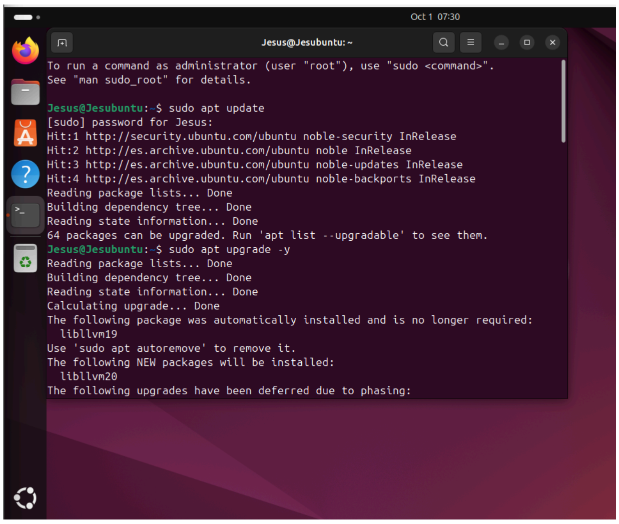
Jesús García

Configuración avanzada de Servidores Web y HTTPS

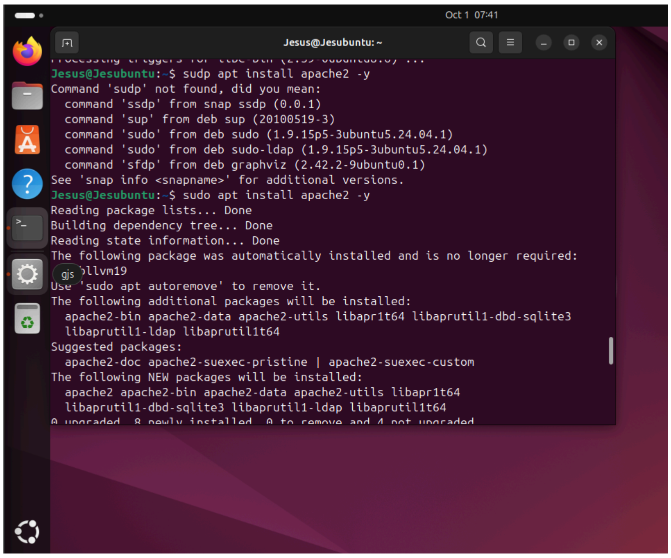
PARTE 1

1. Actualizar el sistema
sudo apt update && sudo apt upgrade -y



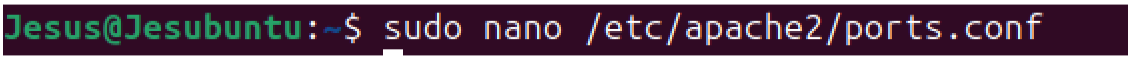
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Jesus@Jesubuntu:~$ sudo apt update
[sudo] password for Jesus:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
64 packages can be upgraded. Run 'apt list --upgradable' to see them.
Jesus@Jesubuntu:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  libllvm20
The following upgrades have been deferred due to phasing:
```

2. Instalar Apache
sudo apt install apache2 -y



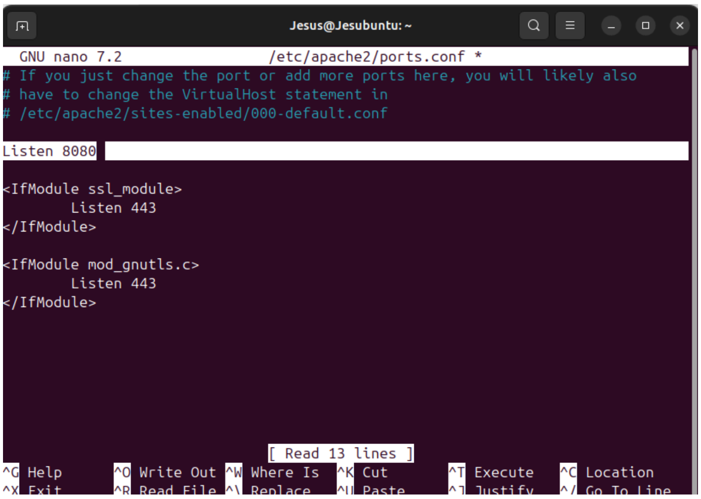
```
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Jesus@Jesubuntu:~$ sudp apt install apache2 -y
Command 'sudp' not found, did you mean:
  command 'ssdp' from snap ssdp (0.0.1)
  command 'sup' from deb sup (20100519-3)
  command 'sudo' from deb sudo (1.9.15p5-3ubuntu5.24.04.1)
  command 'sudo' from deb sudo-ldap (1.9.15p5-3ubuntu5.24.04.1)
  command 'sfdp' from deb graphviz (2.42.2-9ubuntu0.1)
See 'snap info <snapname>' for additional versions.
Jesus@Jesubuntu:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 upgraded, 8 newly installed, 0 to remove and 4 not upgraded.
```

3. Cambiar el puerto a 8080
sudo nano /etc/apache2/ports.conf



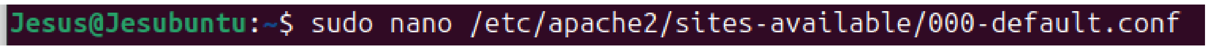
```
Jesus@Jesubuntu:~$ sudo nano /etc/apache2/ports.conf
```

-Cambiar `Listen 80` por:
Listen 8080



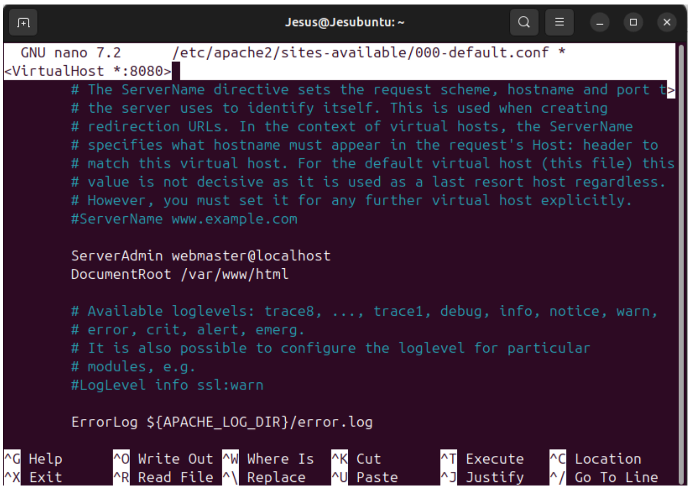
```
  GNU nano 7.2                  /etc/apache2/ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
        Listen 443
</IfModule>

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




                               [ Read 13 lines ]
^G Help       ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```

