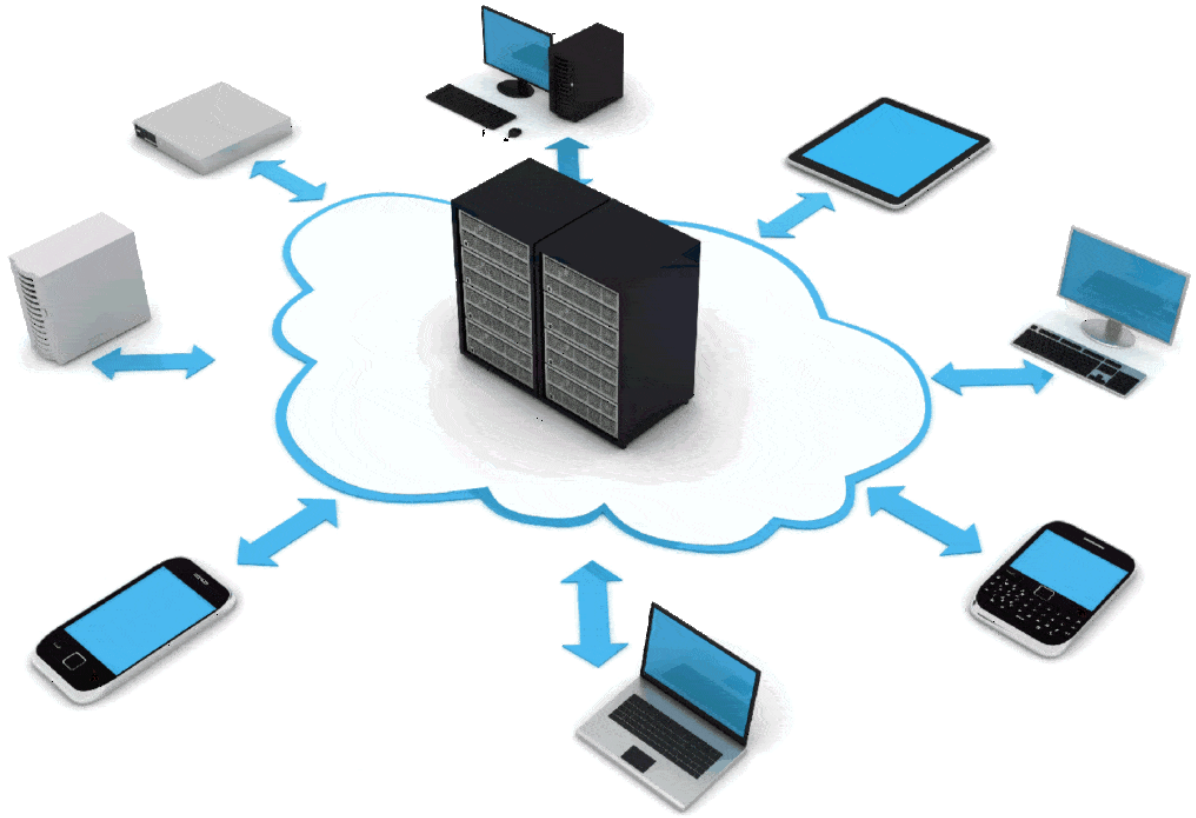
TEMARIO: ARQUITECTURA EN LA NUBE - CONFIGURACIÓN AVANZADA DE SERVIDORES WEB Y HTTPS

Evelyn Arcentales



El trabajo se procede a realizar desde una máquina virtual de ubuntu server.

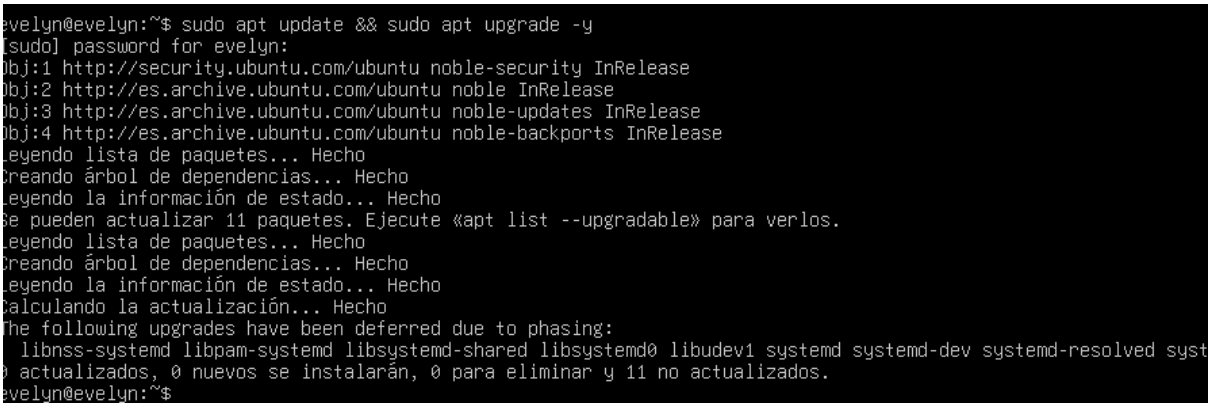
1. COMANDOS Y PROCEDIMIENTOS PARTE

1: INSTALACIÓN Y CONFIGURACIÓN DE APACHE

1. Actualizar el sistema Comando:

sudo apt update && sudo apt upgrade -y

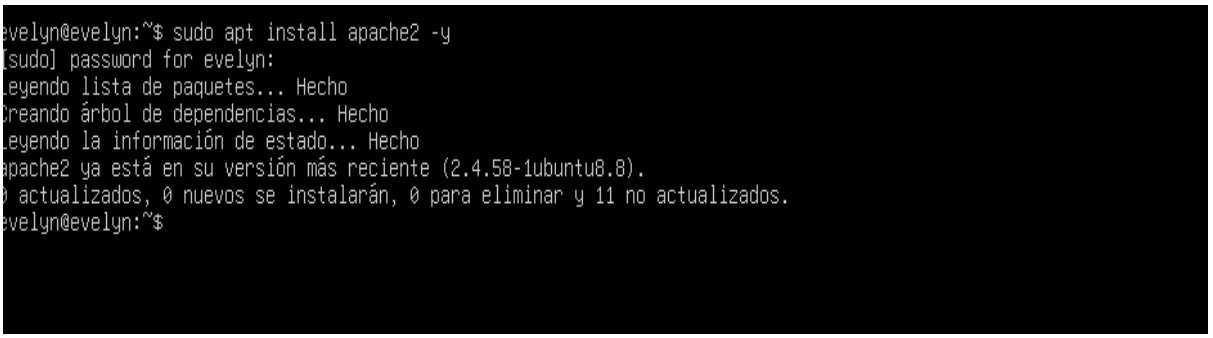
Descripción: Actualiza la lista de paquetes y mejora el sistema a las últimas versiones.



