /var/www/caddy/README.md
## Características
```

```
mario@mario-VirtualBox:~$ echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
- Servidor moderno
```

```
mario@mario-VirtualBox:~$ echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md
- HTTPS automático
```

```
mario@mario-VirtualBox:~$ echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
- Fácil configuración
```

Utilizo curl para descargar un archivo de imagen desde una URL específica y lo guarda (-o) temporalmente en la ruta /tmp/test-image.jpg.

```
mario@mario-VirtualBox:~$ curl -o /tmp/test-image.jpg "http
s://www.python.org/static/apple-touch-icon-144x144-precomp
sed.png"
  % Total    % Received % Xferd  Average Speed   Time    Ti
me      Time     Current                Dload  Upload  Total  Sp
ent      Left   Speed
  0     0     0      0     0      0     0      0  --:--:--  --:--
100  7382  100  7382    0     0  91172      0  --:--:--  --:--
--:--  --:--:--  92275
```

Muevo (mv) el archivo de imagen recién descargado desde la carpeta temporal a la carpeta de contenido web de Caddy, renombrándolo a test.jpg y usando privilegios de administrador.

```
mario@mario-VirtualBox:~$ sudo mv /tmp/test-image.jpg /var/
www/caddy/test.jpg
```

Abro y edito el Caddyfile para declarar mis sitios (dominio o localhost), rutas y reglas de proxy/reverse-proxy; así defino qué contenido sirve Caddy y cómo gestionar el TLS automático.

```
GNU nano 7.2 /etc/caddy/Caddyfile *
# The Caddyfile is an easy way to configure your Caddy web>
#
# Unless the file starts with a global options block, the >
# uncommented line is always the address of your site.
#
# To use your own domain name (with automatic HTTPS), firs>
# sure your domain's A/AAAA DNS records are properly point>
# this machine's public IP, then replace ":80" below with >
# domain name.

:8082 {
    # Set this path to your site's directory.
    root * /var/www/caddy

    # Enable the static file server.
    file_server browse

    @markdown path*.md
    header @markdown Content-Type text/plain
}

# Refer to the Caddy docs for more information:
# https://caddyserver.com/docs/caddyfile
```


Habilito y arranco el servicio de Caddy para que se inicie al arrancar y se ponga en funcionamiento de inmediato con la configuración que he creado.

```
mario@mario-VirtualBox:~$ sudo systemctl restart caddy
```

Compruebo el estado del servicio para asegurarme de que Caddy está activo, sin errores de arranque y con los sockets/puertos esperados abiertos.

```
mario@mario-VirtualBox:~$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; vendor preset: enabled)
   Active: active (running) since Mon 2025-10-27 13:50:28 CEST; 1min 1s ago
     Docs: https://caddyserver.com/docs/
  Main PID: 28727 (caddy)
    Tasks: 9 (limit: 4603)
   Memory: 6.8M (peak: 7.1M)
      CPU: 28ms
   CGroup: /system.slice/caddy.service
           └─28727 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

oct 27 13:50:28 mario-VirtualBox caddy[28727]: MEMORY_PRESSURE: 100%
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Starting Caddy","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Using config: /etc/caddy/Caddyfile","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Advert: Caddy is running as root. It is recommended to run Caddy as an unprivileged user, e.g. 'caddy:admin' or 'caddy:www-data'.","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"info","msg":"Autoconfiguring certificates for https://caddyserver.com:443","module":"caddy","version":"v2.10.0"}
oct 27 13:50:28 mario-VirtualBox systemd[1]: Started caddy.
lines 1-21/21 (END)
```

Lo utilizo para verificar qué puertos de red están abiertos y en uso (escuchando) por el servidor web Caddy.

La salida confirma que el proceso Caddy (28727/caddy) está escuchando en el puerto 8082 para conexiones IPv6 (y posiblemente un puerto de API interna en 2019), lo que indica que el servidor se está ejecutando correctamente.

```
mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep caddy
tcp        0      0 127.0.0.1:2019      0.0.0.0:*
            ESCUCHAR      28727/caddy
tcp6       0      0 :::8082             :::*
            ESCUCHAR      28727/caddy
mario@mario-VirtualBox:~$
```

Esto lo hago funcionar como una prueba rápida para confirmar que el servidor web es accesible y está respondiendo correctamente en el puerto 8082, mostrando el contenido que está sirviendo.

