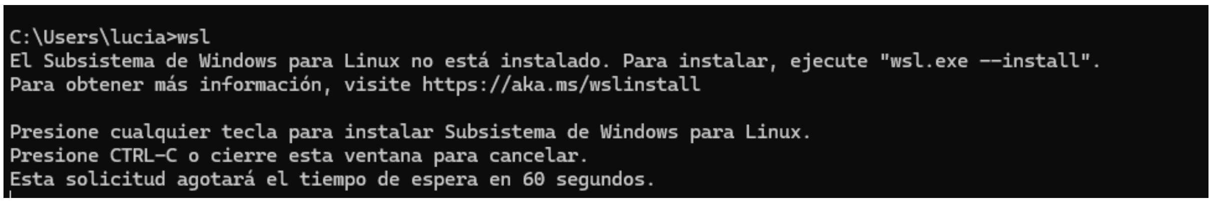
SUBIDA DE UN SERVICIO EN LA NUBE

Autor: Diestefano Jiampieri
Asignatura: Arquitectura en la Nube
Tutor: Rubén
Fecha de Entrega: 30/09/2025

ÍNDICE

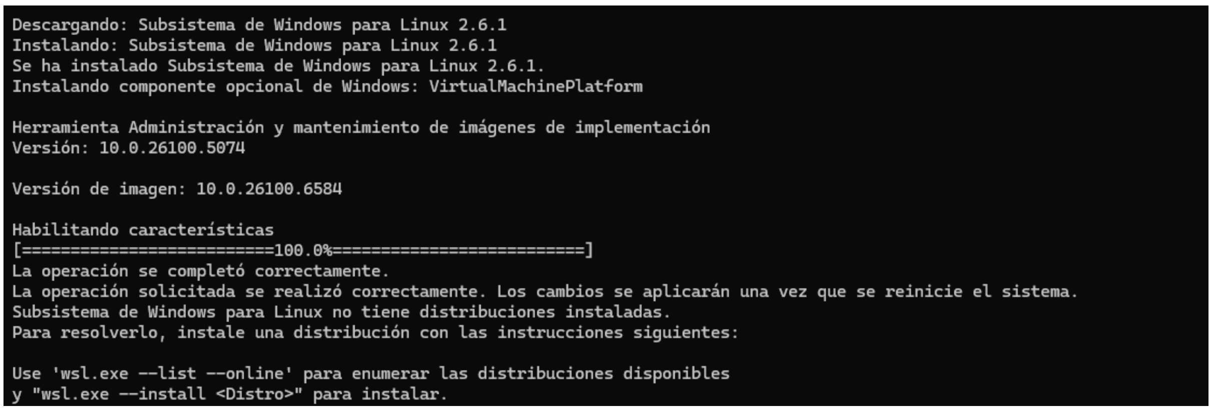
INSTALACIÓN WSL

1. Empezaremos instalando WSL en una máquina física, para ello, pondremos en la CMD "***WSL***" y le daremos enter.



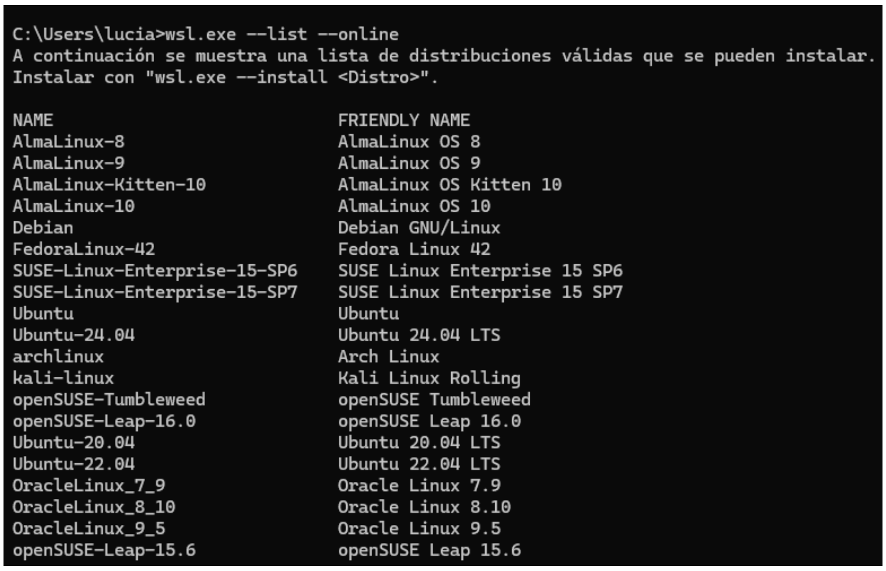
```
C:\Users\lucia>wsl
El Subsistema de Windows para Linux no está instalado. Para instalar, ejecute "wsl.exe --install".
Para obtener más información, visite https://aka.ms/wslinstall

Presione cualquier tecla para instalar Subsistema de Windows para Linux.
Presione CTRL-C o cierre esta ventana para cancelar.
Esta solicitud agotará el tiempo de espera en 60 segundos.
```

