

Apache 2

Manuel Querol Aicart

2ºDAW

Apache 2.....	1
Resumen.....	3
Introducción.....	4
Paso 1: Instalar Apache.....	4
Paso 2: Ajustar el firewall.....	5
Paso 3: Comprobar su servidor web.....	6
Paso 4: Administrar el proceso de Apache.....	8

Resumen

En este documento se explicará detalladamente cómo instalar y administrar el servidor web Apache2 en un sistema operativo basado en Linux. Se comenzará con la instalación de Apache y cómo ajustar el firewall para permitir el tráfico hacia el servidor web. Se enseñará luego cómo verificar el estado del servidor y cómo administrar el proceso de Apache utilizando varios comandos.

También se proporcionará una guía paso a paso sobre cómo crear un nuevo sitio web en Apache, que incluirá la configuración de los archivos necesarios y cómo activar el sitio en Apache. Se demostrará cómo verificar la configuración de Apache y cómo solucionar posibles problemas.

Finalmente, se mostrará el resultado final de la configuración y se proporcionará una conclusión sobre el proceso y cómo administrar eficazmente el servidor web Apache en el futuro. Este documento será una guía completa para cualquier persona que busque entender y administrar Apache2 en un sistema operativo basado en Linux.

Introducción

Este proyecto es realizado en la Academia Ágil, un centro educativo conocido por su enfoque innovador y práctico en la enseñanza. Manuel Querol Aicart, estuvo presente durante todas las fases del proyecto, participó activamente en su desarrollo y ejecución.

Apache es un servidor web de código abierto esencial para el desarrollo de la World Wide Web. Fue creado y es cuidado por una comunidad global de voluntarios conocida como la Apache Software Foundation. Lanzado en 1995, Apache se convirtió pronto en el servidor web más popular en Internet por su eficiente rendimiento, arquitectura modular y amplio rango de características. Sirve contenido estático y dinámico en la web y ha jugado un papel significativo en el desarrollo de la infraestructura web moderna. Su versatilidad es notable, utilizándose en una variedad de aplicaciones web, desde sitios personales hasta e-commerce y servicios web.

Algunas alternativas al servicio Apache:

- **Nginx:** Es un servidor web de código abierto que también se utiliza como balanceador de carga, proxy inverso y servidor de correo electrónico. Es conocido por su alto rendimiento y estabilidad, así como por su bajo consumo de recursos.
- **Microsoft IIS:** Es un servidor web creado por Microsoft para sus sistemas operativos Windows. Es conocido por su integración con otras ofertas de Microsoft, como el .NET framework.
- **LiteSpeed:** Es un servidor web comercial que es compatible con las configuraciones de Apache, pero ofrece un rendimiento superior y una mayor eficiencia en el uso de los recursos.

Paso 1: Instalar Apache

Apache está disponible en los repositorios de software predeterminados de Ubuntu, lo que permite instalarlo con las herramientas convencionales de administración de paquetes.

Comencemos actualizando el índice de paquetes locales para que reflejen los últimos cambios anteriores:

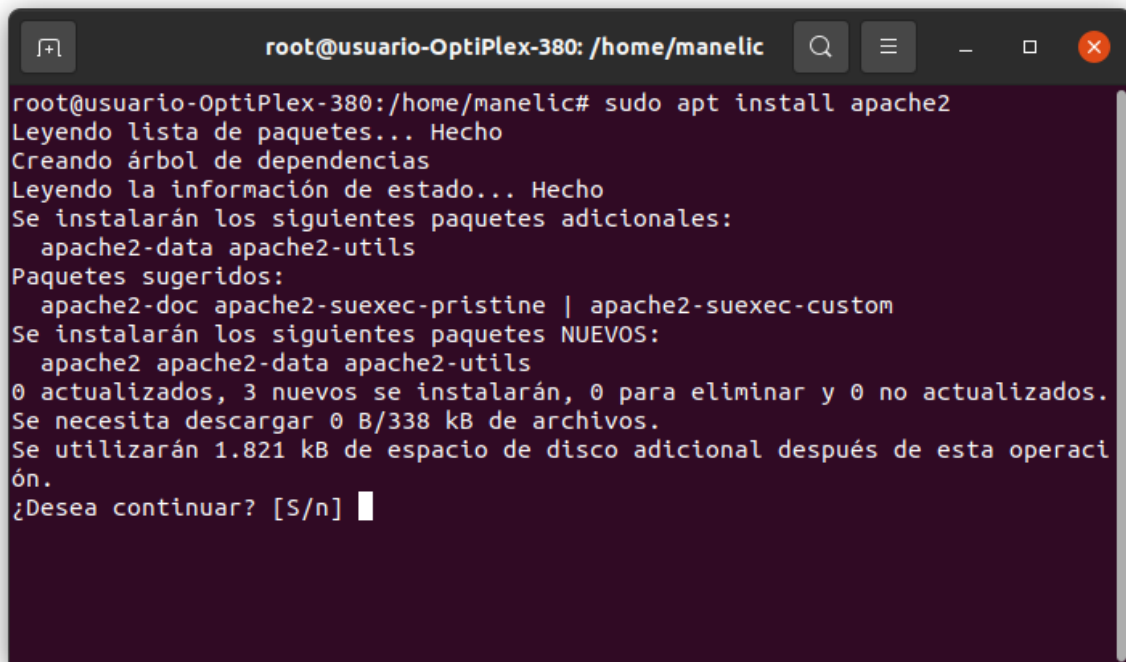


```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu focal InRelease
Obj:3 http://dl.google.com/linux/chrome/deb stable InRelease
Obj:4 https://download.docker.com/linux/ubuntu focal InRelease
Des:5 https://deb.nodesource.com/node_12.x focal InRelease [4.583 B]
Obj:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Obj:7 https://packages.microsoft.com/repos/vscode stable InRelease
Descargados 4.583 B en 1s (3.471 B/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
W: El objetivo Packages (stable/binary-amd64/Packages) está configurado varias veces en /etc/apt/sources.list:54 y /etc/apt/sources.list.d/docker.list:1
W: El objetivo Packages (stable/binary-all/Packages) está configurado varias veces en /etc/apt/sources.list:54 y /etc/apt/sources.list.d/docker.list:1
W: El objetivo Translations (stable/i18n/Translation-es_ES) está configurado varias veces en /etc/apt/sources.list:54 y /etc/apt/sources.list.d/docker.list:1
```

- **"sudo apt-get update"** es un comando muy importante en los sistemas operativos basados en Linux, como Ubuntu. Este comando se utiliza para actualizar la lista de paquetes disponibles en los repositorios configurados en el sistema. "sudo" permite ejecutar el comando como superusuario, "apt-get" es la herramienta de gestión de paquetes, y "update" indica que queremos actualizar la lista de paquetes. Es importante recordar que este comando no actualiza los paquetes en sí, sino solamente las listas de paquetes.

Para actualizar los paquetes ya instalados a las versiones más recientes, se debe usar el comando "sudo apt-get upgrade".

A continuación, instale el paquete apache2:



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-data apache2-utils
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-data apache2-utils
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 0 B/338 kB de archivos.
Se utilizarán 1.821 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

- **"sudo apt install apache2"** es un comando utilizado en sistemas operativos basados en Linux para instalar Apache2, un servidor web muy popular. "sudo" permite ejecutar el comando como superusuario, "apt" es la herramienta de gestión de paquetes, "install" indica que queremos instalar un paquete, y "apache2" es el paquete que queremos instalar. Este comando instalará Apache2 y todas las dependencias necesarias.

Una vez confirmada la instalación, apt instalará Apache y todas las dependencias necesarias.

Paso 2: Ajustar el firewall

Antes de probar Apache, es necesario modificar los ajustes de firewall para permitir el acceso externo a los puertos web predeterminados. Suponiendo que siguió las instrucciones de los requisitos previos, debería tener un firewall UFW configurado para que restrinja el acceso a su servidor.

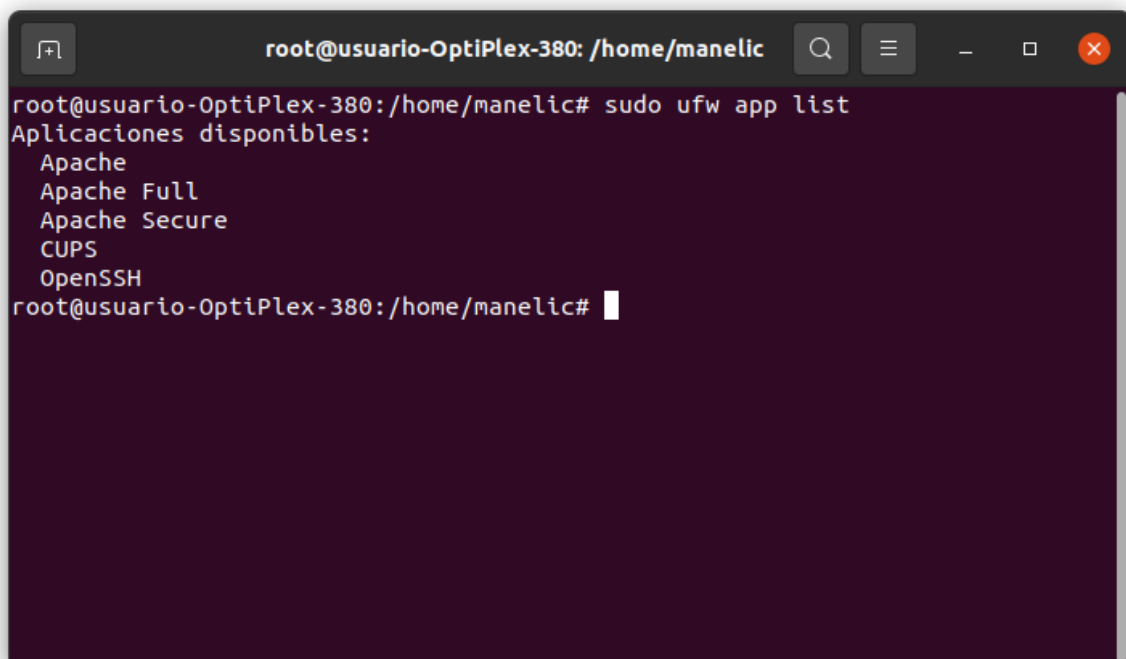
Durante la instalación, Apache se registra con UFW para proporcionar algunos perfiles de aplicación que pueden utilizarse para habilitar o deshabilitar el acceso a Apache a través del firewall.