```
mario@mario-VirtualBox:~$ curl http://localhost:8082/  
<!DOCTYPE html>  
<html>  
  
<head>  
    <title>/</title>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <style>  
        * {  
            padding: 0;  
            margin: 0;  
        }  
  
        body {  
            font-family: sans-serif;  
            text-rendering: optimizespeed;  
            background-color: #ffffff;  
        }
```

Esto confirma que el archivo de bienvenida creado previamente es accesible y que el servidor web Caddy está sirviendo archivos estáticos correctamente en el puerto 8082.

```
mario@mario-VirtualBox:~$ curl http://localhost:8082/README
.md
# Bienvenido a Caddy

## Características
- Servidor moderno
- HTTPS automático
- Fácil configuración
mario@mario-VirtualBox:~$
```

4. CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

Instalo Certbot y su plugin para Apache para poder solicitar certificados TLS/SSL y que Certbot automatice la configuración en Apache.

```
mario@mario-VirtualBox:~$ sudo apt install certbot python3-
certbot-apache -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma auto
mática y ya no es necesario.
    libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
    augeas-lenses libaugeas0 python3-acme python3-augeas
python3-certbot python3-configargparse python3-icu
python3-josepy python3-openssl python3-parsedatetime
python3-rfc3339
Paquetes sugeridos:
    augeas-doc python-certbot-doc python3-certbot-nginx
augeas-tools python-acme-doc python-certbot-apache-doc
python-openssl-doc python3-openssl-dbg
```

Genero un nuevo certificado SSL/TLS autofirmado junto con su clave privada, ambos necesarios para habilitar HTTPS en un servidor web.

[illegible]

Este es un paso fundamental para que Apache pueda entender y gestionar conexiones HTTPS. Como indica el resultado, es necesario reiniciar el servicio de Apache (systemctl restart apache2) para que la nueva configuración surta efecto.

```
mario@mario-VirtualBox:~$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
mario@mario-VirtualBox:~$
```

Este comando openssl se utiliza para generar un nuevo certificado SSL/TLS autofirmado junto con su clave privada. El certificado se crea con una validez de 365 días (-days 365), la clave privada RSA de 2048 bits se genera sin protección por contraseña (-nodes), y los archivos resultantes se guardan en directorios estándar del sistema (/etc/ssl/) para ser utilizados por un servidor web y habilitar HTTPS.

```
<VirtualHost *:8443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available
    # enabled or disabled at a global level, it is possible
    # to include a line for only one particular virtual host
    # following line enables the CGI configuration for this host
    # after it has been globally disabled with "a2discouragecgi"
    #Include conf-available/serve-cgi-bin.conf

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on
```

Lo utilizo para activar la configuración del sitio "default-ssl" en el servidor web Apache. Esto le indica a Apache que cargue y utilice este archivo de configuración. El sistema avisa que es necesario recargar Apache (systemctl reload apache2) para aplicar los cambios.

```
mario@mario-VirtualBox:~$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Reinicio el servidor web Apache para aplicar los cambios de configuración SSL realizados anteriormente.

```
mario@mario-VirtualBox:~$ sudo systemctl restart apache2
```

Realizo una prueba de conexión al servidor local a través de HTTPS en el puerto 8443, mostrando las cabeceras (-i) e ignorando los errores del certificado autofirmado (-k), para confirmar que el sitio SSL está funcionando.

```
mario@mario-VirtualBox:~$ curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Mon, 27 Oct 2025 12:58:20 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Mon, 27 Oct 2025 11:45:05 GMT
ETag: "29af-642226e2d20fb"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//
EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.d
t">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; cha
rset=UTF-8" />
```

5. VERIFICACIÓN FINAL DE LOS 3 SERVIDORES

Confirmo que el servicio está "active (running)" (activo y en ejecución).

```
mario@mario-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.servi>
   Active: active (running) since Mon 2025-10-27 13:58:0>
   Docs: https://httpd.apache.org/docs/2.4/
  Process: 29368 ExecStart=/usr/sbin/apachectl start (co>
 Main PID: 29372 (apache2)
    Tasks: 6 (limit: 4603)
  Memory: 12.2M (peak: 12.5M)
     CPU: 62ms
   CGroup: /system.slice/apache2.service
           └─29372 /usr/sbin/apache2 -k start
             └─29374 /usr/sbin/apache2 -k start
               └─29375 /usr/sbin/apache2 -k start
                 └─29376 /usr/sbin/apache2 -k start
                   └─29377 /usr/sbin/apache2 -k start
                     └─29378 /usr/sbin/apache2 -k start

oct 27 13:58:03 mario-VirtualBox systemd[1]: Starting apac>
oct 27 13:58:03 mario-VirtualBox apachectl[29371]: AH00558>
oct 27 13:58:03 mario-VirtualBox systemd[1]: Started apach>
lines 1-20/20 (END)
```