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



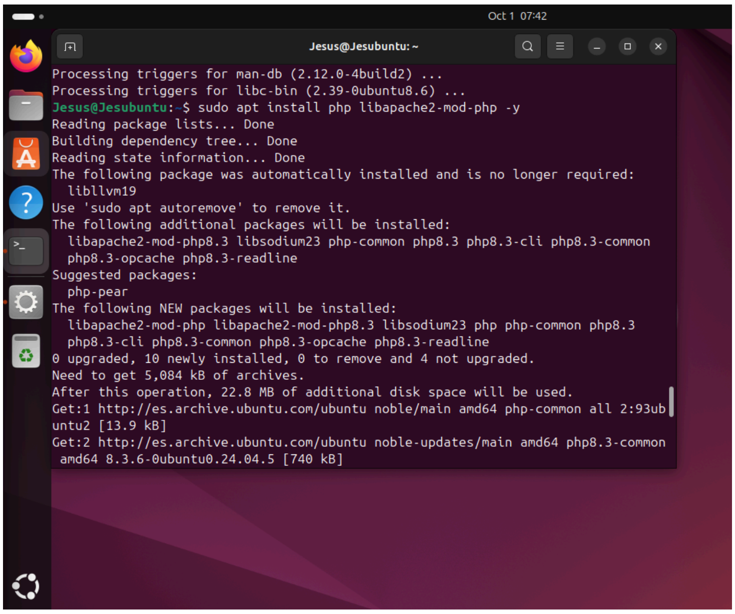
```
Jesus@Jesubuntu:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

-Cambiar `<VirtualHost *:80>` por:
<VirtualHost *:8080>



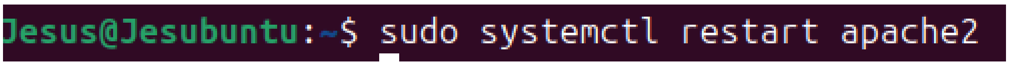
```
  GNU nano 7.2       /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:8080>
        # The ServerName directive sets the request scheme, hostname and port t>
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual 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

^G Help       ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste      ^J Justify    ^/ Go To Line
```

5. Instalar PHP
sudo apt install php libapache2-mod-php -y



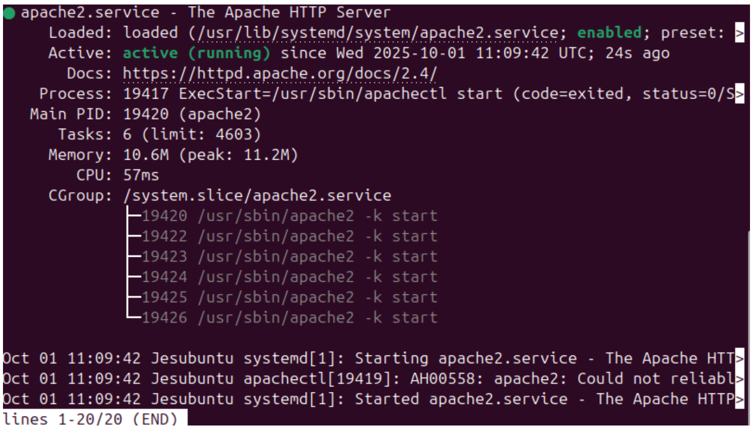
```
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Jesus@Jesubuntu:~$ sudo apt install php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libapache2-mod-php8.3 libsodium23 php-common php8.3 php8.3-cli php8.3-common
  php8.3-opcache php8.3-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  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 upgraded, 10 newly installed, 0 to remove and 4 not upgraded.
Need to get 5,084 kB of archives.
After this operation, 22.8 MB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 php-common all 2:93ub
untu2 [13.9 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 php8.3-common
 amd64 8.3.6-0ubuntu0.24.04.5 [740 kB]
```

6. Reiniciar Apache
sudo systemctl restart apache2



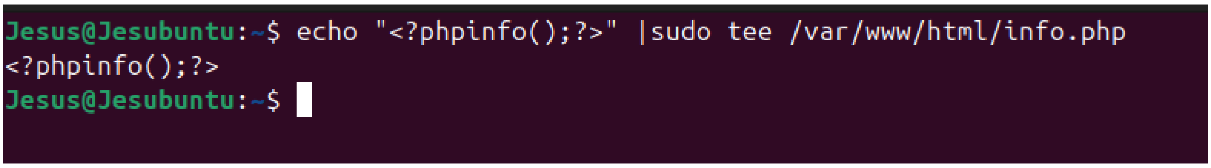
```
Jesus@Jesubuntu:~$ sudo systemctl restart apache2
```

7. Comprobar estado
sudo systemctl status apache2



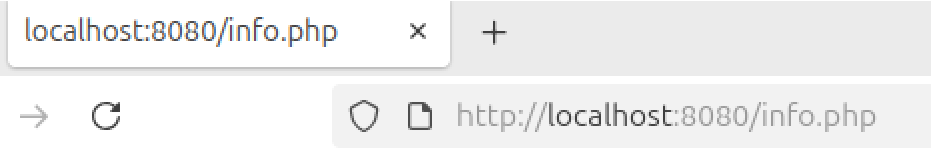
```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
     Active: active (running) since Wed 2025-10-01 11:09:42 UTC; 24s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 19417 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
   Main PID: 19420 (apache2)
      Tasks: 6 (limit: 4603)
     Memory: 10.6M (peak: 11.2M)
        CPU: 57ms
     CGroup: /system.slice/apache2.service
             ├─19420 /usr/sbin/apache2 -k start
             ├─19422 /usr/sbin/apache2 -k start
             ├─19423 /usr/sbin/apache2 -k start
             ├─19424 /usr/sbin/apache2 -k start
             ├─19425 /usr/sbin/apache2 -k start
             └─19426 /usr/sbin/apache2 -k start