2. Instalar Apache2 Comando:

sudo apt install apache2 -y

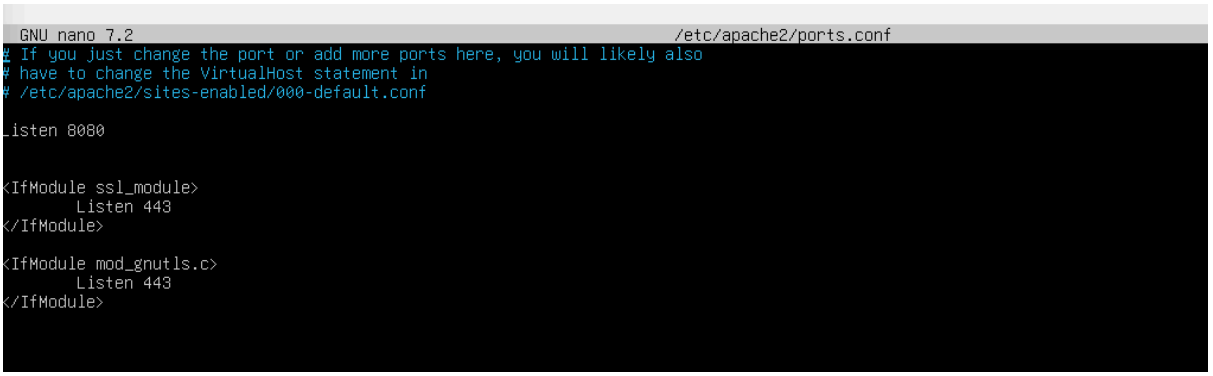
Descripción: Instala el servidor web Apache en tu sistema.



3. Configurar Apache en puerto 8080 Comando:

sudo nano /etc/apache2/ports.conf

Descripción: Abre el archivo de configuración de puertos. Cambia Listen 80 por Listen 8080.



4. Modificar el VirtualHost Comando:

sudo nano /etc/apache2/sites-available/000-default.conf

Descripción: Cambia por

```
 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
</VirtualHost>
```

5. Instalar PHP Comando:

sudo apt install php libapache2-mod-php -y s

Descripción: Instala PHP y su módulo para funcionar con Apache.

```
evelyn@evelyn:~$ sudo apt install php libapache2-mod-php -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
php ya está en su versión más reciente (2:8.3+93ubuntu2).
libapache2-mod-php ya está en su versión más reciente (2:8.3+93ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
evelyn@evelyn:~$ _
```

6. Reiniciar Apache Comando:

sudo systemctl restart apache2

Descripción: Reinicia Apache para aplicar los cambios.

```
evelyn@evelyn:~$ sudo systemctl restart apache2
evelyn@evelyn:~$
```

7. Verificar estado de Apache Comando:

sudo systemctl status apache2

Descripción: Comprueba que Apache está funcionando correctamente en el puerto 8080

```
evelyn@evelyn:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-10-14 07:04:51 UTC; 2min 33s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1267 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1272 (apache2)
      Tasks: 6 (limit: 7080)
     Memory: 10.5M (peak: 11.7M)
        CPU: 47ms
     CGroup: /system.slice/apache2.service
             ├─1272 /usr/sbin/apache2 -k start
             ├─1274 /usr/sbin/apache2 -k start
             ├─1275 /usr/sbin/apache2 -k start
             ├─1276 /usr/sbin/apache2 -k start
             ├─1277 /usr/sbin/apache2 -k start
             └─1278 /usr/sbin/apache2 -k start

oct 14 07:04:51 evelyn systemd[1]: Starting apache2.service - The Apache HTTP Server...
oct 14 07:04:51 evelyn apachectl[1271]: AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
oct 14 07:04:51 evelyn systemd[1]: Started apache2.service - The Apache HTTP Server.

evelyn@evelyn:~$
evelyn@evelyn:~$
```

comando: ss -tuln | grep 8080

```
evelyn@evelyn:~$
evelyn@evelyn:~$ ss -tuln |grep 8080
tcp   LISTEN 0      511                *:8080             *:*
evelyn@evelyn:~$
```

para que este en apache en el puerto 8080

```
evelyn@evelyn:~$ sudo apt install net-tools
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 2 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 2.10-0.1ubuntu4.4 [204 kB]
Descargados 204 kB en 0s (683 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 84996 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) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
evelyn@evelyn:~$ netstat -tulpn | grep 8080
(No info could be read for "-p": geteuid()=1000 but you should be root.)
tcp6       0      0 :::8080                 :::*                    LISTEN      -
evelyn@evelyn:~$ sudo systemctl restart networking
Failed to restart networking.service: Unit networking.service not found.
evelyn@evelyn:~$ sudo netstat -tulpn | grep 8080
tcp6       0      0 :::8080                 :::*                    LISTEN      1272/apache2
evelyn@evelyn:~$
```

8. Crear archivo PHP de prueba Comando:

echo "" | sudo tee /var/www/html/info.php

Descripción: Crea un archivo que muestra información del PHP instalado.

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

9. Probar Apache desde terminal Comando:

curl http://localhost:8080/info.php

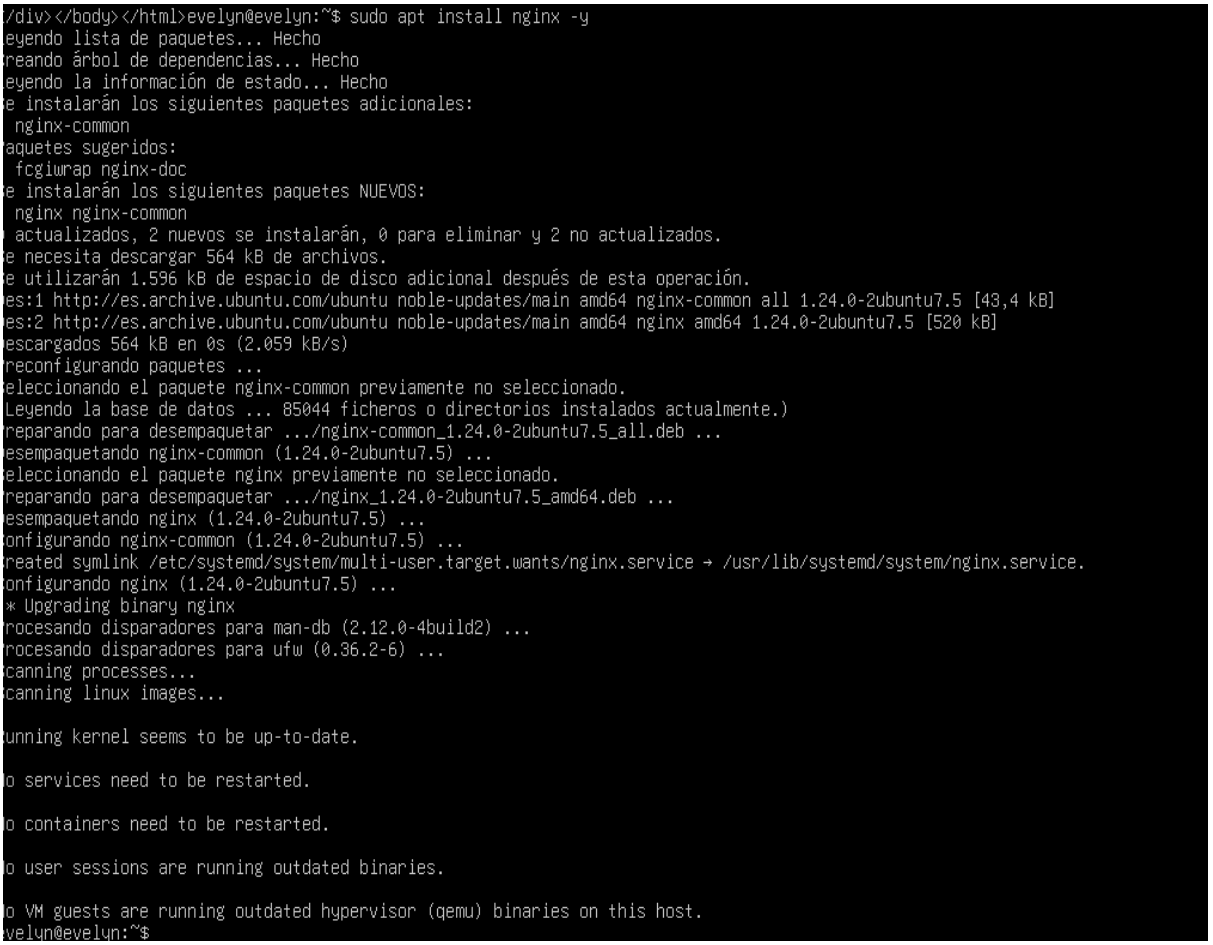
Descripción: Verifica que Apache sirve correctamente el contenido PHP.