Enumere los perfiles de aplicación ufw escribiendo lo siguiente:

- **"sudo ufw app list"** es un comando utilizado en sistemas operativos basados en Linux para listar todas las aplicaciones registradas en UFW (Uncomplicated Firewall). "sudo" permite ejecutar el comando como superusuario, "ufw" es el firewall sin complicaciones, "app" se refiere a las aplicaciones y "list" indica que queremos listar todas las aplicaciones registradas. Este comando es útil para ver qué aplicaciones tienen perfiles de firewall predefinidos.

Obtendrá una lista de los perfiles de aplicación:

Output



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
root@usuario-OptiPlex-380:/home/manelic#
```

Como lo indica el resultado, hay tres perfiles disponibles para Apache:

Apache: este perfil abre solo el puerto 80 (tráfico web normal no cifrado)

Apache Full: este perfil abre el puerto 80 (tráfico web normal no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)

Apache Secure: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

Se recomienda habilitar el perfil más restrictivo, que de todos modos permitirá el tráfico que se configuró. Solo deberemos permitir el tráfico en el puerto 80:

- **"ufw allow 'Apache'"** es un comando utilizado en sistemas operativos basados en Linux para permitir todo el tráfico hacia Apache a través del firewall UFW (Uncomplicated Firewall). "sudo" permite ejecutar el comando como superusuario, "ufw" es el firewall sin complicaciones, "allow" indica que queremos permitir el tráfico y "'Apache'" es el perfil de la aplicación para la cual queremos permitir el tráfico. Este comando es esencial para asegurarse de que el servidor web Apache sea accesible desde otras redes.



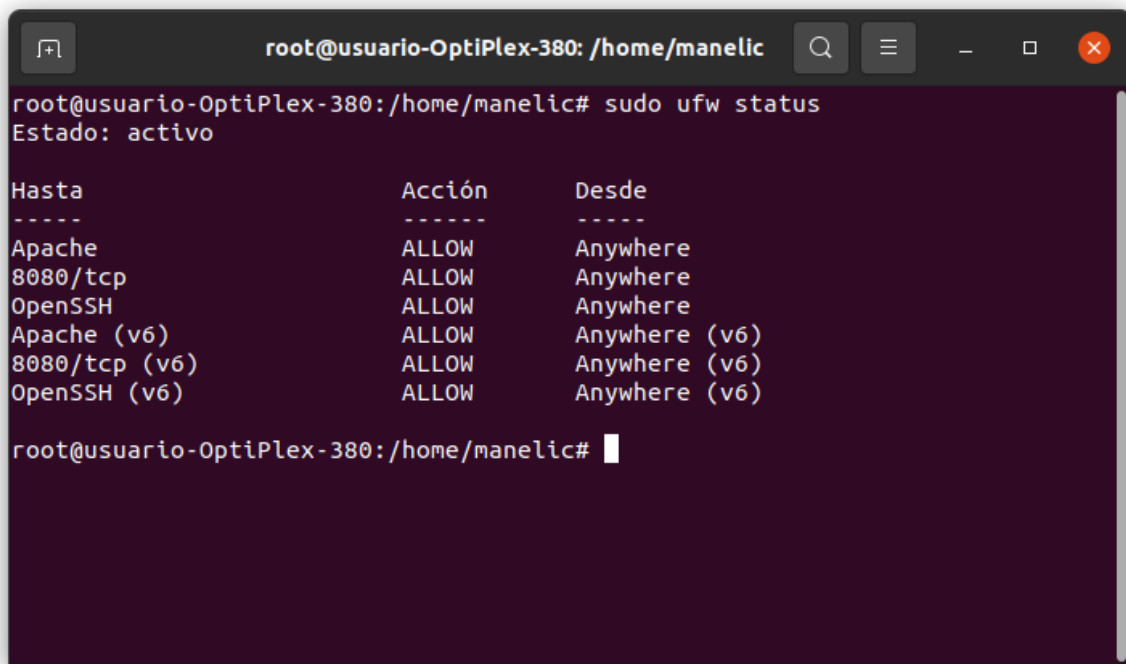
```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo ufw allow 'Apache'
Omitiendo adición de regla ya existente
Omitiendo adición de regla ya existente (v6)
root@usuario-OptiPlex-380:/home/manelic#
```

Puede verificar el cambio escribiendo lo siguiente:

- **"ufw status"** es un comando utilizado en sistemas operativos basados en Linux para verificar el estado del firewall UFW (Uncomplicated Firewall). Este comando muestra si el firewall está activo y qué reglas de tráfico están actualmente en vigor. Es útil para comprobar rápidamente las configuraciones de seguridad de la red.

El resultado proporcionará una lista del tráfico de HTTP que se permite:

Output



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo ufw status
Estado: activo

Hasta      Acción      Desde
-----
Apache      ALLOW       Anywhere
8080/tcp    ALLOW       Anywhere
OpenSSH     ALLOW       Anywhere
Apache (v6) ALLOW       Anywhere (v6)
8080/tcp (v6) ALLOW       Anywhere (v6)
OpenSSH (v6) ALLOW       Anywhere (v6)

root@usuario-OptiPlex-380:/home/manelic#
```

Como lo indica el resultado, el perfil se activó para permitir el acceso al servidor web Apache.

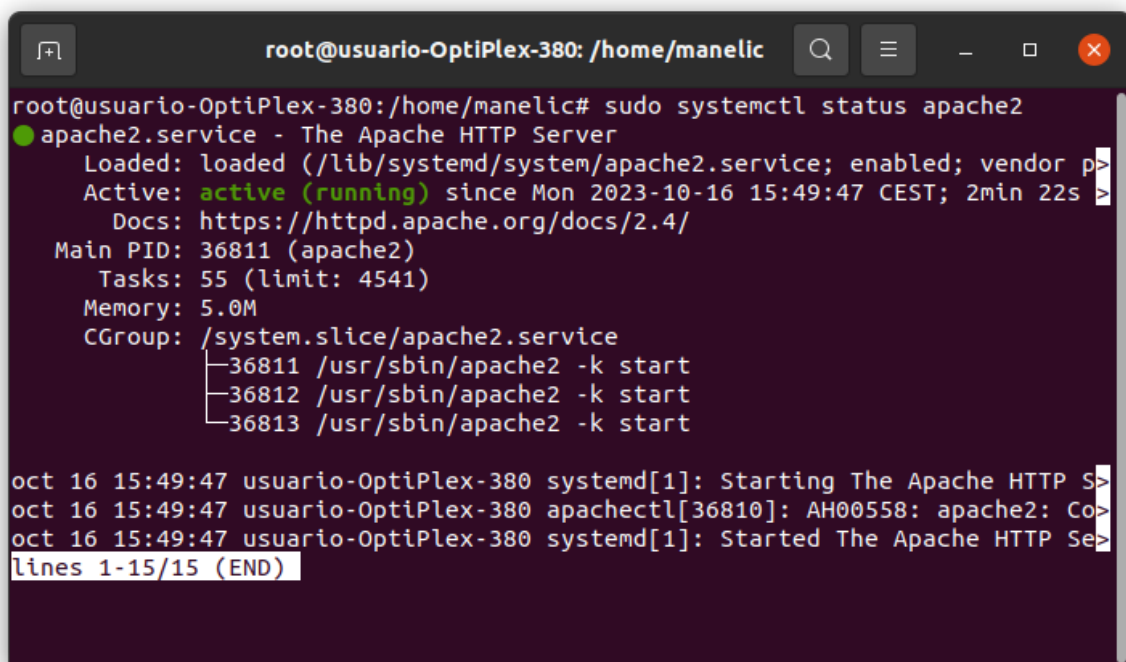
Paso 3: Comprobar su servidor web

Al final del proceso de instalación, Ubuntu 20.04 inicia Apache. El servidor web ya debería estar activo.

Realice una verificación con el sistema `init systemd` para saber si se encuentra en ejecución el servicio escribiendo lo siguiente:

- **"systemctl status apache2"** es un comando utilizado en sistemas operativos basados en Linux para revisar el estado del servicio Apache2. Este comando proporciona información sobre si el servicio está en ejecución, detenido o inactivo, así como detalles sobre la configuración y el rendimiento del servicio. Es muy útil para la resolución de problemas y el monitoreo del servidor web Apache.