Oct 01 11:09:42 Jesubuntu systemd[1]: Starting apache2.service - The Apache HTT>
Oct 01 11:09:42 Jesubuntu apachectl[19419]: AH00558: apache2: Could not reliabl>
Oct 01 11:09:42 Jesubuntu systemd[1]: Started apache2.service - The Apache HTTP>
lines 1-20/20 (END)
```

8. Crear archivo PHP de prueba
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php



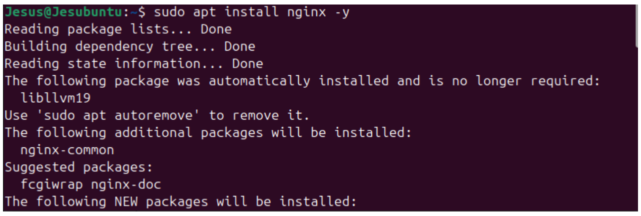
```
Jesus@Jesubuntu:~$ echo "<?phpinfo();?>"  |sudo tee /var/www/html/info.php
<?phpinfo();?>
Jesus@Jesubuntu:~$ 
```

9. Probar Apache
curl http://localhost:8080/info.php



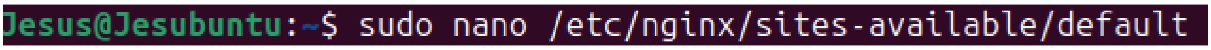
PARTE 2: Nginx

1. Instalar Nginx
sudo apt install nginx -y



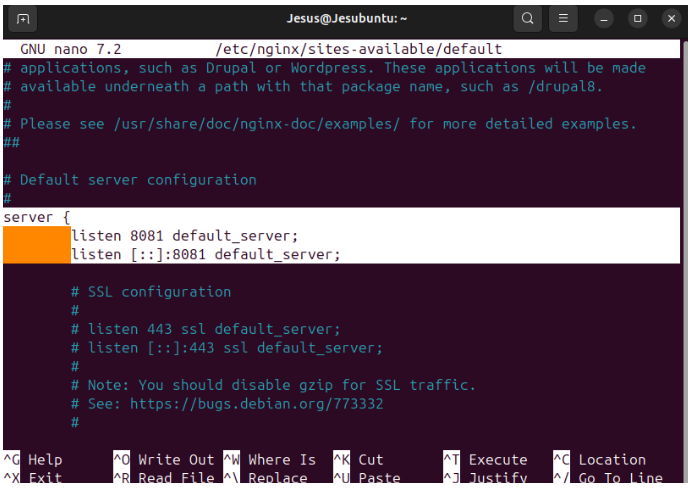
```
Jesus@Jesubuntu:~$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
```

2. Cambiar puerto
sudo nano /etc/nginx/sites-available/default



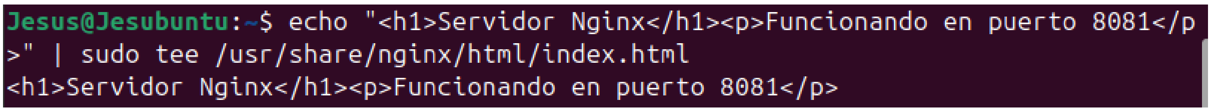
```
Jesus@Jesubuntu:~$ sudo nano /etc/nginx/sites-available/default
```

Cambia `listen 80` por:
listen 8081;



```
  GNU nano 7.2            /etc/nginx/sites-available/default
# 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;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

3. Crear página HTML
echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee /usr/share/nginx/html/index.html

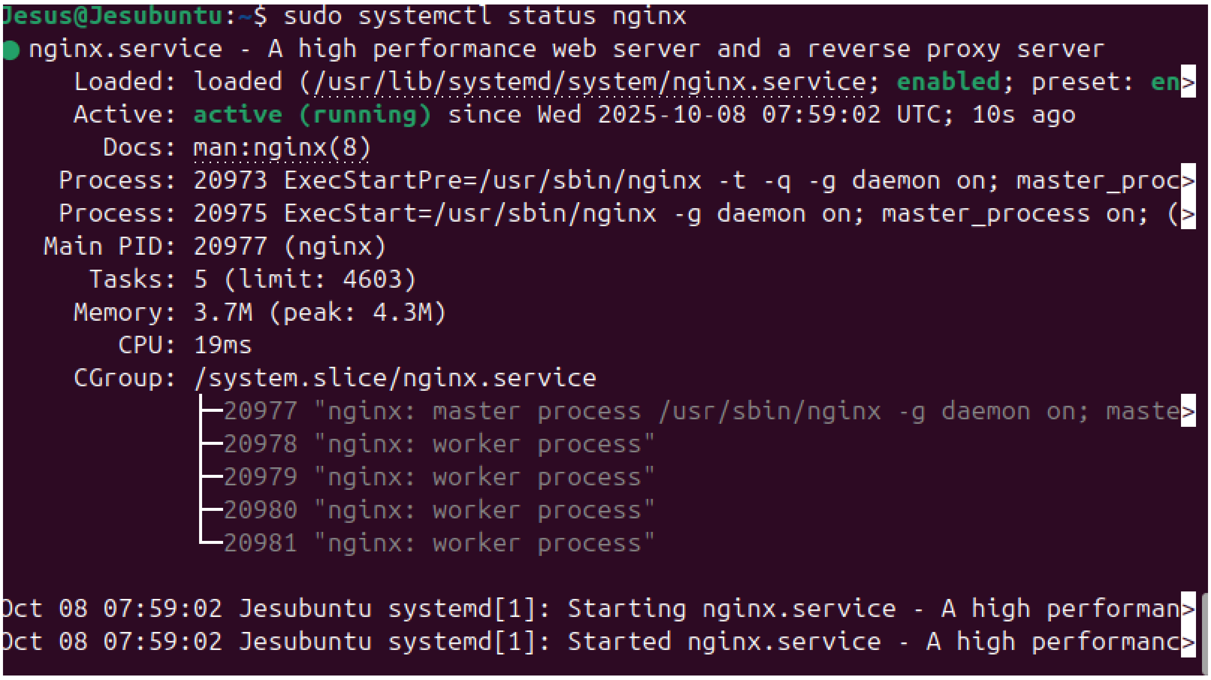


```
Jesus@Jesubuntu:~$ echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p
>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
```

4. Reiniciar Nginx
sudo systemctl restart nginx

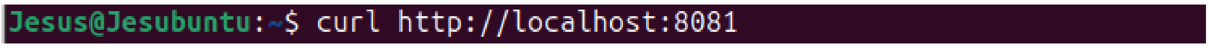
```
Jesus@Jesubuntu:~$ sudo systemctl restart nginx
```

5. Verificar estado
sudo systemctl status nginx

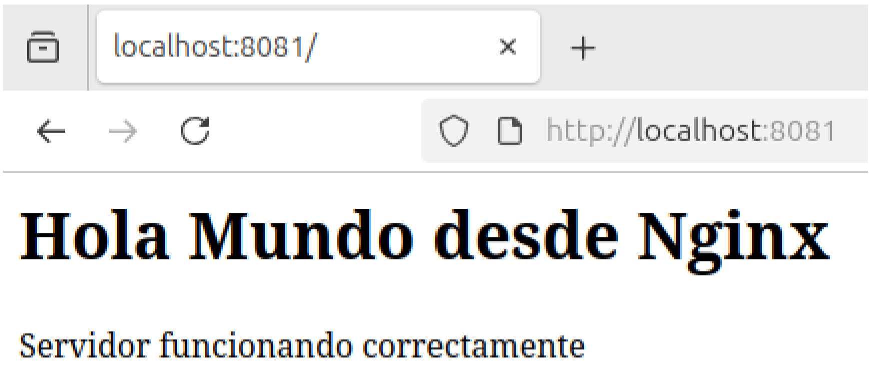


```
Jesus@Jesubuntu:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
     Active: active (running) since Wed 2025-10-08 07:59:02 UTC; 10s ago
       Docs: man:nginx(8)
    Process: 20973 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proc>
    Process: 20975 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (>
   Main PID: 20977 (nginx)
      Tasks: 5 (limit: 4603)
     Memory: 3.7M (peak: 4.3M)
        CPU: 19ms
     CGroup: /system.slice/nginx.service
             ├─20977 "nginx: master process /usr/sbin/nginx -g daemon on; maste>
             ├─20978 "nginx: worker process"
             ├─20979 "nginx: worker process"
             ├─20980 "nginx: worker process"
             └─20981 "nginx: worker process"

Oct 08 07:59:02 Jesubuntu systemd[1]: Starting nginx.service - A high performan>
Oct 08 07:59:02 Jesubuntu systemd[1]: Started nginx.service - A high performanc>
```