con el comando indicado ha devuelto un html, in dicando lo correcto que esta

```
r><td class="e">tidy </td><td class="v">John Coggeshall, Ilia Alshanetsky </td></tr>
r><td class="e">tokenizer </td><td class="v">Andrei Zmievski, Johannes Schlueter </td></tr>
r><td class="e">XML </td><td class="v">Stig Bakken, Thies C. Arntzen, Sterling Hughes </td></tr>
r><td class="e">XMLReader </td><td class="v">Rob Richards </td></tr>
r><td class="e">XMLWriter </td><td class="v">Rob Richards, Pierre-Alain Joye </td></tr>
r><td class="e">XSL </td><td class="v">Christian Stocker, Rob Richards </td></tr>
r><td class="e">Zip </td><td class="v">Pierre-Alain Joye, Remi Collet </td></tr>
r><td class="e">Zlib </td><td class="v">Rasmus Lerdorf, Stefan Roehrich, Zeev Suraski, Jade Nicoletti, Michael Wallner </td><
table>
able>
r class="h"><th colspan="2">PHP Documentation</th></tr>
r><td class="e">Authors </td><td class="v">Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Philip
 Jakub Vrana, Adam Harvey </td></tr>
r><td class="e">Editor </td><td class="v">Peter Cowburn </td></tr>
r><td class="e">User Note Maintainers </td><td class="v">Daniel P. Brown, Thiago Henrique Pojda </td></tr>
r><td class="e">Other Contributors </td><td class="v">Previously active authors, editors and other contributors are listed in
table>
able>
r class="h"><th>PHP Quality Assurance Team</th></tr>
r><td class="e">Ilia Alshanetsky, Joerg Behrens, Antony Dovgal, Stefan Esser, Moriyoshi Koizumi, Magnus Maatta, Sebastian Nohr
Pierre-Alain Joye, Dmitry Stogov, Felipe Pena, David Soria Parra, Stanislav Malyshev, Julien Pauli, Stephen Zarkos, Anatol Be
td></tr>
table>
able>
r class="h"><th colspan="2">Websites and Infrastructure team</th></tr>
r><td class="e">PHP Websites Team </td><td class="v">Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas Kahwe Smith, Pierre
 Peter Cowburn, Adam Harvey, Ferenc Kovacs, Levi Morrison </td></tr>
r><td class="e">Event Maintainers </td><td class="v">Damien Seguy, Daniel P. Brown </td></tr>
r><td class="e">Network Infrastructure </td><td class="v">Daniel P. Brown </td></tr>
r><td class="e">Windows Infrastructure </td><td class="v">Alex Schoenmaker </td></tr>
table>
able>
r class="h"><th colspan="1">Debian Packaging</th></tr>
r><td class="e">Ondřej Surý </td></tr>
table>
2>PHP License</h2>
able>
r class="v"><td>
>
is program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the
ibution in the file:  LICENSE
p>
>This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty
 PARTICULAR PURPOSE.
p>
>If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.
p>
td></tr>
table>
div></body></html>evelyn@evelyn:~$
```

PARTE 2: INSTALACIÓN Y CONFIGURACIÓN DE NGINX

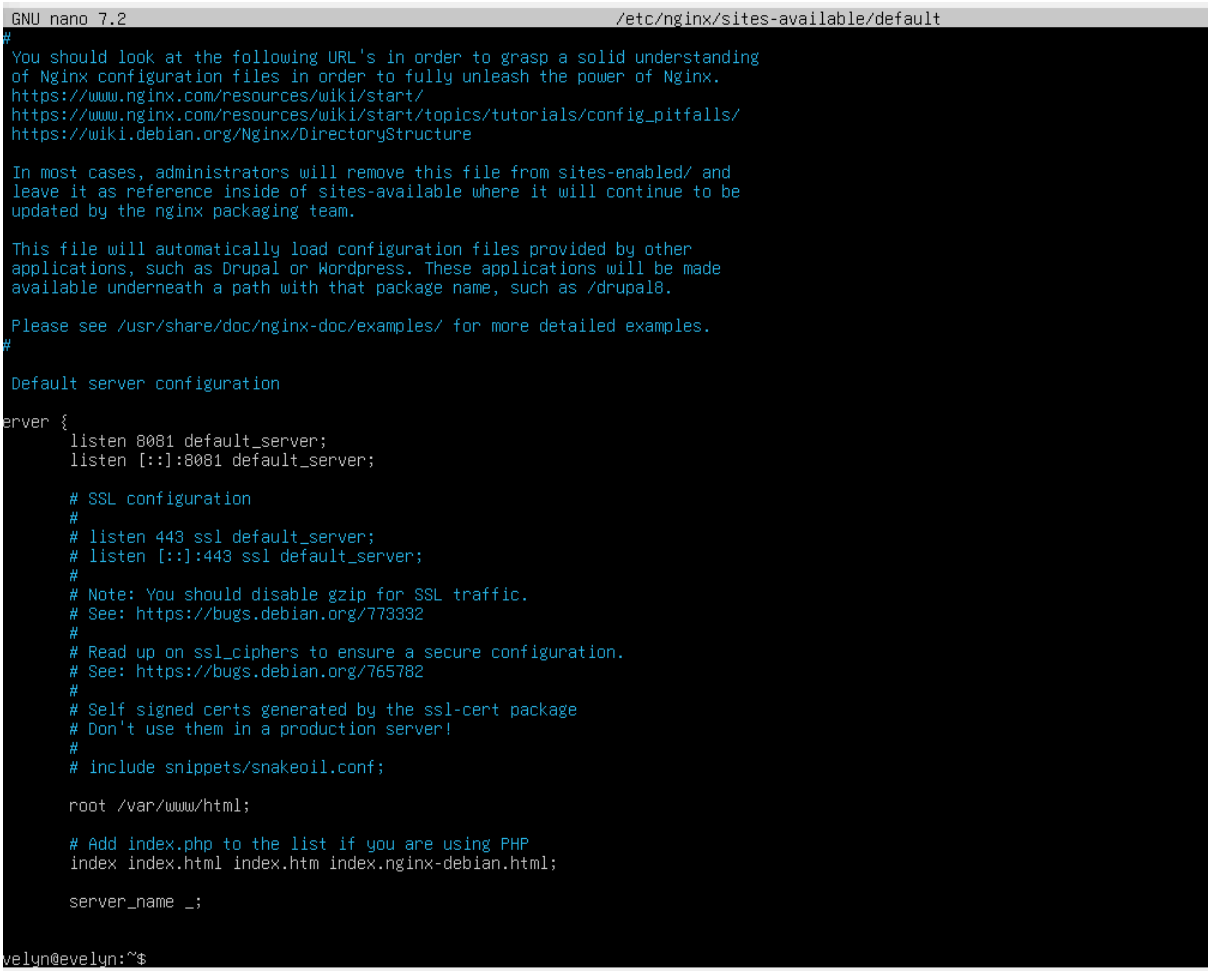
1. Instalar Nginx Comando:
sudo apt install nginx -y
Descripción: Instala el servidor web Nginx en tu sistema aqui se instala como segundo servidor web, Es conocido por su eficiencia y capacidad para manejar multiplex conexiones simultaneas.