Filtro para mostrar que Apache está escuchando conexiones en los puertos 443 (HTTPS estándar) y 8443.

```
mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep apache
2
tcp6        0      0 :::443          :::*
              ESCUCHAR      29372/apache2
tcp6        0      0 :::8443         :::*
              ESCUCHAR      29372/apache2
mario@mario-VirtualBox:~$
```


Es lo mismo de arriba pero en nginx.

```
mario@mario-VirtualBox:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; vendor preset: en>
   Active: active (running) since Mon 2025-10-27 13:14:51; 1min 1s ago>
     Docs: man:nginx(8)
    Main PID: 25864 (nginx)
      Tasks: 5 (limit: 4603)
     Memory: 3.8M (peak: 4.6M)
        CPU: 19ms
    CGroup: /system.slice/nginx.service
            └─25864 "nginx: master process /usr/sbin/nginx"
               └─25865 "nginx: worker process"
                  └─25866 "nginx: worker process"
                     └─25867 "nginx: worker process"
                        └─25868 "nginx: worker process"

oct 27 13:14:51 mario-VirtualBox systemd[1]: Starting nginx: high performance web server and a reverse proxy server.
oct 27 13:14:51 mario-VirtualBox systemd[1]: Started nginx: high performance web server and a reverse proxy server.
lines 1-17/17 (END)
```

```
mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep nginx
tcp        0      0 0.0.0.0:80          0.0.0.0:*
           ESCUCHAR      25864/nginx: master
tcp6       0      0 :::8081            :::*
           ESCUCHAR      25864/nginx: master
mario@mario-VirtualBox:~$
```


Y otra vez lo mismo pero con Caddy.

```
mario@mario-VirtualBox:~$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service>
   Active: active (running) since Mon 2025-10-27 13:50:2>
     Docs: https://caddyserver.com/docs/
  Main PID: 28727 (caddy)
    Tasks: 9 (limit: 4603)
   Memory: 7.3M (peak: 7.7M)
      CPU: 69ms
   CGroup: /system.slice/caddy.service
           └─28727 /usr/bin/caddy run --environ --config>

oct 27 13:50:28 mario-VirtualBox caddy[28727]: MEMORY_PRES>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox caddy[28727]: {"level":"i>
oct 27 13:50:28 mario-VirtualBox systemd[1]: Started caddy>
lines 1-21/21 (END)

mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep caddy
tcp        0      0 127.0.0.1:2019        0.0.0.0:*
           ESCUCHAR    28727/caddy
tcp6       0      0 :::8082               :::*
           ESCUCHAR    28727/caddy
mario@mario-VirtualBox:~$
```

Lo utilizo para revisar todos los puertos que están "ESCUCHANDO" en el sistema (-tulpn) y luego filtra (grep -E) la lista para mostrar únicamente las líneas que coincidan con los puertos 80, 8081, 8082 o 8443. La salida es un resumen que muestra qué servicio (Apache, Caddy, Nginx) está operando en cada uno de esos puertos específicos.

```
mario@mario-VirtualBox:~$ sudo netstat -tulpn | grep -E '80
80|8081|8082|8443'
tcp6        0      0 :::8443          :::*
            ESCUCHAR    29372/apache2
tcp6        0      0 :::8082          :::*
            ESCUCHAR    28727/caddy
tcp6        0      0 :::8081          :::*
            ESCUCHAR    25864/nginx: master
mario@mario-VirtualBox:~$
```

He probado y verifico que cada servidor responde correctamente en su puerto asignado y te pongo la comprobación de uno por no ponerte la misma captura de los otros tres puertos ya que si no la practica te la voy a alargar innecesariamente.

```
mario@mario-VirtualBox:~$ curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//
EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.d
t">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; cha
rset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }
```