6. Probar
curl http://localhost:8081



```
Jesus@Jesubuntu:~$ curl http://localhost:8081
```

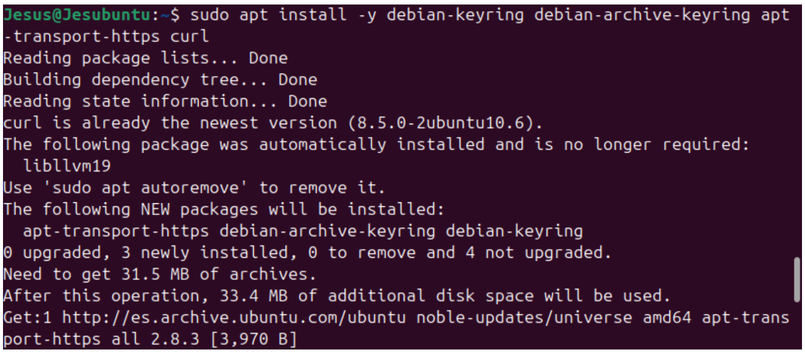


Hola Mundo desde Nginx

Servidor funcionando correctamente

PARTE 3: Caddy

1. Instalar dependencias
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl



```
Jesus@Jesubuntu:~$ sudo apt install -y debian-keyring debian-archive-keyring apt
-transport-https curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  apt-transport-https debian-archive-keyring debian-keyring
0 upgraded, 3 newly installed, 0 to remove and 4 not upgraded.
Need to get 31.5 MB of archives.
After this operation, 33.4 MB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu noble-updates/universe amd64 apt-trans
port-https all 2.8.3 [3,970 B]
```

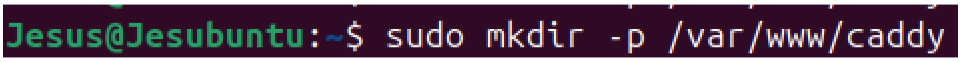
2. Añadir repositorio
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list

3. Instalar Caddy
sudo apt update && sudo apt install caddy -y

```
Jesus@Jesubuntu:~$ sudo apt update
```

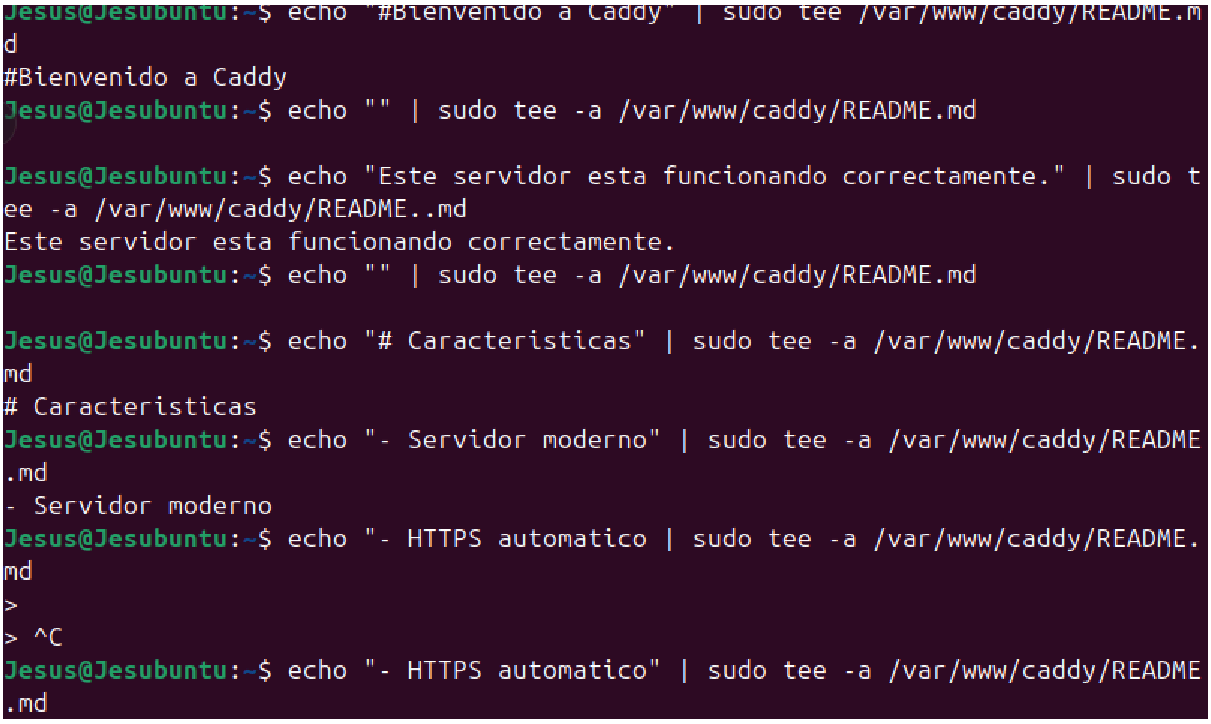
```
Jesus@Jesubuntu:~$ sudo apt install caddy -y
```

4. Crear directorio
sudo mkdir -p /var/www/caddy

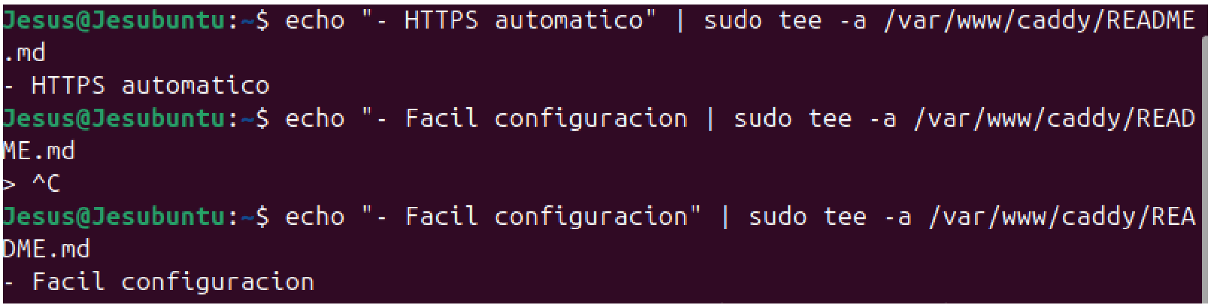


```
Jesus@Jesubuntu:~$ sudo mkdir -p /var/www/caddy
```