2. Configurar Nginx en puerto 8081

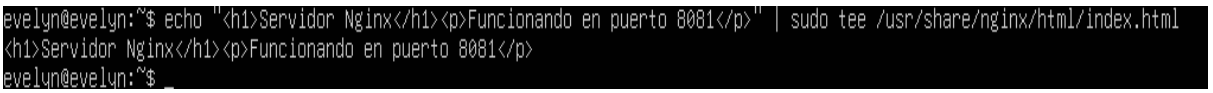
Comando: sudo nano /etc/nginx/sites-available/default

Descripción: Abre la configuración por defecto. Cambia listen 80 por listen 8081. Cada servidor web necesita su propio puerto para evitar conflictos.



3. Crear página HTML personalizada

Descripción: Crea una página HTML identificable para Nginx. Se crea contenido HTML específico que identifica inequívocamente que estamos accediendo al servidor Nginx.



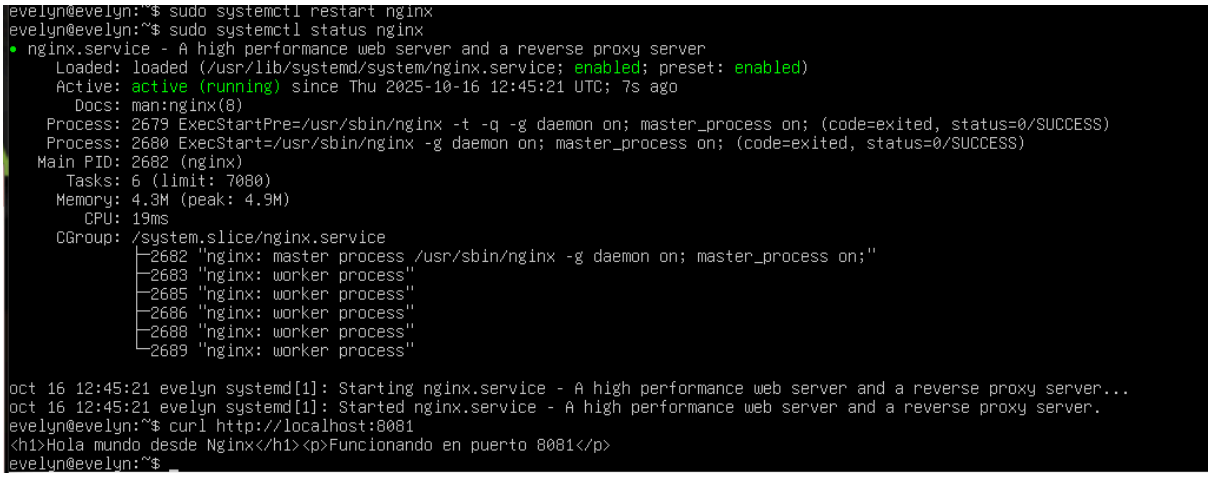
Nota: he tenido que levantar nginx con para funcionar

- sudo systemctl start nginx

- sudo systemctl status nginx
- echo de hola mundo
- sudo systemctl restart nginx
- sudo systemctl status nginx
- curl http://localhost:8081

4. Reiniciar Nginx

Comando: sudo systemctl restart nginx
Descripción: Reinicia Nginx para aplicar los cambios de configuración.



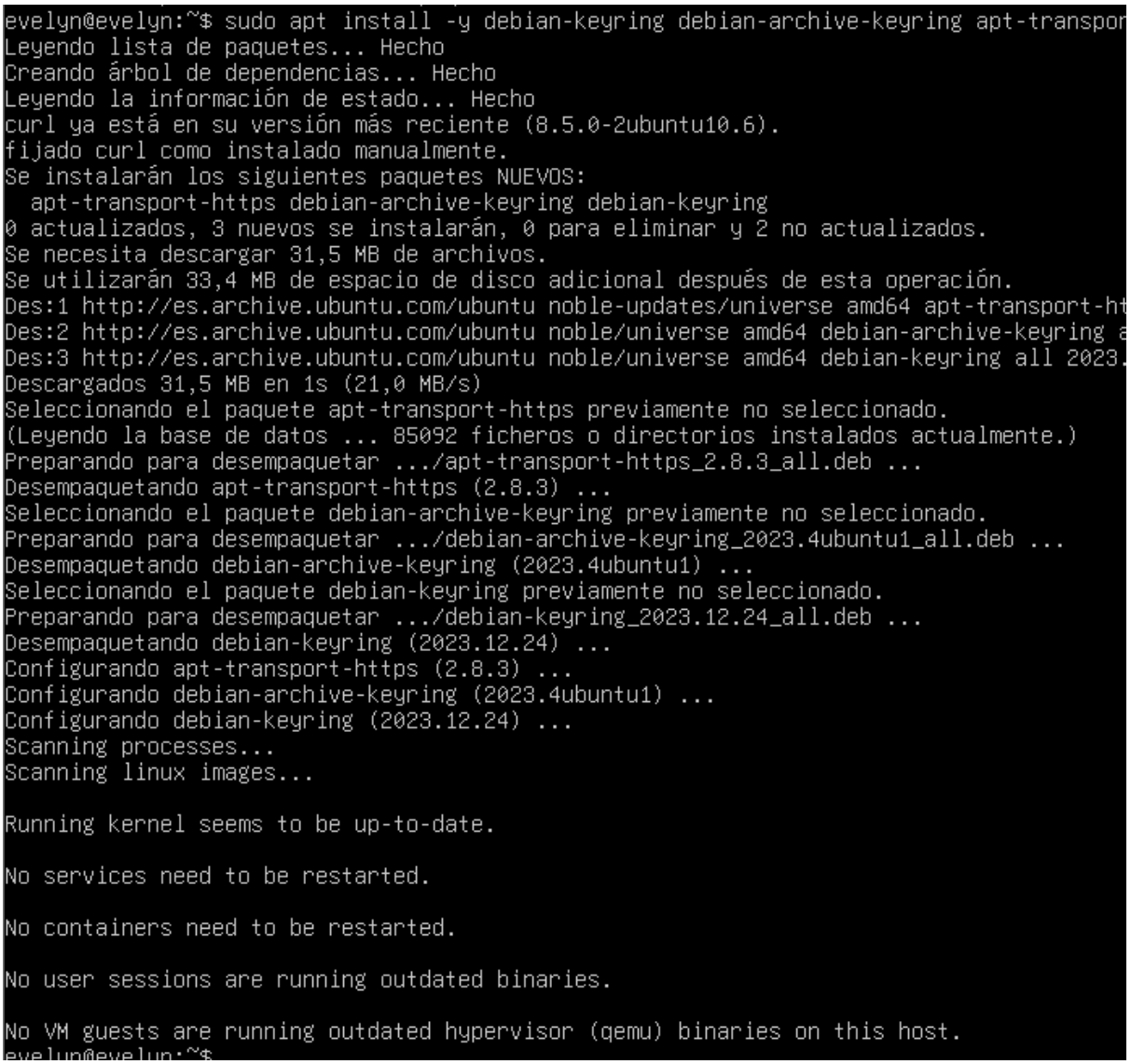
Verificación con curl

PARTE 3: INSTALACIÓN Y CONFIGURACIÓN DE CADDY

1. Instalar dependencias necesarias

Comando: sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl

Descripción: Instala herramientas necesarias para añadir repositorios externos



2. Agregar repositorio de Caddy

Comando: curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg - dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg

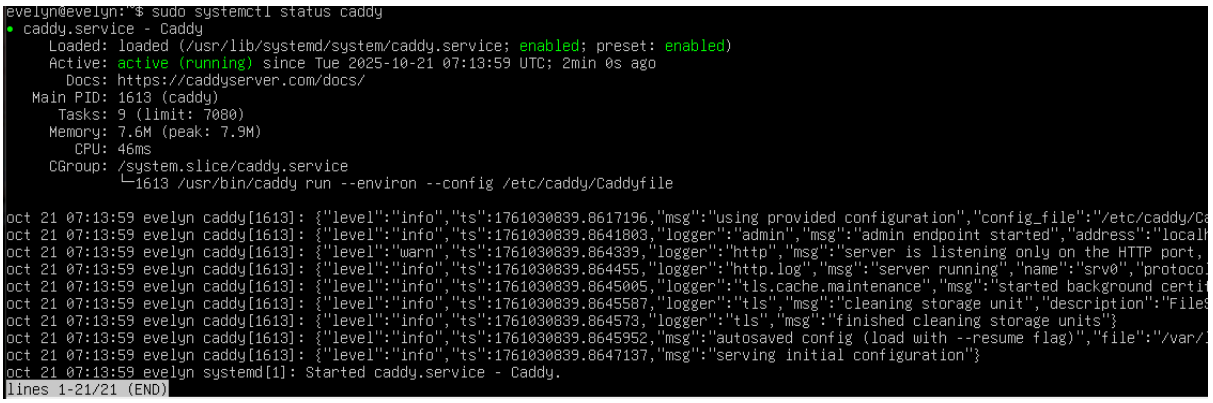
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list Descripción: Añade el repositorio oficial de Caddy a tu sistema

```
es:34 http://es.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5.564 B]
es:35 http://es.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
es:36 http://es.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [484 B]
es:37 http://es.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7.140 B]
es:38 http://es.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
es:39 http://es.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11,0 kB]
es:40 http://es.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
escargados 12,4 MB en 2s (5.959 kB/s)
eyendo lista de paquetes... Hecho
reando árbol de dependencias... Hecho
eyendo la información de estado... Hecho
e pueden actualizar 9 paquetes. Ejecute «apt list --upgradable» para verlos.
velyn@evelyn:~$ sudo apt install caddy -y
eyendo lista de paquetes... Hecho
reando árbol de dependencias... Hecho
eyendo la información de estado... Hecho
e instalarán los siguientes paquetes adicionales:
 libnss3-tools
e instalarán los siguientes paquetes NUEVOS:
 caddy libnss3-tools
 actualizados, 2 nuevos se instalarán, 0 para eliminar y 9 no actualizados.
e necesita descargar 11,2 MB de archivos.
e utilizarán 41,2 MB de espacio de disco adicional después de esta operación.
es:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 libnss3-tools amd64 2:3.98-1build1 [615 kB]
es:2 http://es.archive.ubuntu.com/ubuntu noble-updates/universe amd64 caddy amd64 2.6.2-6ubuntu0.24.04.3 [10,5 MB]
escargados 11,2 MB en 1s (7.445 kB/s)
eleccionando el paquete libnss3-tools previamente no seleccionado.
Leyendo la base de datos ... 85128 ficheros o directorios instalados actualmente.)
reparando para desempaquetar .../libnss3-tools_2%3a3.98-1build1_amd64.deb ...
esempaquetando libnss3-tools (2:3.98-1build1) ...
eleccionando el paquete caddy previamente no seleccionado.
reparando para desempaquetar .../caddy_2.6.2-6ubuntu0.24.04.3_amd64.deb ...
esempaquetando caddy (2.6.2-6ubuntu0.24.04.3) ...
onfigurando libnss3-tools (2:3.98-1build1) ...
onfigurando caddy (2.6.2-6ubuntu0.24.04.3) ...
reated symlink /etc/systemd/system/multi-user.target.wants/caddy.service → /usr/lib/systemd/system/caddy.service.
rocesando disparadores para man-db (2.12.0-4build2) ...
canning processes...
canning linux images...

unning kernel seems to be up-to-date.

o services need to be restarted.

o containers need to be restarted.

o user sessions are running outdated binaries.

o VM guests are running outdated hypervisor (qemu) binaries on this host.
velyn@evelyn:~$
```

verificación de la instalación

```
evelyn@evelyn:~$ caddy version
2.6.2
evelyn@evelyn:~$
```

comprobación del estado del servicio

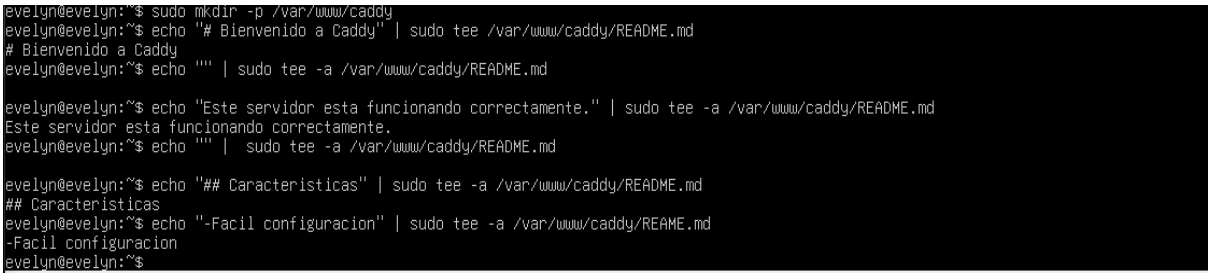


4. Crear directorio para Caddy

Comando: sudo mkdir -p /var/www/caddy
Descripción: Crea un directorio específico para los archivos de Caddy

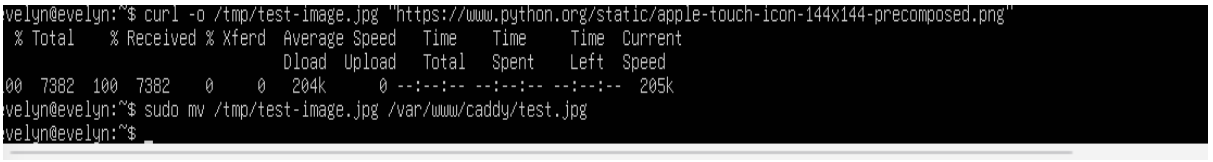
5. Crear archivo Markdown de prueba

Comando: echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md echo "" | sudo tee -a /var/www/caddy/README.md echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md echo "" | sudo tee -a /var/www/caddy/README.md echo "## Características" | sudo tee -a /var/www/caddy/README.md echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
Descripción: Crea un archivo Markdown con contenido de ejemplo.