Output



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p>
   Active: active (running) since Mon 2023-10-16 15:49:47 CEST; 2min 22s >
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 36811 (apache2)
     Tasks: 55 (limit: 4541)
    Memory: 5.0M
     CGroup: /system.slice/apache2.service
             └─36811 /usr/sbin/apache2 -k start
               └─36812 /usr/sbin/apache2 -k start
                 └─36813 /usr/sbin/apache2 -k start

oct 16 15:49:47 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 15:49:47 usuario-OptiPlex-380 apachectl[36810]: AH00558: apache2: Co>
oct 16 15:49:47 usuario-OptiPlex-380 systemd[1]: Started The Apache HTTP Se>
lines 1-15/15 (END)
```

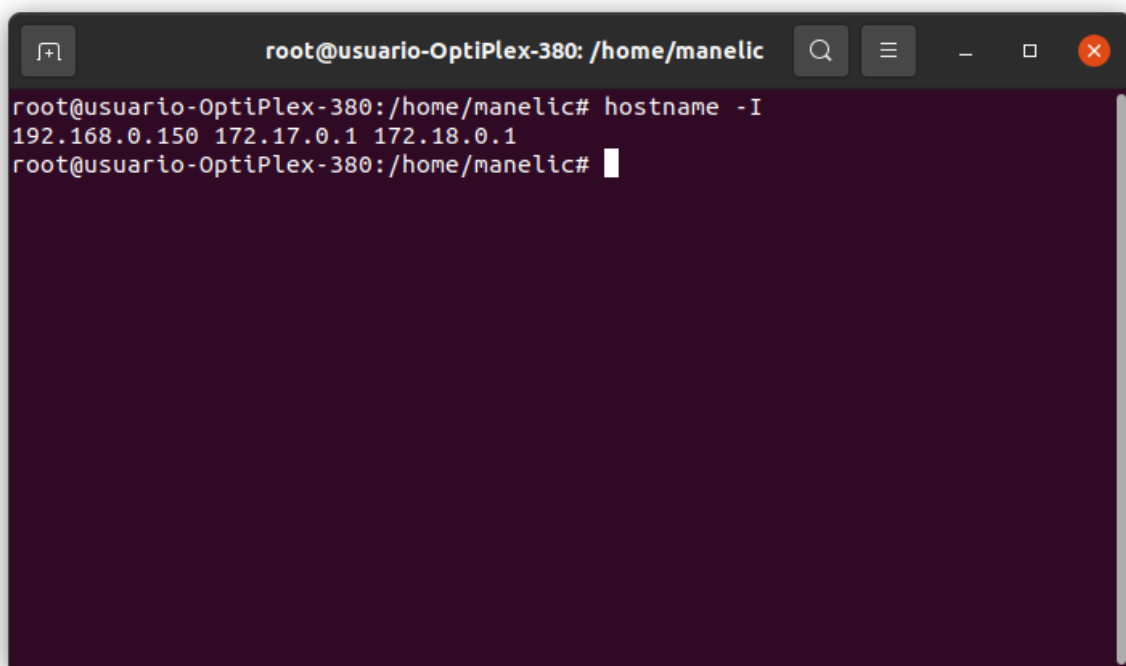
Como lo confirma este resultado, el servicio se inició correctamente. Sin embargo, la mejor forma de comprobarlo es solicitar una página de Apache.

Puede acceder a la página de destino predeterminada de Apache para confirmar que el software funcione correctamente mediante su dirección IP: Si no conoce la dirección IP de su servidor, puede obtenerla de varias formas desde la línea de comandos.

Intente escribir esto en la línea de comandos:

- **"hostname -I"** es un comando utilizado en sistemas operativos basados en Linux para obtener la dirección IP del sistema en la red. El comando "hostname" se utiliza para mostrar o establecer el nombre del sistema, pero con la opción "-I", muestra todas las direcciones de red del sistema. Este comando es muy útil cuando necesitas conocer la dirección IP del sistema para configurar conexiones de red o para tareas de resolución de problemas de red.

Obtendrá algunas direcciones separadas por espacios. Puede probar cada uno en el navegador web para determinar si funcionan.

A terminal window with a dark purple background. The title bar at the top shows 'root@usuario-OptiPlex-380: /home/manelic' and standard window controls. The terminal text shows the command 'hostname -I' being executed, resulting in the output '192.168.0.150 172.17.0.1 172.18.0.1'. The prompt 'root@usuario-OptiPlex-380:/home/manelic#' is visible before and after the command.

```
root@usuario-OptiPlex-380:/home/manelic# hostname -I
192.168.0.150 172.17.0.1 172.18.0.1
root@usuario-OptiPlex-380:/home/manelic#
```

Otra opción es utilizar la herramienta Icanhazip, que debería proporcionar su dirección IP pública como aparece en otra ubicación en Internet:

- **"curl -4 icanhazip.com"** es un comando utilizado en sistemas operativos basados en Linux para obtener la dirección IP pública de su sistema. "curl" es una herramienta de línea de comandos utilizada para transferir datos desde o hacia un servidor. La opción "-4" obliga a curl a resolver los nombres de host a direcciones IPv4. "icanhazip.com" es un servicio web que simplemente responde con su dirección IP pública. Este comando es útil cuando necesitas conocer tu dirección IP pública para la configuración de la red o la resolución de problemas.

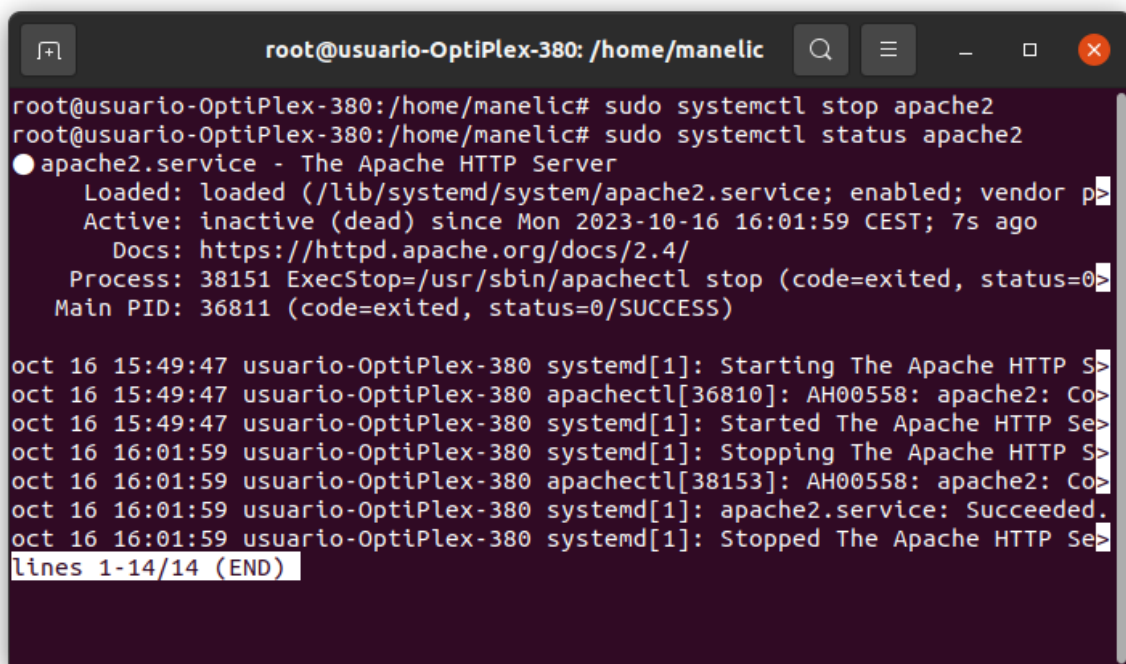
Cuando tenga la dirección IP de su servidor, introdúcela en la barra de direcciones de su navegador

Administrar el proceso de Apache

Ahora que el servidor web está listo y en funcionamiento, estos son algunos comandos de administración básicos con systemctl.

Para detener su servidor web, escriba lo siguiente:

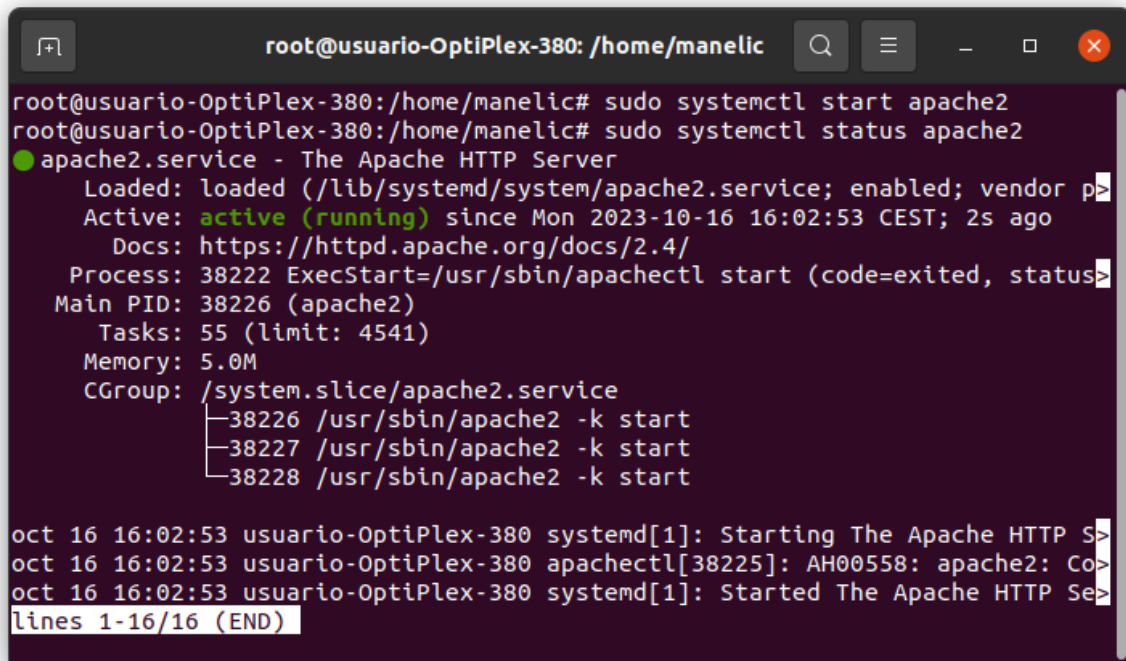
- "systemctl stop apache2" es un comando utilizado en sistemas operativos basados en Linux para detener el servicio Apache2. Este comando es muy útil cuando necesitas detener el servidor web Apache por razones de mantenimiento, actualizaciones o para resolver problemas. Al usar este comando, el servicio Apache2 se detendrá y no aceptará nuevas conexiones hasta que se reinicie con el comando "systemctl start apache2". Es importante tener en cuenta que detener el servicio puede afectar a los usuarios que estén accediendo al servidor web en ese momento.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl stop apache2
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p>
   Active: inactive (dead) since Mon 2023-10-16 16:01:59 CEST; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 38151 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0>
   Main PID: 36811 (code=exited, status=0/SUCCESS)

oct 16 15:49:47 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 15:49:47 usuario-OptiPlex-380 apachectl[36810]: AH00558: apache2: Co>
oct 16 15:49:47 usuario-OptiPlex-380 systemd[1]: Started The Apache HTTP Se>
oct 16 16:01:59 usuario-OptiPlex-380 systemd[1]: Stopping The Apache HTTP S>
oct 16 16:01:59 usuario-OptiPlex-380 apachectl[38153]: AH00558: apache2: Co>
oct 16 16:01:59 usuario-OptiPlex-380 systemd[1]: apache2.service: Succeeded.
oct 16 16:01:59 usuario-OptiPlex-380 systemd[1]: Stopped The Apache HTTP Se>
lines 1-14/14 (END)
```