5. Crear archivo Markdown
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



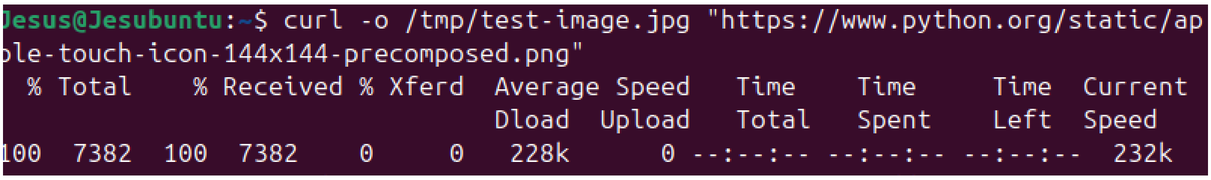
```
Jesus@Jesubuntu:~$ echo "#Bienvenido a Caddy" | sudo tee /var/www/caddy/README.m
d
#Bienvenido a Caddy
Jesus@Jesubuntu:~$ echo "" | sudo tee -a /var/www/caddy/README.md

Jesus@Jesubuntu:~$ echo "Este servidor esta funcionando correctamente." | sudo t
ee -a /var/www/caddy/README..md
Este servidor esta funcionando correctamente.
Jesus@Jesubuntu:~$ echo "" | sudo tee -a /var/www/caddy/README.md

Jesus@Jesubuntu:~$ echo "# Caracteristicas" | sudo tee -a /var/www/caddy/README.
md
# Caracteristicas
Jesus@Jesubuntu:~$ echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README
.md
- Servidor moderno
Jesus@Jesubuntu:~$ echo "- HTTPS automatico | sudo tee -a /var/www/caddy/README.
md
>
> ^C
Jesus@Jesubuntu:~$ echo "- HTTPS automatico" | sudo tee -a /var/www/caddy/README
.md
```



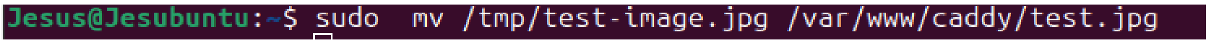
```
Jesus@Jesubuntu:~$ echo "- HTTPS automatico" | sudo tee -a /var/www/caddy/README
.md
- HTTPS automatico
Jesus@Jesubuntu:~$ echo "- Facil configuracion | sudo tee -a /var/www/caddy/READ
ME.md
> ^C
Jesus@Jesubuntu:~$ echo "- Facil configuracion" | sudo tee -a /var/www/caddy/REA
DME.md
- Facil configuracion
```

6. Descargar imagen

```
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
```



```
Jesus@Jesubuntu:~$ curl -o /tmp/test-image.jpg "https://www.python.org/static/ap
ple-touch-icon-144x144-precomposed.png"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7382  100  7382    0     0   228k      0 --:--:-- --:--:-- --:--:--  232k
```



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

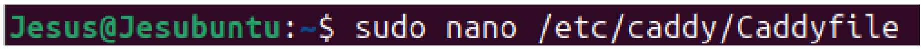
7. Configurar Caddyfile

```
sudo nano /etc/caddy/Caddyfile
```

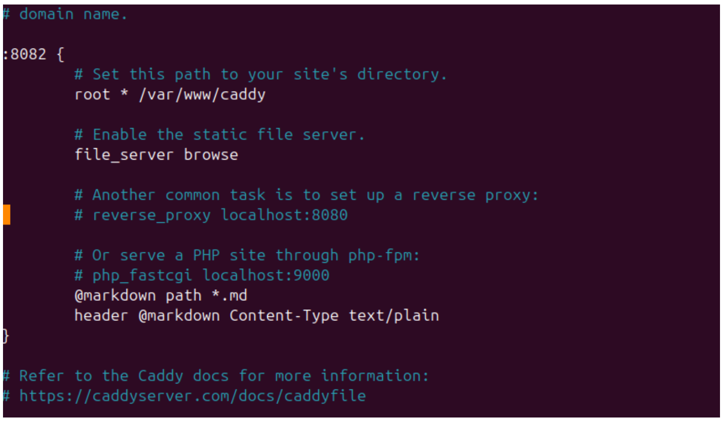
Pega esto:

```
:8082 {
   root * /var/www/caddy
   file_server browse

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

}



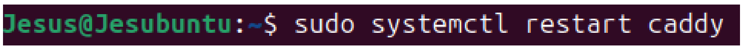
```
Jesus@Jesubuntu:~$ sudo nano /etc/caddy/Caddyfile
```



```
# domain name.

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

        # Enable the static file server.
        file_server browse

        # Another common task is to set up a reverse proxy:
        # reverse_proxy localhost:8080

        # Or serve a PHP site through php-fpm:
        # php_fastcgi localhost:9000
        @markdown path *.md
        header @markdown Content-Type text/plain
}

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

8. Reiniciar Caddy

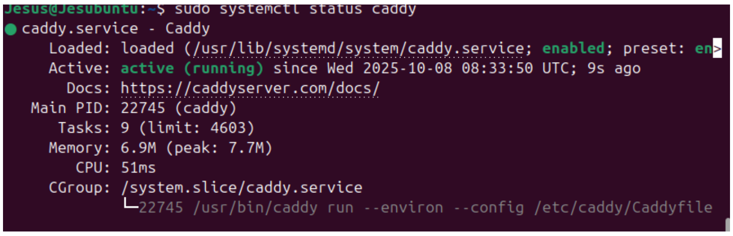
sudo systemctl restart caddy



```
Jesus@Jesubuntu:~$ sudo systemctl restart caddy
```