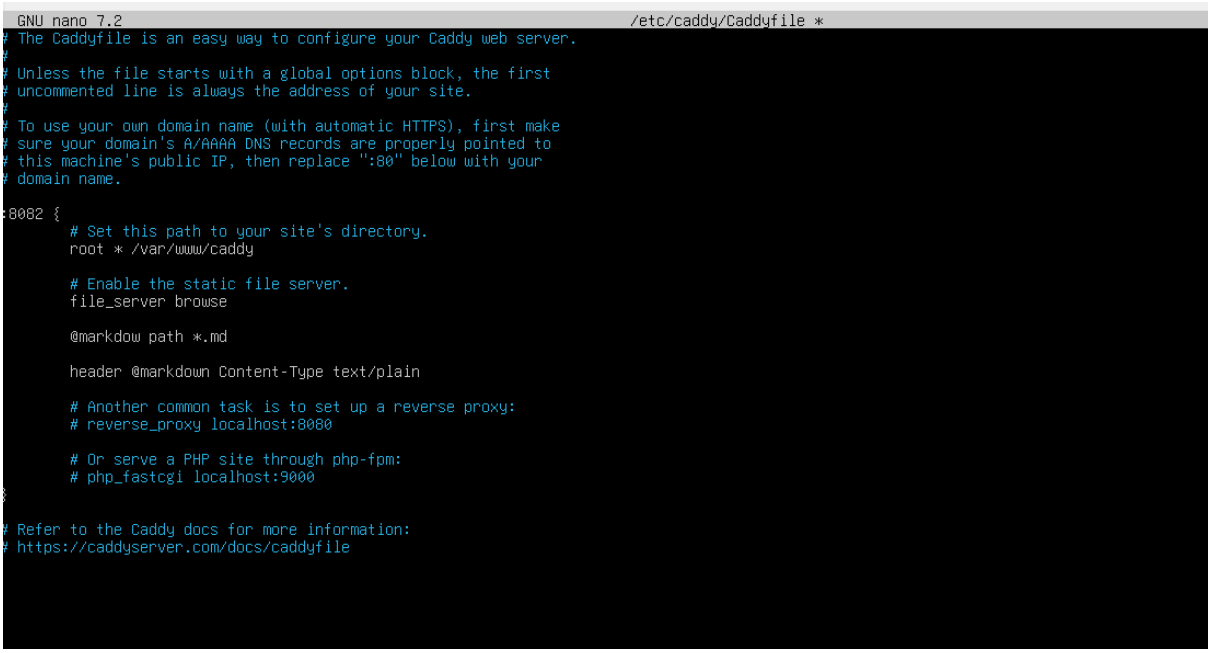
6. Crear imagen de prueba (cuidado wsl hay que hacer ajustes)

Comando: curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch icon-144x144-precomposed.png" sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg Descripción: Descarga una imagen de prueba para verificar que Caddy sirve archivos estáticos



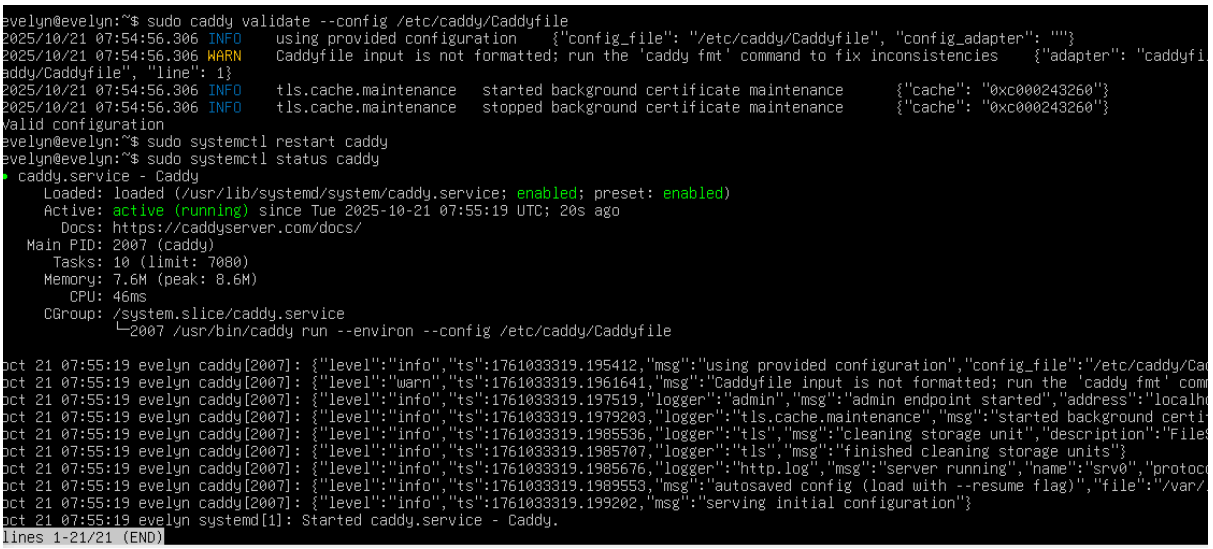
7. Crear Caddyfile personalizado

Comando: sudo nano /etc/caddy/Caddyfile
Descripción: Abre el archivo de configuración de Caddy.



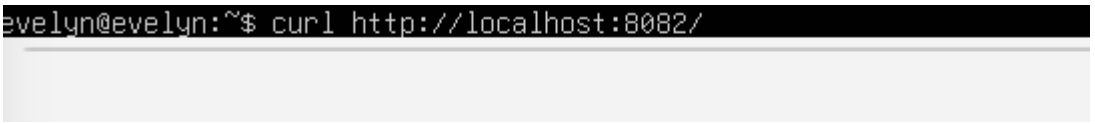
8 y 9 Reiniciar Caddy

Comando: sudo systemctl restart caddy
Descripción: Reinicia Caddy para aplicar la nueva configuración



10. Probar Caddy desde terminal

Comando: curl http://localhost:8082/
Descripción: Lista los archivos disponibles en el servidor Caddy



```
        filterEl.focus({preventScroll: true});

        function initFilter() {
            if (!filterEl.value) {
                var filterParam = new URL(window.location.href).searchParams.get('filter');
                if (filterParam) {
                    filterEl.value = filterParam;
                }
            }
            filter();
        }

        function filter() {
            var q = filterEl.value.trim().toLowerCase();
            var elems = document.querySelectorAll('tr.file');
            elems.forEach(function (el) {
                if (!q) {
                    el.style.display = '';
                    return;
                }
                var nameEl = el.querySelector('.name');
                var nameVal = nameEl.textContent.trim().toLowerCase();
                if (nameVal.indexOf(q) !== -1) {
                    el.style.display = '';
                } else {
                    el.style.display = 'none';
                }
            });
        }

        function localizeDatetime(e, index, ar) {
            if (e.textContent === undefined) {
                return;
            }
            var d = new Date(e.getAttribute('datetime'));
            if (isNaN(d)) {
                d = new Date(e.textContent);
                if (isNaN(d)) {
                    return;
                }
            }
            e.textContent = d.toLocaleString([], {day: "2-digit", month: "2-digit", year: "numeric", hour: "2-digit", minute: "2-digit", secc
        }
        var timeList = Array.prototype.slice.call(document.getElementsByTagName("time"));
        timeList.forEach(localizeDatetime);
    </script>
</body>

</html>
evelyn@evelyn:~$
```

11. Probar archivo Markdown

Comando: curl http://localhost:8082/README.md

Descripción: Verifica que Caddy sirve correctamente archivos Markdown.