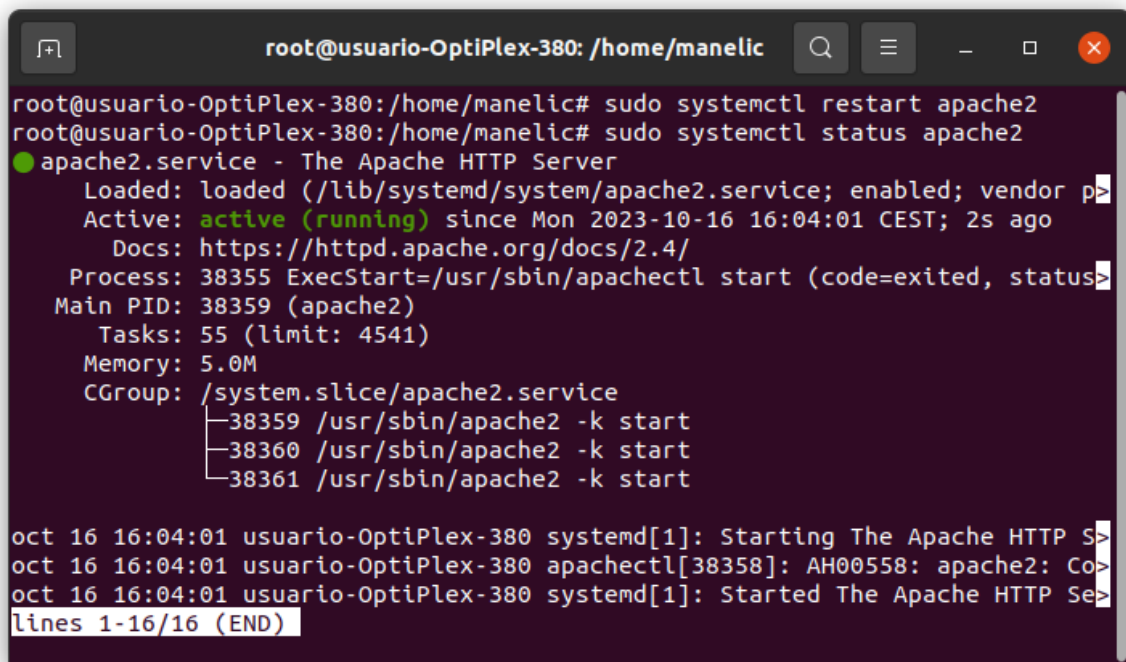
- "systemctl start apache2" es un comando utilizado en sistemas operativos basados en Linux para iniciar el servicio Apache2. Este comando es extremadamente útil cuando necesitas poner en marcha el servidor web Apache después de haberlo detenido por razones de mantenimiento, actualizaciones o para resolver problemas. Al ejecutar este comando, el servicio Apache2 se inicia y comienza a aceptar nuevas conexiones. Es importante recordar que el servidor web solo estará disponible para los usuarios una vez que se haya iniciado con este comando.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl start apache2
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p>
   Active: active (running) since Mon 2023-10-16 16:02:53 CEST; 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 38222 ExecStart=/usr/sbin/apachectl start (code=exited, status>
   Main PID: 38226 (apache2)
    Tasks: 55 (limit: 4541)
   Memory: 5.0M
   CGroup: /system.slice/apache2.service
           └─38226 /usr/sbin/apache2 -k start
             └─38227 /usr/sbin/apache2 -k start
               └─38228 /usr/sbin/apache2 -k start

oct 16 16:02:53 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 16:02:53 usuario-OptiPlex-380 apachectl[38225]: AH00558: apache2: Co>
oct 16 16:02:53 usuario-OptiPlex-380 systemd[1]: Started The Apache HTTP Se>
lines 1-16/16 (END)
```

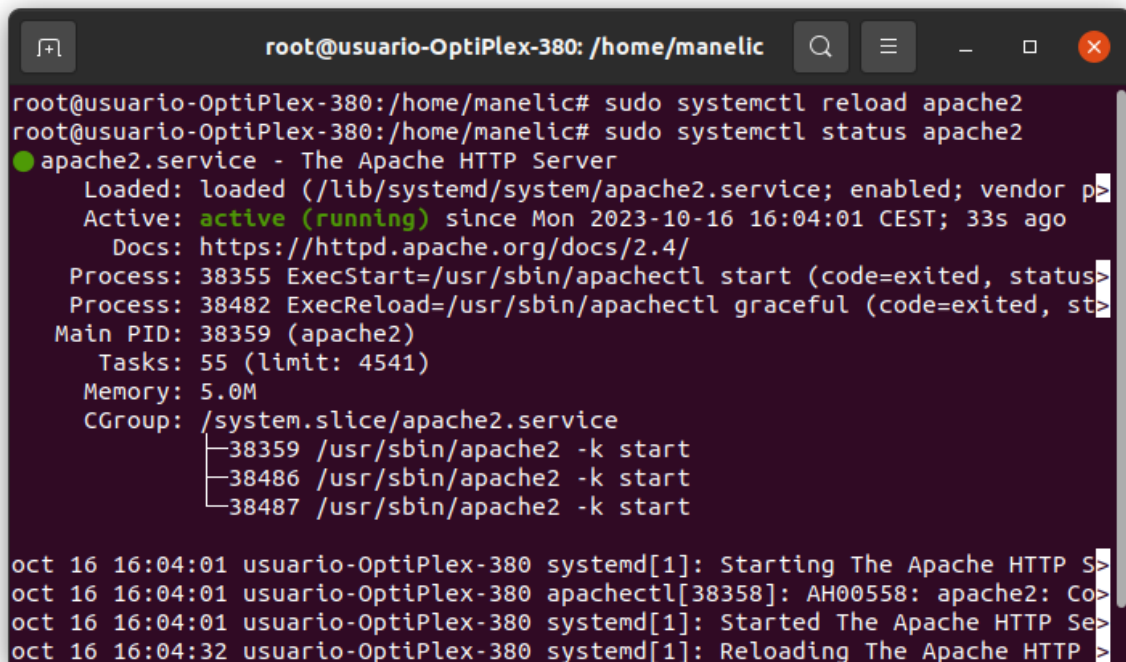
- "systemctl restart apache2" es un comando utilizado en sistemas operativos basados en Linux para reiniciar el servicio Apache2. Este comando es especialmente útil cuando se han realizado cambios en la configuración del servidor web Apache y estos necesitan ser aplicados. Al ejecutar este comando, el servicio Apache2 se detiene y luego se inicia de nuevo automáticamente. Este comando asegura que cualquier cambio en la configuración sea implementado y que el servidor esté funcionando con la configuración más actualizada. Es importante recordar que durante el reinicio, el servidor web puede estar temporalmente inaccesible para los usuarios.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl restart apache2
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p>
   Active: active (running) since Mon 2023-10-16 16:04:01 CEST; 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 38355 ExecStart=/usr/sbin/apachectl start (code=exited, status>
  Main PID: 38359 (apache2)
    Tasks: 55 (limit: 4541)
   Memory: 5.0M
    CGroup: /system.slice/apache2.service
            └─38359 /usr/sbin/apache2 -k start
              └─38360 /usr/sbin/apache2 -k start
                └─38361 /usr/sbin/apache2 -k start

oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 16:04:01 usuario-OptiPlex-380 apachectl[38358]: AH00558: apache2: Co>
oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Started The Apache HTTP Se>
lines 1-16/16 (END)
```