9. Verificar estado

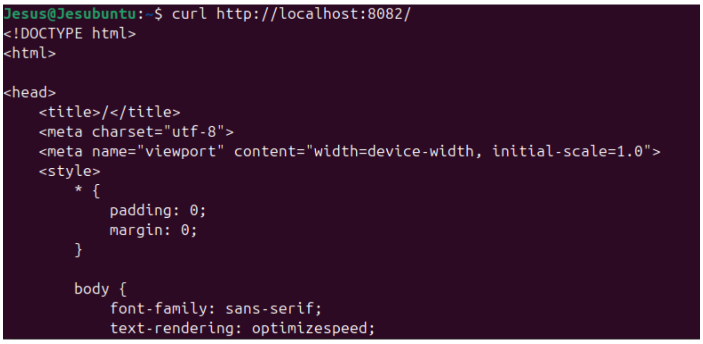
sudo systemctl status caddy



```
Jesus@Jesubuntu:~$ sudo systemctl status caddy
● caddy.service - Caddy
     Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: en>
     Active: active (running) since Wed 2025-10-08 08:33:50 UTC; 9s ago
       Docs: https://caddyserver.com/docs/
   Main PID: 22745 (caddy)
      Tasks: 9 (limit: 4603)
     Memory: 6.9M (peak: 7.7M)
        CPU: 51ms
     CGroup: /system.slice/caddy.service
             └─22745 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

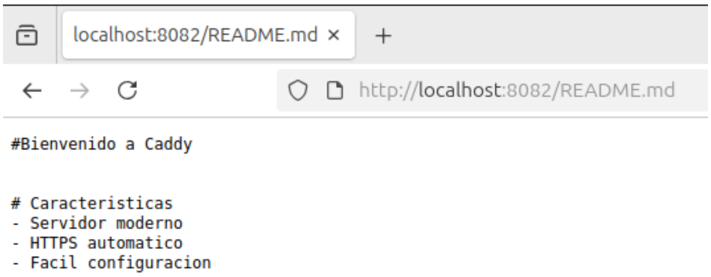
10. Probar

curl http://localhost:8082/



```
Jesus@Jesubuntu:~$ 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;
```

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

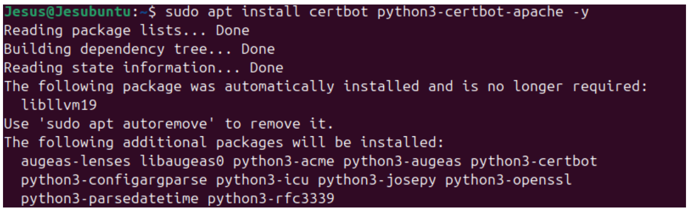


```
#Bienvenido a Caddy


# Caracteristicas
- Servidor moderno
- HTTPS automatico
- Facil configuracion
```

HTTPS en Apache

1. Instalar Certbot

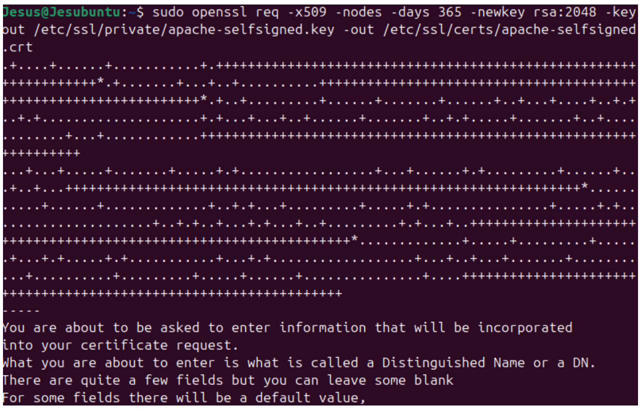
sudo apt install certbot python3-certbot-apache -y



```
Jesus@Jesubuntu:~$ sudo apt install certbot python3-certbot-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot
  python3-configargparse python3-icu python3-josepy python3-openssl
  python3-parsedatetime python3-rfc3339
```

2. Crear certificado autofirmado

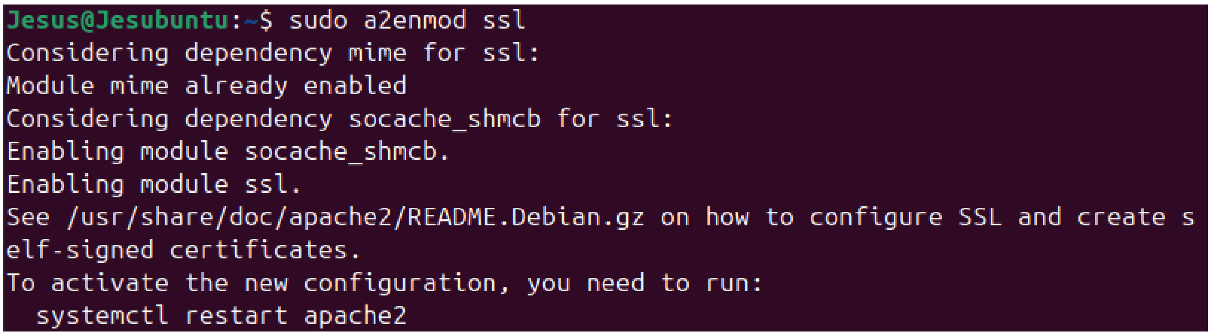
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/apache-selfsigned.key \
-out /etc/ssl/certs/apache-selfsigned.crt



```
Jesus@Jesubuntu:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -key
out /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned
.crt
.+....+......+..........+.++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++*.+.......+...+..+..........+++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++*.+..+.........+.....+.......+......+..+...+....+..+.+
..+.+..................+.+...+...+..+......+.......+..+.+.....+.......+..+....
........+...+...........++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++
...+...+.....+.......+.....+.+................+...+......+.+.........+......+..
.+..+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*......
.....+......+............+..+.+...+.........+.....+.+...............+.....+.+..
...................+..+.+..+...+.+...+..+.........+.+...+..+++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++*............+.....+.........+.....
.+...+.+.....+.+..........+...+.+.................+...+..+...+.......+.......
...+..........+.........+.....+......+...............+....++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++
-----
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,
```

3. Activar SSL

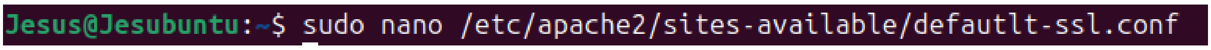
sudo a2enmod ssl



```
Jesus@Jesubuntu:~$ 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 s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

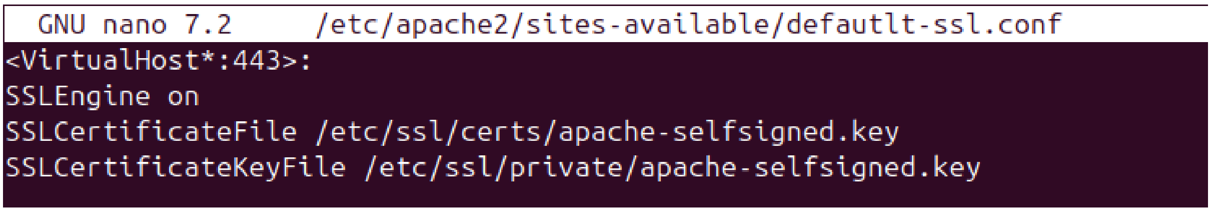
4. Editar configuración SSL

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



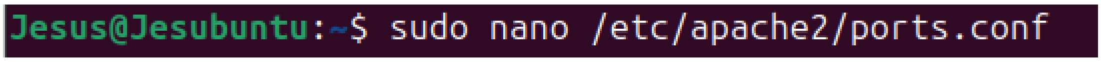
```
Jesus@Jesubuntu:~$ sudo nano /etc/apache2/sites-available/defautlt-ssl.conf
```