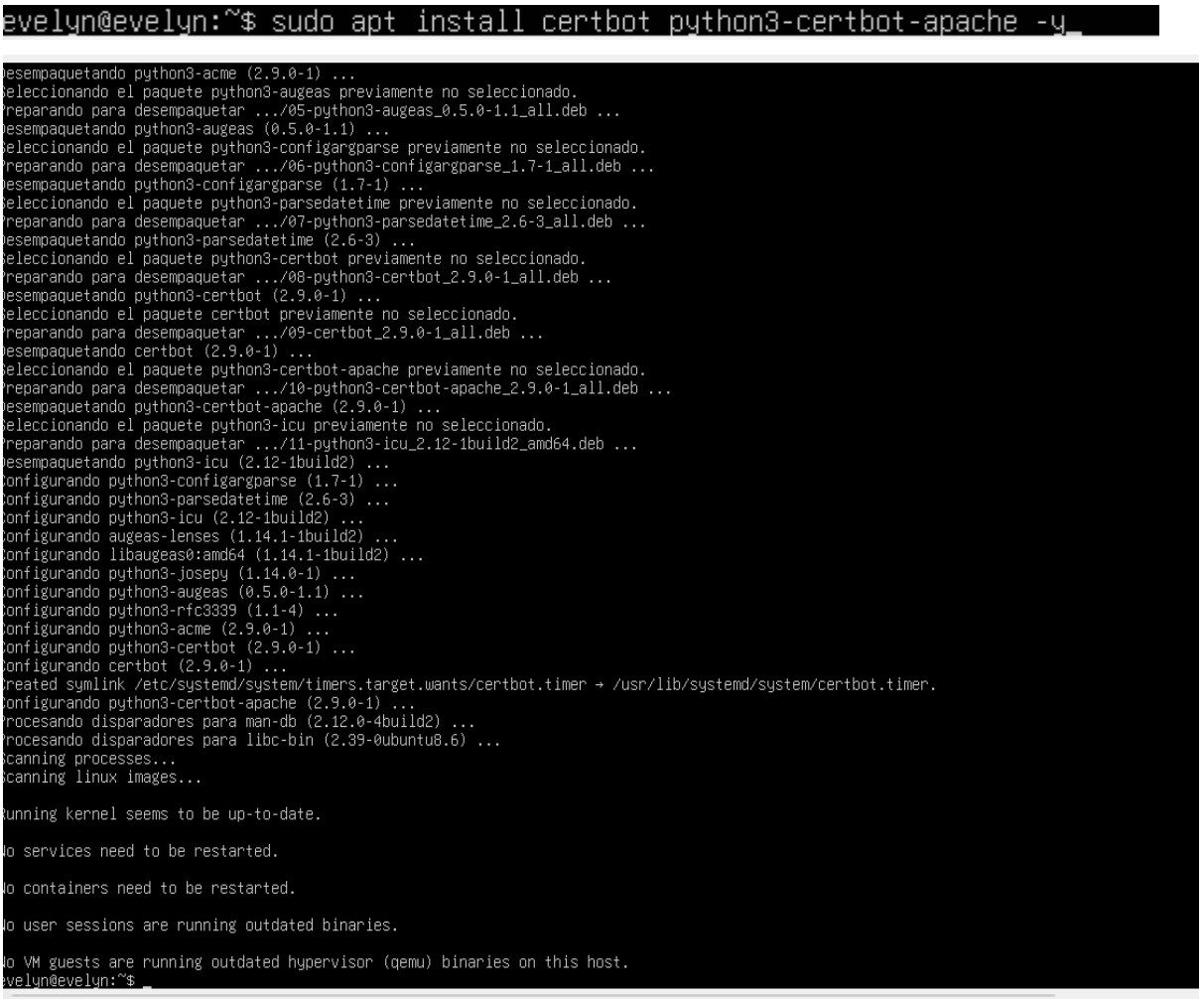
```
evelyn@evelyn:~$ curl http://localhost:8082/REAME.md
-Facil configuracion
evelyn@evelyn:~$
```

PARTE 4: CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

1. Instalar Certbot y el plugin de Apache

Comando: sudo apt install certbot python3-certbot-apache -y

Descripción: Instala Certbot y su integración con Apache para gestionar certificados SSL



2. Verificar dominio o usar localhost

Nota: Para obtener certificados reales de Let's Encrypt necesitas un dominio público.

Para esta práctica usaremos certificados autofirmados.

Comando: sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

Descripción: Crea un certificado autofirmado para practicar HTTPS localmente. Completa los campos solicitados (puedes usar valores por defecto)

```
evelyn@evelyn:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/c
rt_
```

```
Configurando python3-icu (2.12-1build2) ...
Configurando augeas-lenses (1.14.1-1build2) ...
Configurando libaugeas0:amd64 (1.14.1-1build2) ...
Configurando python3-josepy (1.14.0-1) ...
Configurando python3-augeas (0.5.0-1.1) ...
Configurando python3-rfc3339 (1.1-4) ...
Configurando python3-acme (2.9.0-1) ...
Configurando python3-certbot (2.9.0-1) ...
Configurando certbot (2.9.0-1) ...
Created symlink /etc/systemd/system/timers.target.wants/certbot.timer → /usr/lib/systemd/system/certbot.timer.
Configurando python3-certbot-apache (2.9.0-1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
evelyn@evelyn:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.c
rt
.....+........+...+.....+.+........+....+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.......+..+.......+..+.........+.+.....+.+..........
..+.....+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*..........+......+..+.......+...+.....+......+.+...+........++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++
.....+.....+.+...+..................+..+...+...+.+......+..+......+....+..+.+...+.....+....+..+..............+.............+..........+....+...+..+.+........
.......+..++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+...........+.+..........+.......+..++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++*.....+..+.+...+...........+.+............+..+......+...+.+.........+..+....+.........+......+........+............+.+..+...++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:es
State or Province Name (full name) [Some-State]:madrid
Locality Name (eg, city) []:madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:prometeo
Organizational Unit Name (eg, section) []:evelyn
Common Name (e.g. server FQDN or YOUR name) []:evelyn
Email Address []:evelynap@hotmail.com
evelyn@evelyn:~$ _
```

3. Habilitar módulo SSL en Apache

Comando: sudo a2enmod ssl

Descripción: Activa el módulo SSL necesario para HTTPS en Apache


```
evelyn@evelyn:~$ sudo a2enmod ssl
[sudo] password for evelyn:
Sorry, try again.
[sudo] password for evelyn:
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
evelyn@evelyn:~$
```

4. Crear configuración SSL para Apache

Comando: sudo nano /etc/apache2/sites-available/default-ssl.conf

Descripción: Edita el archivo y asegúrate de que incluye estas líneas dentro de *:443>: SSLEngine on SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

```
GNU nano 7.2                                    /etc/apache2/sites-available/default-ssl.conf
      #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

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

      #   A self-signed (snakeoil) certificate can be created by installing
      #   the ssl-cert package. See
      #   /usr/share/doc/apache2/README.Debian.gz for more info.
      #   If both key and certificate are stored in the same file, only the
      #   SSLCertificateFile directive is needed.
      SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
      SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key

      #   Server Certificate Chain:
      #   Point SSLCertificateChainFile at a file containing the
      #   concatenation of PEM encoded CA certificates which form the
      #   certificate chain for the server certificate. Alternatively
      #   the referenced file can be the same as SSLCertificateFile
      #   when the CA certificates are directly appended to the server
      #   certificate for convinience.
      #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

      #   Certificate Authority (CA):
      #   Set the CA certificate verification path where to find CA
      #   certificates for client authentication or alternatively one
      #   huge file containing all of them (file must be PEM encoded)
      #   Note: Inside SSLCACertificatePath you need hash symlinks
      #         to point to the certificate files. Use the provided
      #         Makefile to update the hash symlinks after changes.
      #SSLCACertificatePath /etc/ssl/certs/
      #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

      #   Certificate Revocation Lists (CRL):
      #   Set the CA revocation path where to find CA CRLs for client
      #   authentication or alternatively one huge file containing all
      #   of them (file must be PEM encoded)
      #   Note: Inside SSLCARevocationPath you need hash symlinks
```

5. Cambiar puerto SSL

Comando: sudo nano /etc/apache2/ports.conf

Descripción: Añade la línea Listen 8443 para que Apache escuche HTTPS en puerto 8443

```
  GNU nano 7.2                                                    /etc/apa
# 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
Listen 8443


<IfModule ssl_module>
        Listen 443
</IfModule>

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

6. Modificar VirtualHost SSL

Comando: sudo nano /etc/apache2/sites-available/default-ssl.conf

Descripción: Cambia por `<VirtualHost *:443>` por `<VirtualHost *:8443>`.