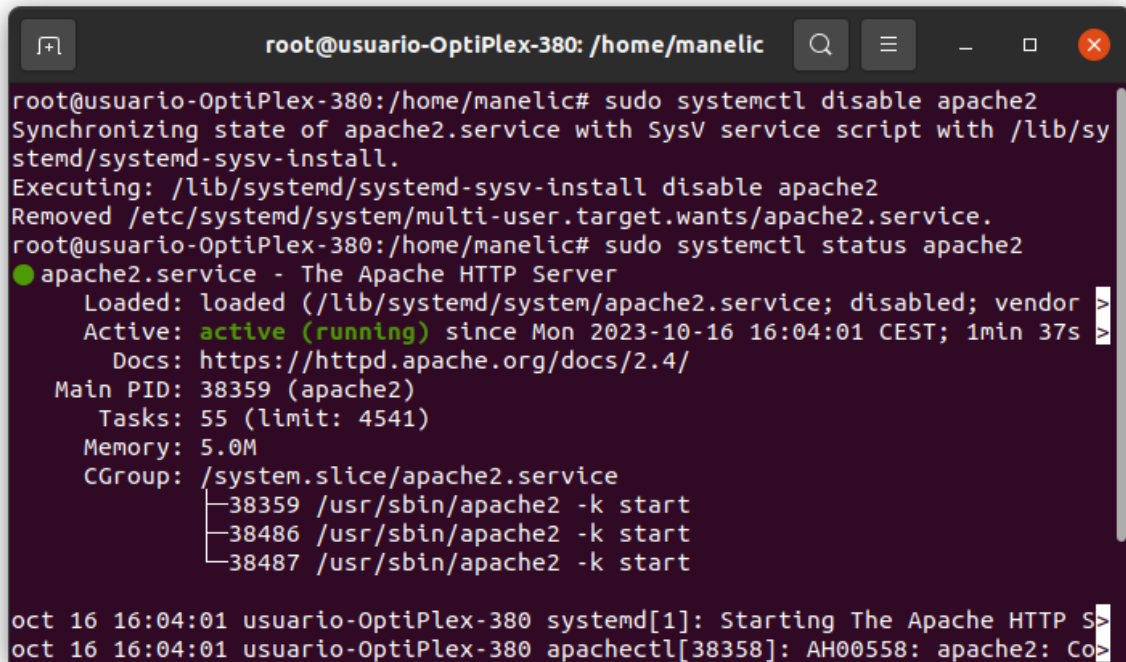
- "systemctl reload apache2" es un comando utilizado en sistemas operativos basados en Linux para recargar la configuración del servicio Apache2 sin interrumpir las conexiones existentes. Este comando es especialmente útil cuando se han realizado cambios en la configuración del servidor web Apache que no requieren un reinicio completo del servicio. Al ejecutar este comando, Apache2 recarga su configuración, aplicando los cambios realizados, sin la necesidad de detener y reiniciar el servicio. Esto minimiza el tiempo de inactividad y permite que las conexiones de los usuarios actuales continúen sin interrupción.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl reload apache2
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p
   Active: active (running) since Mon 2023-10-16 16:04:01 CEST; 33s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 38355 ExecStart=/usr/sbin/apachectl start (code=exited, status>
   Process: 38482 ExecReload=/usr/sbin/apachectl graceful (code=exited, st>
   Main PID: 38359 (apache2)
     Tasks: 55 (limit: 4541)
    Memory: 5.0M
     CGroup: /system.slice/apache2.service
            └─38359 /usr/sbin/apache2 -k start
              └─38486 /usr/sbin/apache2 -k start
                └─38487 /usr/sbin/apache2 -k start

oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 16:04:01 usuario-OptiPlex-380 apachectl[38358]: AH00558: apache2: Co>
oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Started The Apache HTTP Se>
oct 16 16:04:32 usuario-OptiPlex-380 systemd[1]: Reloading The Apache HTTP >
```

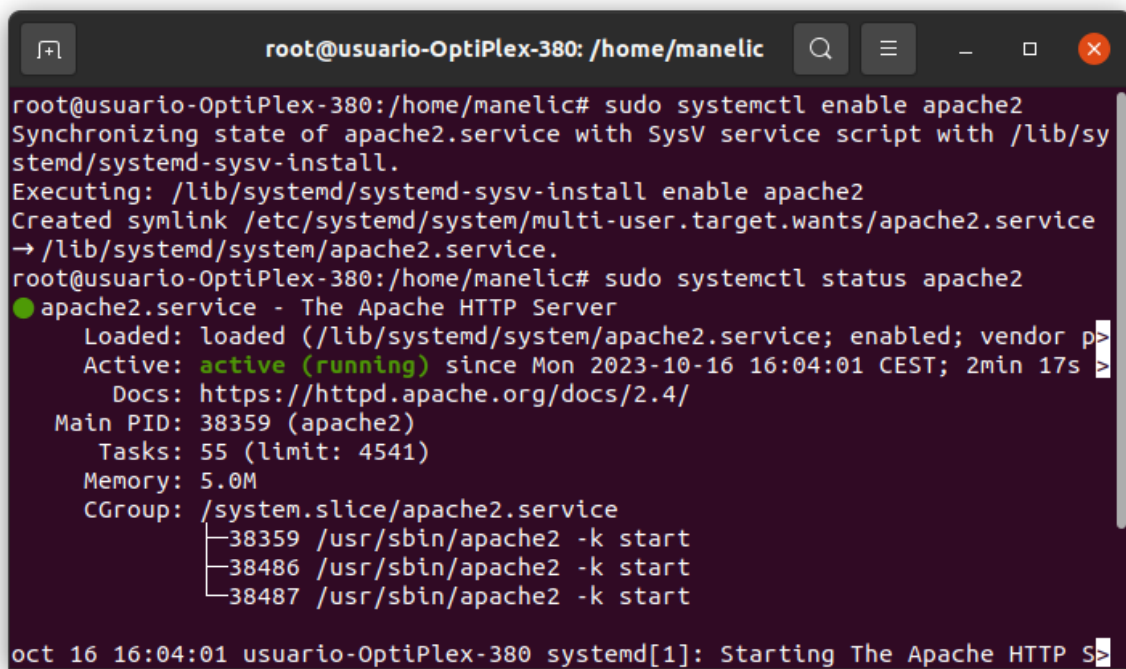
- "systemctl disable apache2" es un comando utilizado en sistemas operativos basados en Linux para desactivar el inicio automático del servicio Apache2. Este comando es particularmente útil cuando no quieres que el servidor web Apache se inicie automáticamente al arrancar el sistema. Al ejecutar este comando, el servicio Apache2 seguirá disponible, pero deberás iniciar manualmente con el comando "systemctl start apache2" cada vez que reinicies el sistema. Este comando es esencial para administrar eficazmente los recursos del sistema y mantener el control sobre cuándo y cómo se ejecutan los servicios.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /lib/sy
stemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor >
   Active: active (running) since Mon 2023-10-16 16:04:01 CEST; 1min 37s >
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 38359 (apache2)
      Tasks: 55 (limit: 4541)
     Memory: 5.0M
    CGroup: /system.slice/apache2.service
            └─38359 /usr/sbin/apache2 -k start
              └─38486 /usr/sbin/apache2 -k start
                └─38487 /usr/sbin/apache2 -k start

oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
oct 16 16:04:01 usuario-OptiPlex-380 apachectl[38358]: AH00558: apache2: Co>
```


- "systemctl enable apache2" es un comando utilizado en sistemas operativos basados en Linux para activar el inicio automático del servicio Apache2. Este comando es especialmente útil cuando quieres que el servidor web Apache se inicie automáticamente cada vez que arranque el sistema. Al ejecutar este comando, el sistema estará configurado para iniciar el servicio Apache2 en cada reinicio. Esto garantiza que el servidor web esté siempre disponible para los usuarios tan pronto como el sistema esté en marcha, sin necesidad de intervención manual. Sin embargo, es importante tener en cuenta que aunque este comando configure el servicio para iniciarse automáticamente, no inicia el servicio inmediatamente. Para iniciar el servicio después de habilitarlo, se debe utilizar el comando "systemctl start apache2".



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/sy
stemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service
→ /lib/systemd/system/apache2.service.
root@usuario-OptiPlex-380:/home/manelic# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p>
   Active: active (running) since Mon 2023-10-16 16:04:01 CEST; 2min 17s >
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 38359 (apache2)
      Tasks: 55 (limit: 4541)
     Memory: 5.0M
    CGroup: /system.slice/apache2.service
            └─38359 /usr/sbin/apache2 -k start
              └─38486 /usr/sbin/apache2 -k start
                └─38487 /usr/sbin/apache2 -k start