Asegúrate de tener esto dentro de `<VirtualHost *:443>`:



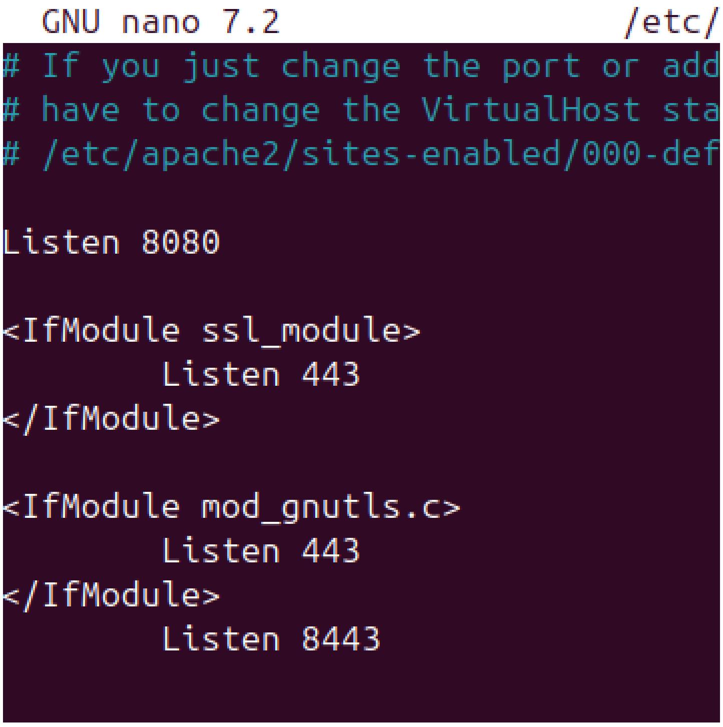
```
  GNU nano 7.2      /etc/apache2/sites-available/defautlt-ssl.conf
<VirtualHost*:443>:
SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.key
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

5. Escuchar en el puerto 8443
sudo nano /etc/apache2/ports.conf



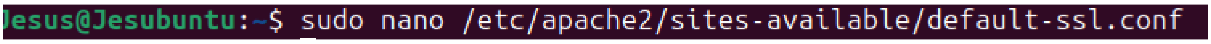
```
Jesus@Jesubuntu:~$ sudo nano /etc/apache2/ports.conf
```

Añade:
Listen 8443



```
  GNU nano 7.2                    /etc/
# If you just change the port or add
# have to change the VirtualHost sta
# /etc/apache2/sites-enabled/000-def

Listen 8080

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>
        Listen 8443
```

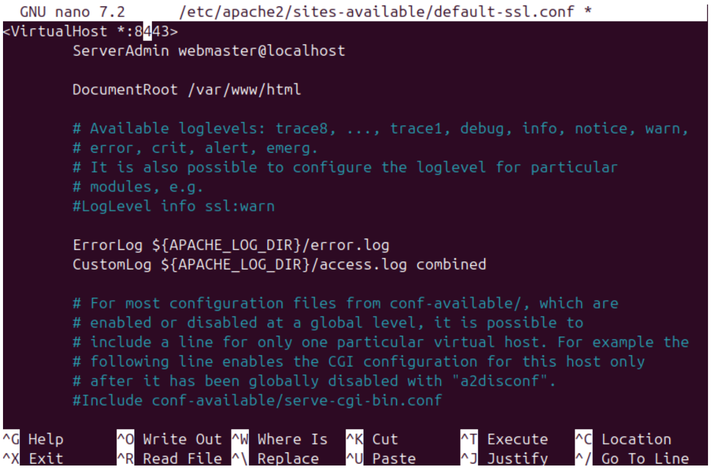
6. Cambiar VirtualHost SSL a 8443
sudo nano /etc/apache2/sites-available/default-ssl.conf



```
Jesus@Jesubuntu:~$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Cambia:
<VirtualHost *:443>

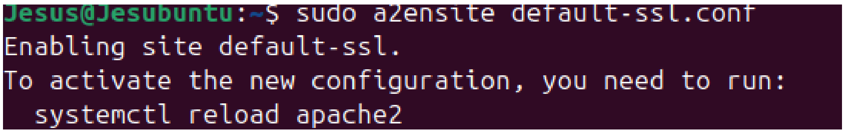
por
<VirtualHost *:8443>



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

^G Help      ^O Write Out ^W Where Is ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste     ^J Justify   ^/ Go To Line
```

7. Activar sitio SSL

sudo a2ensite default-ssl.conf



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

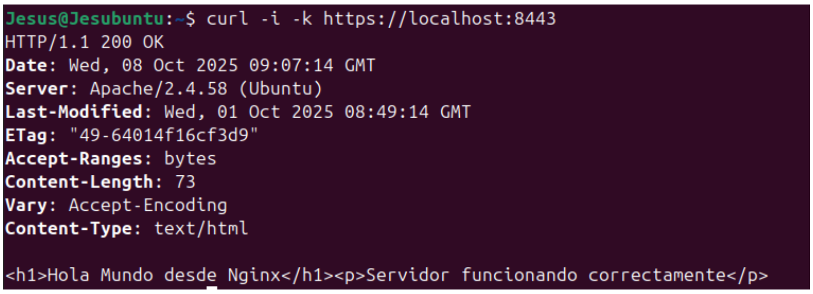
8. Reiniciar Apache

sudo systemctl restart apache2

```
Jesus@Jesubuntu:~$ sudo systemctl restart apache2
```

9. Probar HTTPS

curl -i -k https://localhost:8443

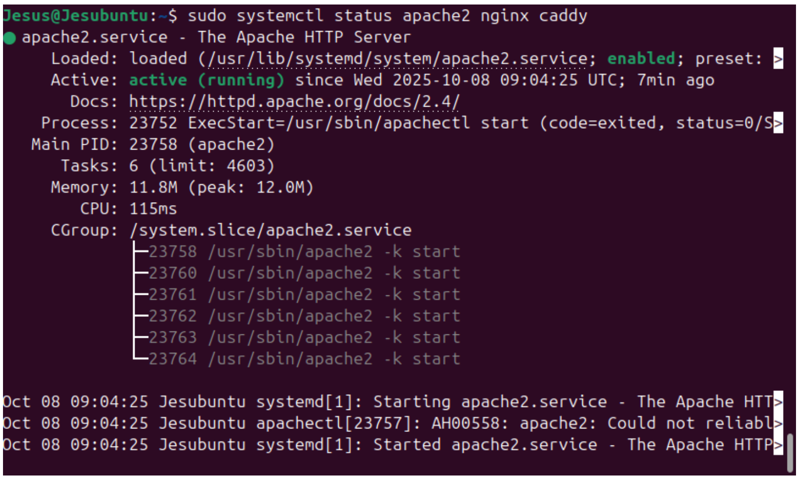