```
 GNU nano 7.2                                    /etc/apache2/sites-available/default-ss
<VirtualHost *:8443>:
        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

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

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
        SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key

        #   Server Certificate Chain:
        #   Point SSLCertificateChainFile at a file containing the
        #   concatenation of PEM encoded CA certificates which form the
        #   certificate chain for the server certificate. Alternatively
        #   the referenced file can be the same as SSLCertificateFile
        #   when the CA certificates are directly appended to the server
        #   certificate for convinience.
        #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

        #   Certificate Authority (CA):
        #   Set the CA certificate verification path where to find CA
        #   certificates for client authentication or alternatively one
        #   huge file containing all of them (file must be PEM encoded)


evelyn@evelyn:~$
```

7. Habilitar sitio SSL

Comando: sudo a2ensite default-ssl.conf

Descripción: Activa la configuración SSL en Apache.

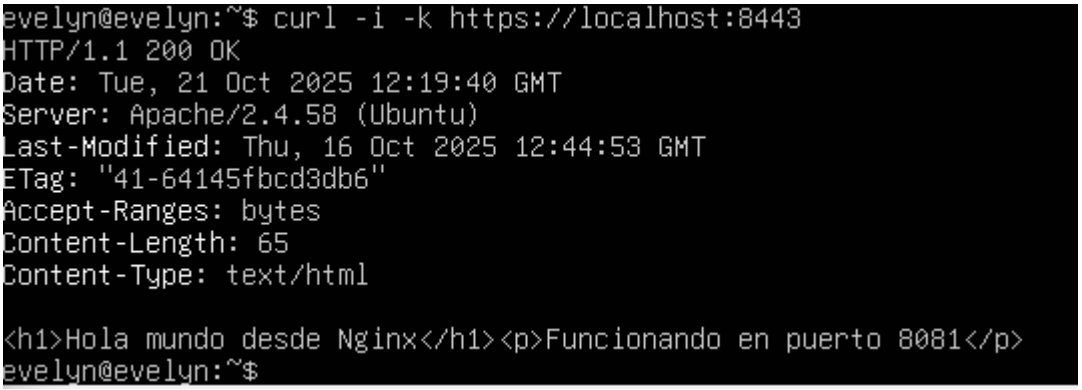
```
evelyn@evelyn:~$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
evelyn@evelyn:~$
```

8. Reiniciar Apache

Comando: sudo systemctl restart apache2
Descripción: Aplica todos los cambios de configuración SSL.

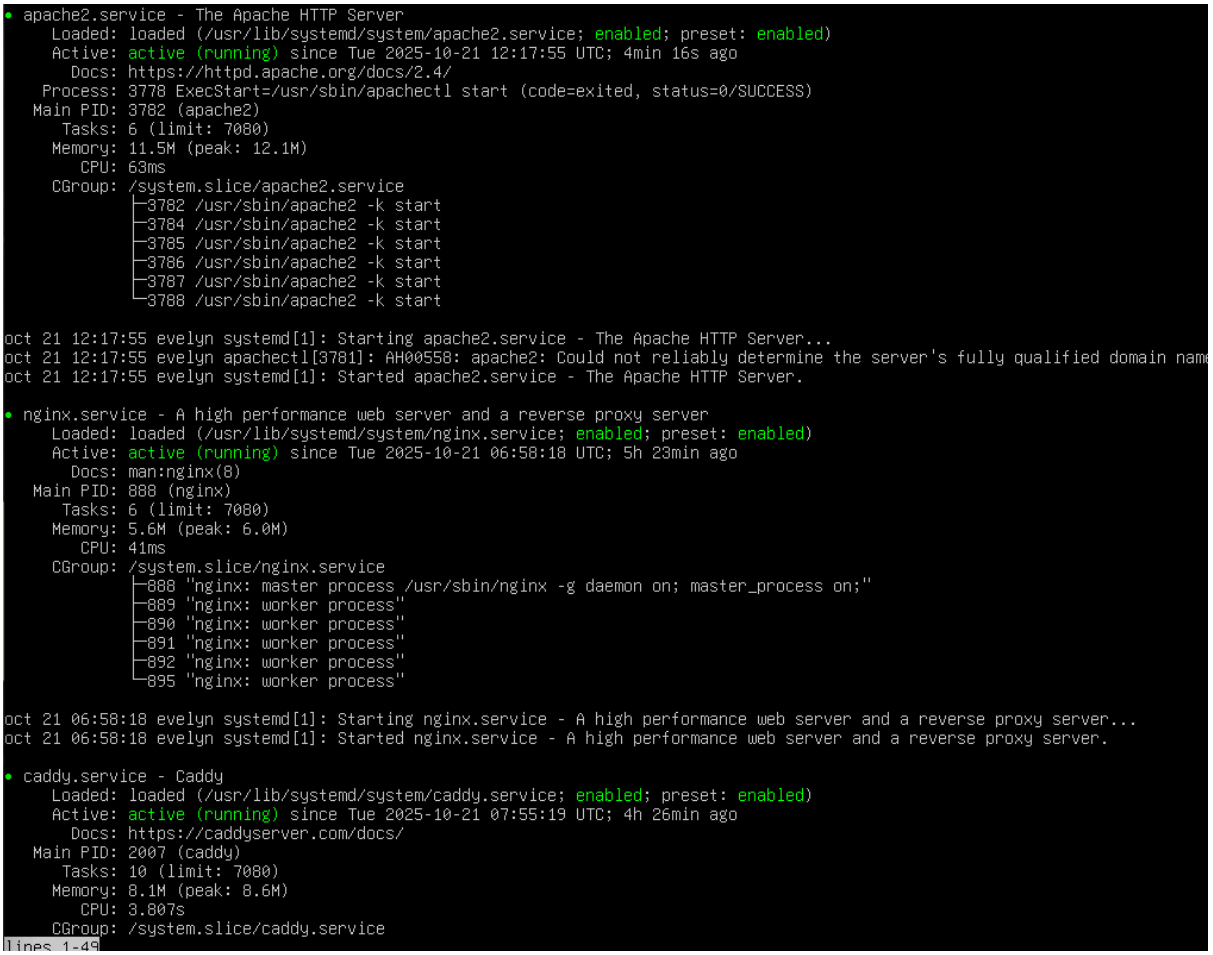
9. Verificar HTTPS

Comando: curl -i -k https://localhost:8443
Descripción: Prueba la conexión HTTPS (el flag -k ignora el aviso del certificado autofirmado).



PARTE 5: VERIFICACIÓN FINAL DE LOS TRES SERVIDORES

1. Verificar que todos los servicios están activos
Comando: sudo systemctl status apache2 nginx caddy
Descripción: Muestra el estado de los tres servidores simultáneamente.



2. Verificar puertos en uso

Comando: sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
Descripción: Lista los puertos donde están escuchando los servidores.