oct 16 16:04:01 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP S>
```

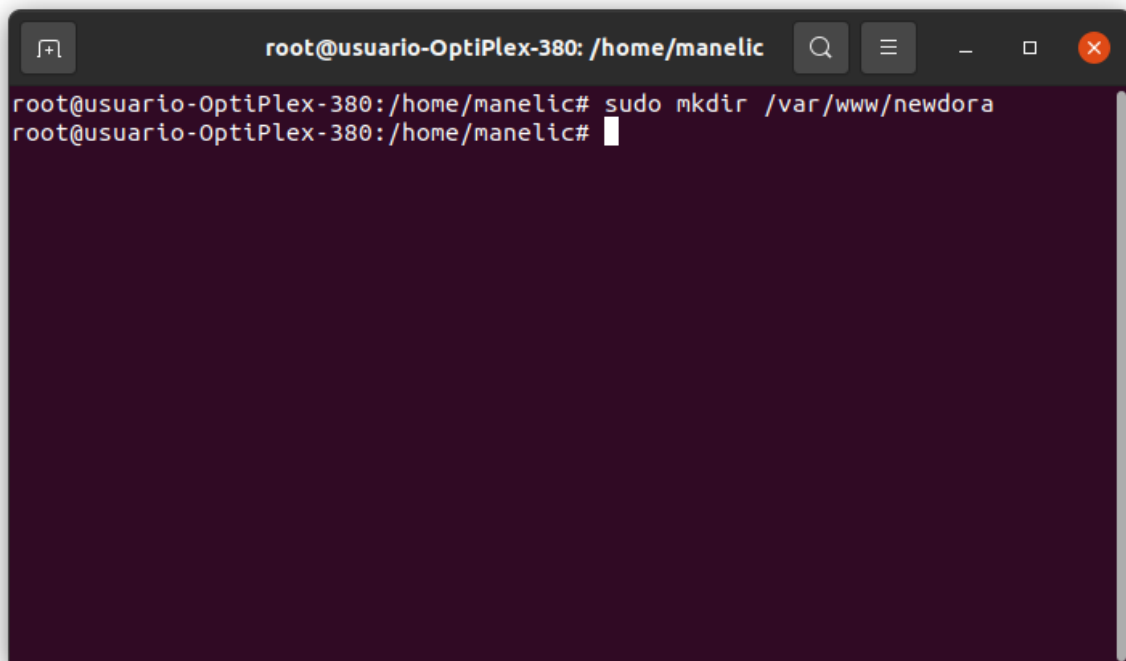

Paso 4: Creando Tu Propio Sitio Web

Por defecto, Apache viene con un sitio básico (el que vimos en el paso anterior) habilitado. Podemos modificar su contenido en `/var/www/html` o configuraciones editando su archivo de Host Virtual que se encuentra en `/etc/apache2/sites-enabled/000-default.conf`.

Podemos modificar cómo Apache maneja las solicitudes entrantes y tener varios sitios funcionando en el mismo servidor editando su archivo de Hosts Virtuales.

Entonces, comencemos creando una carpeta para nuestro nuevo sitio web en `/var/www/` ejecutando:

- `"mkdir /var/www/newdora"` es un comando utilizado en sistemas operativos basados en Linux para crear un nuevo directorio llamado "newdora" en la ubicación `"/var/www/"`. `"mkdir"` es la abreviatura de "make directory", que se utiliza para crear directorios. `"/var/www/"` es una ubicación común para almacenar archivos de sitios web en un servidor web como Apache. Este comando es útil cuando necesitas crear un nuevo espacio para almacenar archivos de un nuevo sitio web o una nueva aplicación web. Recuerda que este comando necesita ser ejecutado con privilegios de superusuario si el directorio `"/var/www/"` es de propiedad del root, para ello se usa `"sudo"` al principio del comando, quedando así: `"sudo mkdir /var/www/newdora"`.

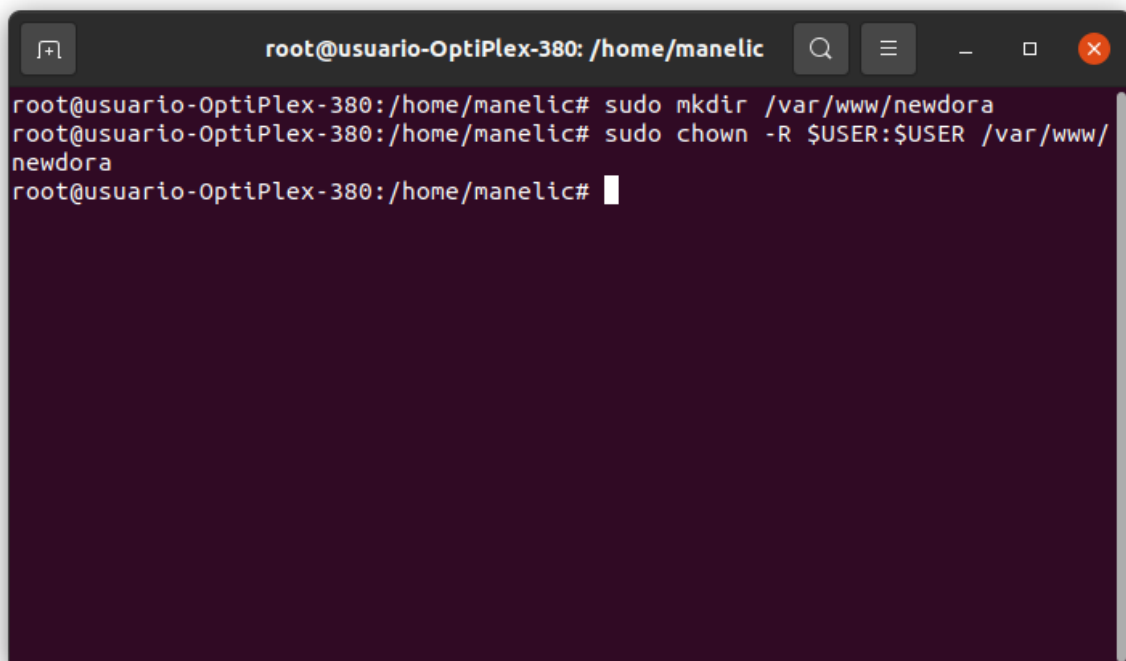
A terminal window with a dark purple background. The title bar at the top shows 'root@usuario-OptiPlex-380: /home/manelic' and standard window controls. The terminal text shows the command 'sudo mkdir /var/www/newdora' being entered and executed, with a cursor on the line below.

```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo mkdir /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic#
```

Lo hemos nombrado newdora aquí, pero cualquier nombre funcionará, siempre y cuando lo marquemos en el archivo de configuración de hosts virtuales más tarde.

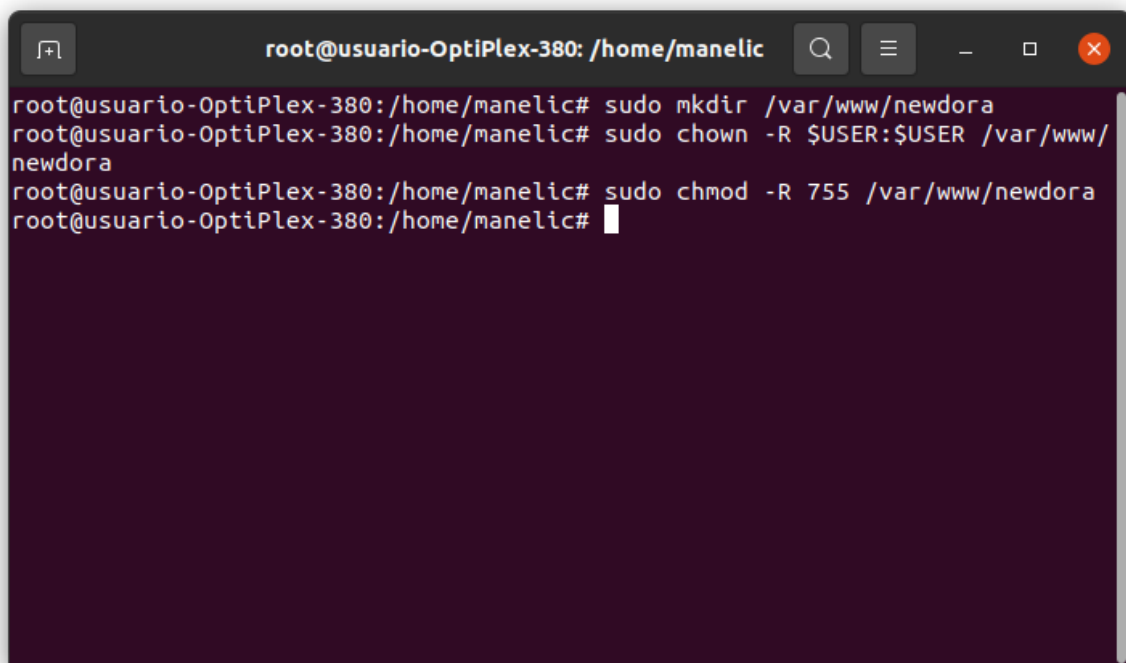
Realizaremos una asignación de propietario y permisos:

- "chown -R \$USER:\$USER /var/www/newdora" es un comando utilizado en sistemas operativos basados en Linux para cambiar la propiedad de un directorio y todos sus subdirectorios. En este caso, el directorio es "/var/www/newdora". "chown" es una abreviatura de "change owner" (cambiar propietario), "-R" indica que la operación debe aplicarse de manera recursiva a todos los subdirectorios y archivos dentro del directorio, y "\$USER:\$USER" indica el nuevo propietario y grupo del directorio. "\$USER" es una variable de entorno que corresponde al usuario actualmente conectado. Este comando es útil para cambiar la propiedad de un directorio cuando necesitas tener permisos de escritura en él, como cuando estás configurando un servidor web.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo mkdir /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic# sudo chown -R $USER:$USER /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic#
```

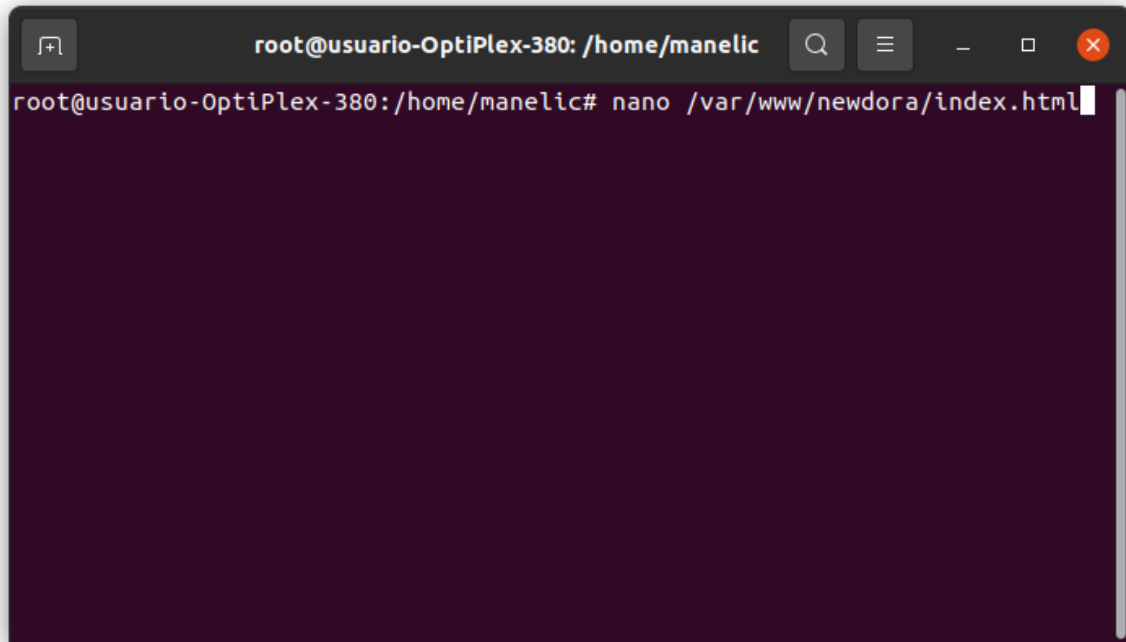
- "chmod -R 755 /var/www/newdora" es un comando utilizado en sistemas operativos basados en Linux para cambiar los permisos de un directorio y todos sus subdirectorios. En este caso, el directorio es "/var/www/newdora". "chmod" es una abreviatura de "change mode" (cambiar modo), "-R" indica que la operación debe aplicarse de manera recursiva a todos los subdirectorios y archivos dentro del directorio, y "755" son los nuevos permisos del directorio. Los permisos "755" permiten al propietario leer, escribir y ejecutar, mientras que otros usuarios sólo pueden leer y ejecutar. Este comando es útil para establecer los permisos de un directorio cuando necesitas asegurar que tienes los permisos apropiados para leer, escribir y ejecutar archivos, como cuando estás configurando un servidor web.



```
root@usuario-OptiPlex-380: /home/manelic
root@usuario-OptiPlex-380:/home/manelic# sudo mkdir /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic# sudo chown -R $USER:$USER /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic# sudo chmod -R 755 /var/www/newdora
root@usuario-OptiPlex-380:/home/manelic#
```