```
Jesus@Jesubuntu:~$ curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Wed, 08 Oct 2025 09:07:14 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Wed, 01 Oct 2025 08:49:14 GMT
ETag: "49-64014f16cf3d9"
Accept-Ranges: bytes
Content-Length: 73
Vary: Accept-Encoding
Content-Type: text/html

<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

PARTE 4: Verificación Final

1. Ver todos los servicios activos

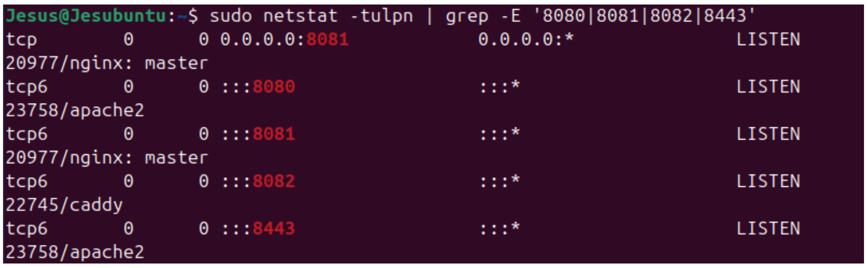
sudo systemctl status apache2 nginx caddy



```
Jesus@Jesubuntu:~$ sudo systemctl status apache2 nginx caddy
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
     Active: active (running) since Wed 2025-10-08 09:04:25 UTC; 7min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 23752 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
   Main PID: 23758 (apache2)
      Tasks: 6 (limit: 4603)
     Memory: 11.8M (peak: 12.0M)
        CPU: 115ms
     CGroup: /system.slice/apache2.service
             ├─23758 /usr/sbin/apache2 -k start
             ├─23760 /usr/sbin/apache2 -k start
             ├─23761 /usr/sbin/apache2 -k start
             ├─23762 /usr/sbin/apache2 -k start
             ├─23763 /usr/sbin/apache2 -k start
             └─23764 /usr/sbin/apache2 -k start

Oct 08 09:04:25 Jesubuntu systemd[1]: Starting apache2.service - The Apache HTT>
Oct 08 09:04:25 Jesubuntu apachectl[23757]: AH00558: apache2: Could not reliabl>
Oct 08 09:04:25 Jesubuntu systemd[1]: Started apache2.service - The Apache HTTP>
```

2. Ver puertos en uso

sudo netstat -tulpn | grep -E '8080|8081|8082|8443'

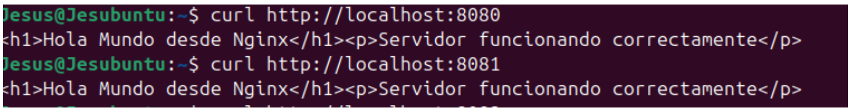


```
Jesus@Jesubuntu:~$ sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp        0      0 0.0.0.0:8081            0.0.0.0:*               LISTEN
20977/nginx: master
tcp6       0      0 :::8080                 :::*                    LISTEN
23758/apache2
tcp6       0      0 :::8081                 :::*                    LISTEN
20977/nginx: master
tcp6       0      0 :::8082                 :::*                    LISTEN
22745/caddy
tcp6       0      0 :::8443                 :::*                    LISTEN
23758/apache2
```

3. Probar todos los servidores

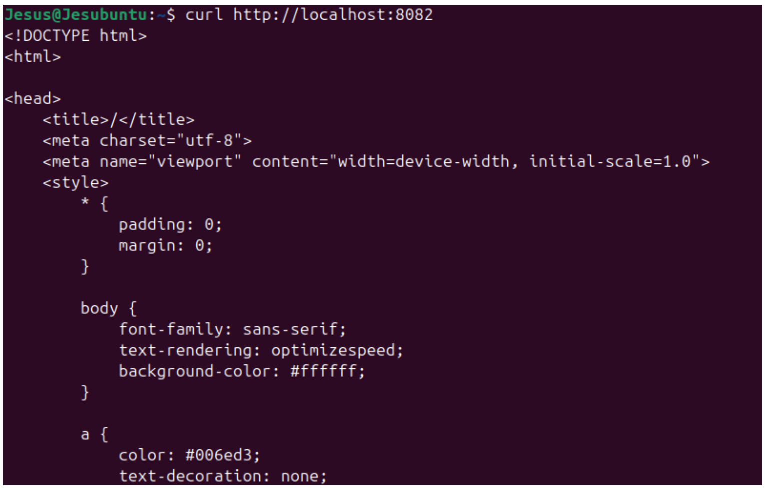
curl http://localhost:8080

curl http://localhost:8081



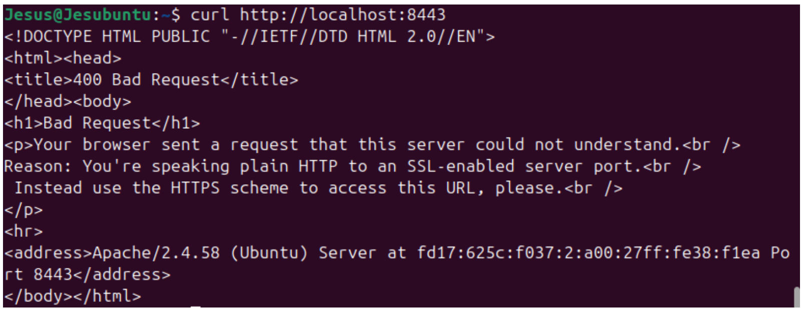
```
Jesus@Jesubuntu:~$ curl http://localhost:8080
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
Jesus@Jesubuntu:~$ curl http://localhost:8081
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

curl http://localhost:8082



```
Jesus@Jesubuntu:~$ 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;
        }

        a {
            color: #006ed3;
            text-decoration: none;
```

curl -k https://localhost:8443



```
Jesus@Jesubuntu:~$ curl http://localhost:8443
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
Reason: You're speaking plain HTTP to an SSL-enabled server port.<br />
 Instead use the HTTPS scheme to access this URL, please.<br />
</p>
<hr>
<address>Apache/2.4.58 (Ubuntu) Server at fd17:625c:f037:2:a00:27ff:fe38:f1ea Po
rt 8443</address>
</body></html>
```