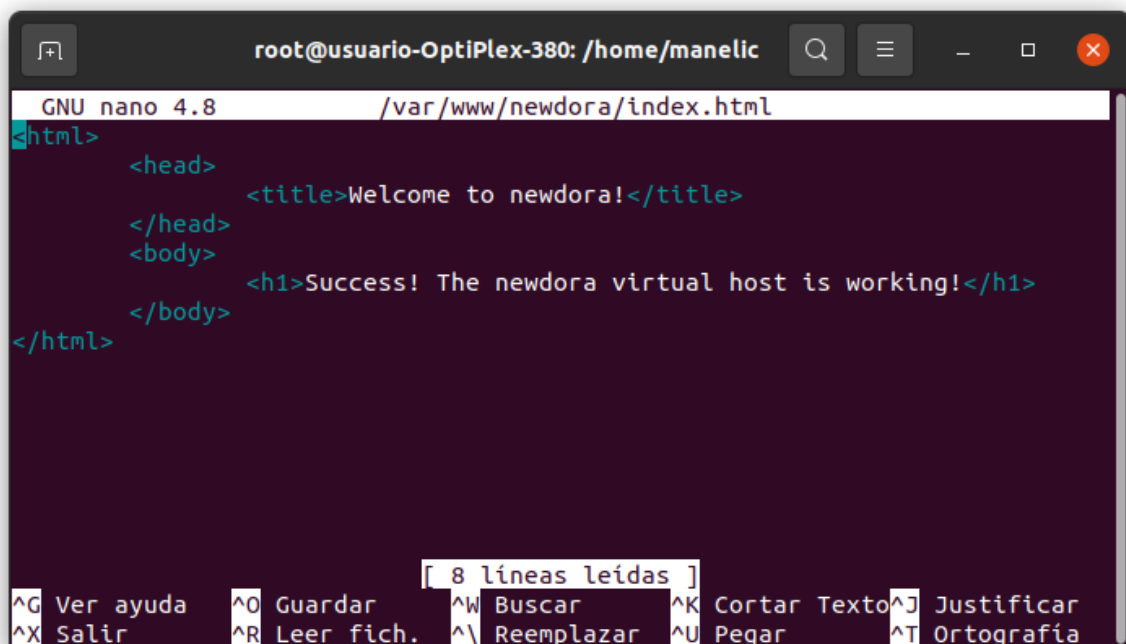
Ahora que hemos creado un directorio para nuestro sitio, coloquemos un archivo HTML en él. Vamos a nuestro directorio recién creado y creamos uno escribiendo:

- "nano /var/www/newdora/index.html" Este comando es útil cuando necesitas editar archivos de texto o código, como cuando estás creando o modificando una página web en un servidor.



A terminal window titled 'root@usuario-OptiPlex-380: /home/manelic'. The command 'nano /var/www/newdora/index.html' has been entered at the prompt, and the cursor is at the end of the command.

Escribe el siguiente código en el archivo index.html:



A terminal window titled 'root@usuario-OptiPlex-380: /home/manelic' showing the GNU nano 4.8 editor. The file being edited is '/var/www/newdora/index.html'. The content of the file is as follows:

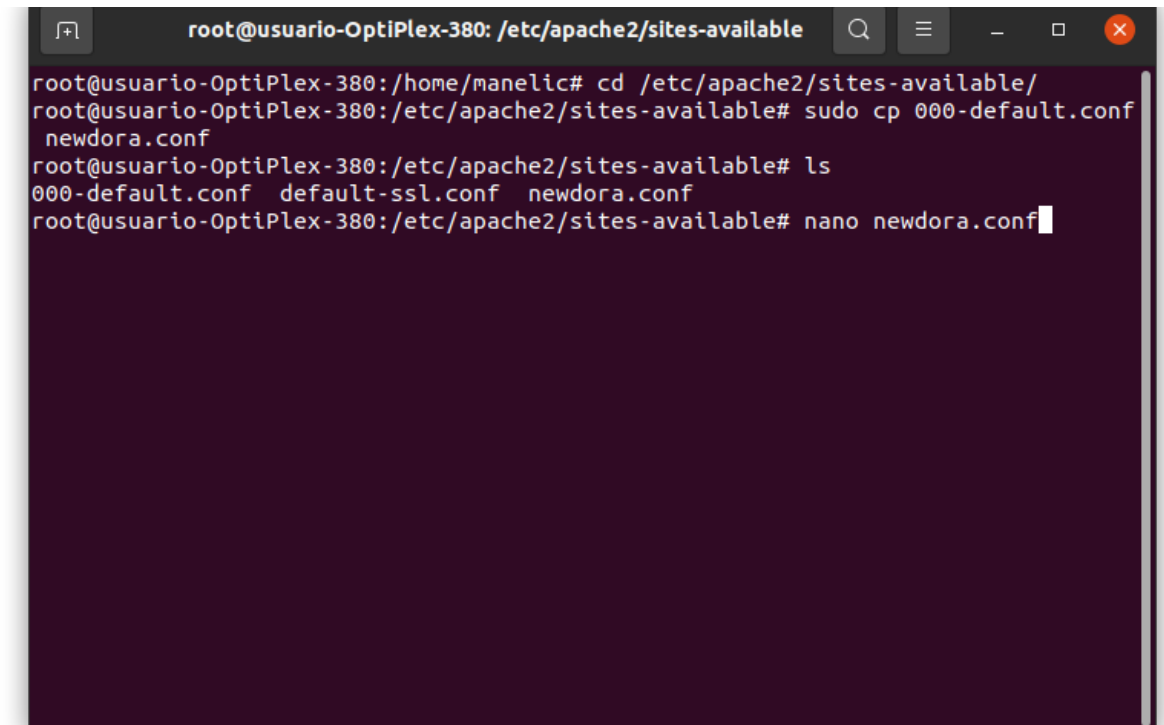
```
<html>
  <head>
    <title>Welcome to newdora!</title>
  </head>
  <body>
    <h1>Success! The newdora virtual host is working!</h1>
  </body>
</html>
```

At the bottom of the window, there is a status bar with the text '[8 líneas leídas]' and a list of keyboard shortcuts: ^G Ver ayuda, ^O Guardar, ^W Buscar, ^K Cortar Texto, ^J Justificar, ^X Salir, ^R Leer fich., ^\ Reemplazar, ^U Pegar, ^T Ortografía.

Paso 5: Creando el Archivo de Configuración

Comenzamos este paso ingresando al directorio de archivos de configuración:

- `cd /etc/apache2/sites-available/`

A terminal window titled 'root@usuario-OptiPlex-380: /etc/apache2/sites-available' with search, menu, and window control icons. The terminal shows the following commands and output:

```
root@usuario-OptiPlex-380:/home/manelic# cd /etc/apache2/sites-available/  
root@usuario-OptiPlex-380:/etc/apache2/sites-available# sudo cp 000-default.conf  
newdora.conf  
root@usuario-OptiPlex-380:/etc/apache2/sites-available# ls  
000-default.conf  default-ssl.conf  newdora.conf  
root@usuario-OptiPlex-380:/etc/apache2/sites-available# nano newdora.conf
```

- `sudo cp 000-default.conf newdora.conf`

Ahora edita el archivo de configuración

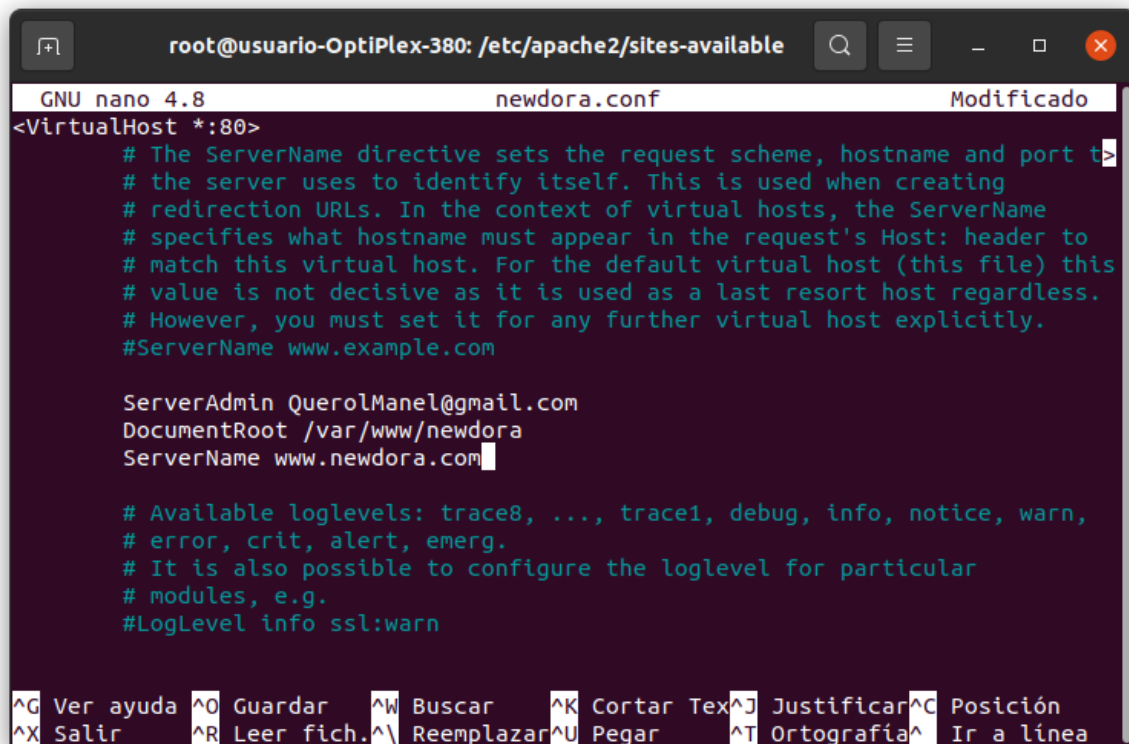
- `sudo nano newdora.conf`

Debemos tener nuestro correo electrónico en ServerAdmin para que los usuarios puedan contactarte en caso de que Apache experimente algún error:

ServerAdmin tunombre@example.com

DocumentRoot /var/www/newdora/

ServerName www.newdora.com



The screenshot shows a terminal window titled 'root@usuario-OptiPlex-380: /etc/apache2/sites-available'. Inside, the nano 4.8 editor is open to a file named 'newdora.conf'. The file content is as follows:

```
<VirtualHost *:80>
    # 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 QuerolManel@gmail.com
    DocumentRoot /var/www/newdora
    ServerName www.newdora.com

    # 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
```

The bottom of the terminal shows the nano editor's command shortcuts: ^G Ver ayuda, ^O Guardar, ^W Buscar, ^K Cortar Text, ^J Justificar, ^C Posición, ^X Salir, ^R Leer fich., ^\ Reemplazar, ^U Pegar, ^T Ortografía, ^_ Ir a línea.

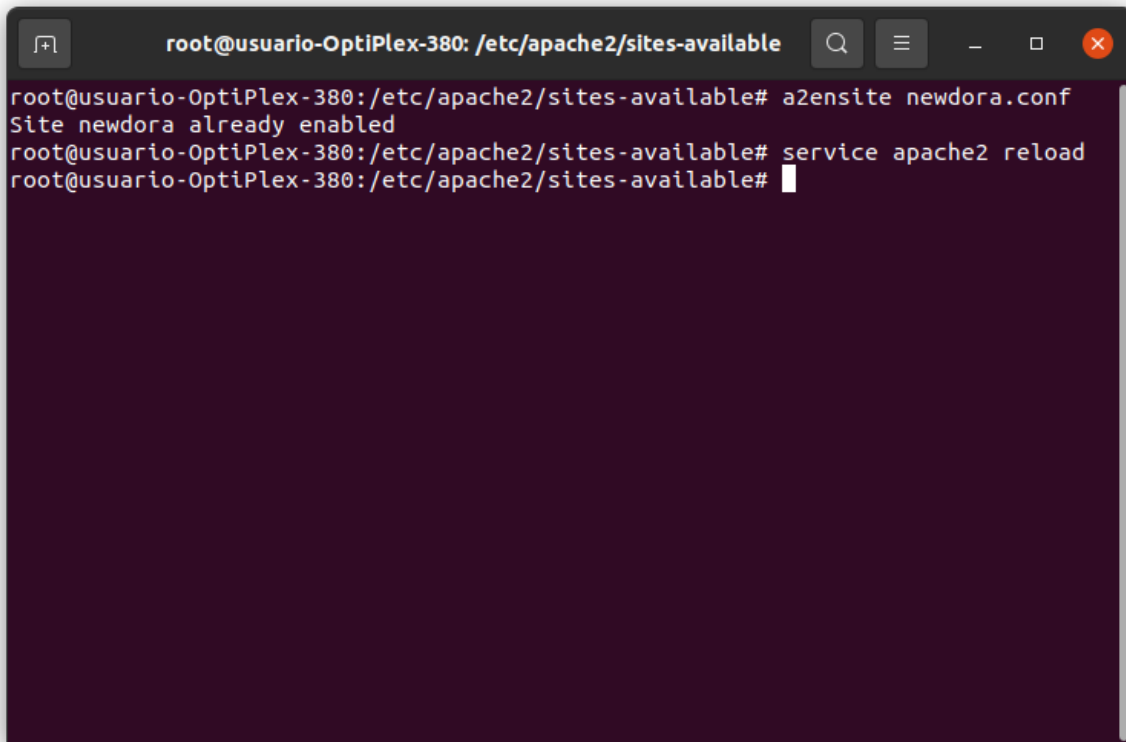
Esto asegura que las personas lleguen al sitio correcto en lugar del predeterminado cuando escriban www.newdora.com.

Paso 6: Activando el archivo VirtualHost

Después de configurar nuestro sitio web, necesitamos activar el archivo de configuración de hosts virtuales para habilitarlo. Lo hacemos ejecutando el siguiente comando en el directorio del archivo de configuración:

- `sudo a2ensite gci.conf`

Deberías ver la siguiente salida:

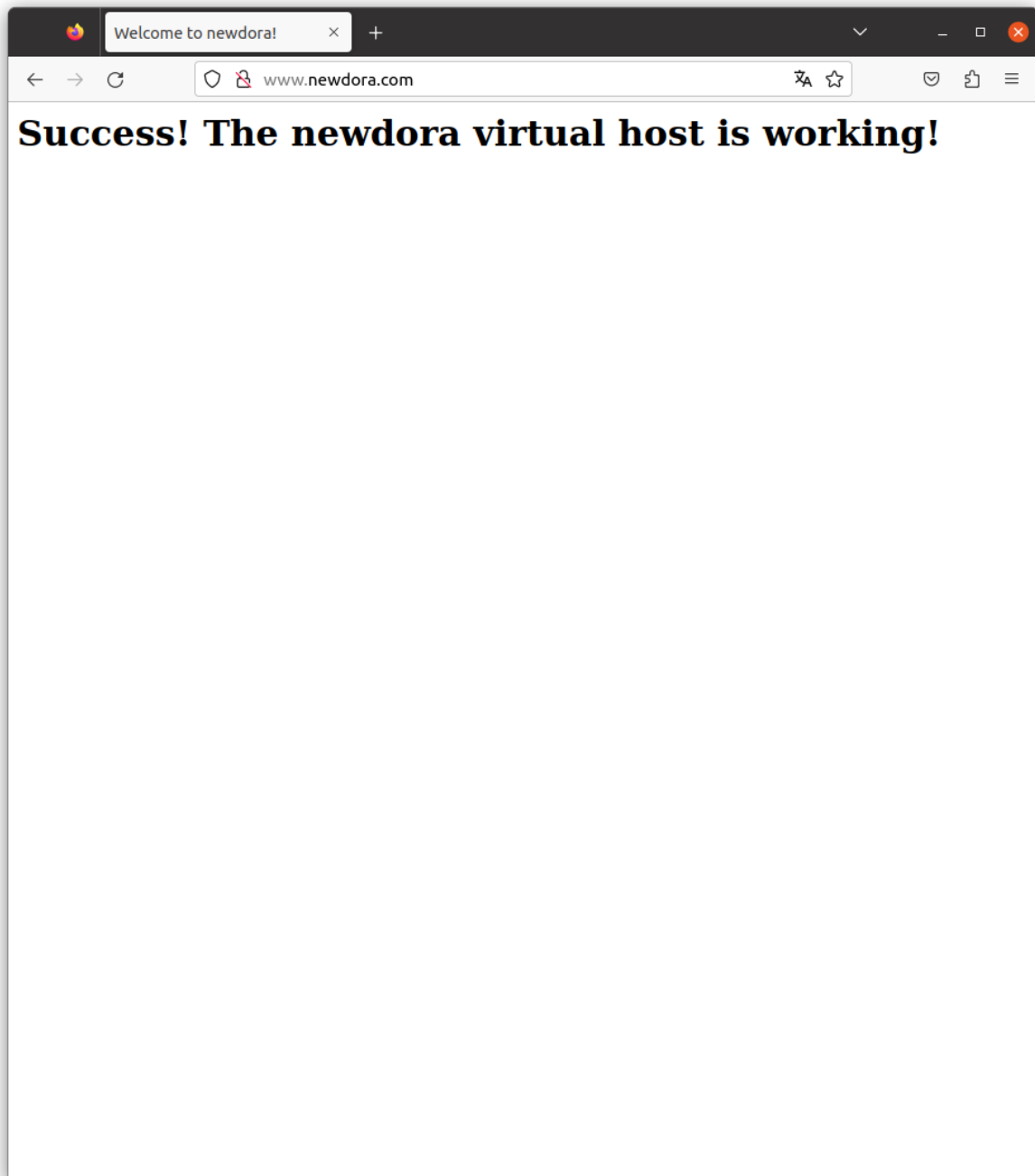
A terminal window with a dark purple background. The title bar shows 'root@usuario-OptiPlex-380: /etc/apache2/sites-available'. The terminal content shows the following commands and output:

```
root@usuario-OptiPlex-380:/etc/apache2/sites-available# a2ensite newdora.conf
Site newdora already enabled
root@usuario-OptiPlex-380:/etc/apache2/sites-available# service apache2 reload
root@usuario-OptiPlex-380:/etc/apache2/sites-available#
```

Para activar la nueva configuración, necesitas ejecutar:

- `service apache2 reload`

Resultado final



Conclusión

En conclusión, trabajar con el servidor Apache ha sido una experiencia enriquecedora y educativa. Su arquitectura modular y la amplia gama de funcionalidades ofrecidas han permitido una gran flexibilidad durante el desarrollo del proyecto. El hecho de que sea un software de código abierto, mantenido por una comunidad global de voluntarios, también significa que hay una gran cantidad de recursos y apoyo disponibles. A pesar de los desafíos que surgieron durante el proceso, la capacidad de Apache para manejar contenido estático y dinámico con eficacia ha demostrado ser invaluable. Sin duda, la experiencia adquirida en este proyecto será de gran utilidad para futuras implementaciones y desarrollos web.

Bibliografía

- <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es?authuser=0>
- <https://ubuntu.com/tutorials/install-and-configure-apache#4-setting-up-the-virtualhost-configuration-file>