2. Le daremos que ***"sí"*** para que se instale.



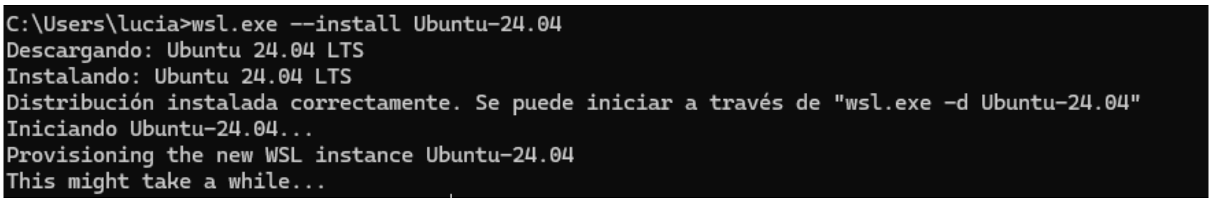
```
Descargando: Subsistema de Windows para Linux 2.6.1
Instalando: Subsistema de Windows para Linux 2.6.1
Se ha instalado Subsistema de Windows para Linux 2.6.1.
Instalando componente opcional de Windows: VirtualMachinePlatform

Herramienta Administración y mantenimiento de imágenes de implementación
Versión: 10.0.26100.5074

Versión de imagen: 10.0.26100.6584

Habilitando características
[==========================100.0%==========================]
La operación se completó correctamente.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
Subsistema de Windows para Linux no tiene distribuciones instaladas.
Para resolverlo, instale una distribución con las instrucciones siguientes:

Use 'wsl.exe --list --online' para enumerar las distribuciones disponibles
y "wsl.exe --install <Distro>" para instalar.
```

3. Una vez instalado, escribiremos ***"wsl.exe –list –online***" así podremos ver todas las opciones que podemos instalar.



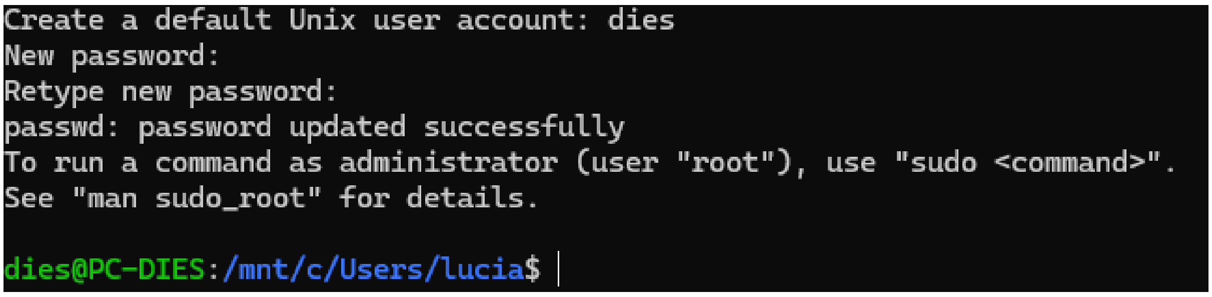
```
C:\Users\lucia>wsl.exe --list --online
A continuación se muestra una lista de distribuciones válidas que se pueden instalar.
Instalar con "wsl.exe --install <Distro>".

NAME                                FRIENDLY NAME
AlmaLinux-8                         AlmaLinux OS 8
AlmaLinux-9                         AlmaLinux OS 9
AlmaLinux-Kitten-10                 AlmaLinux OS Kitten 10
AlmaLinux-10                        AlmaLinux OS 10
Debian                              Debian GNU/Linux
FedoraLinux-42                      Fedora Linux 42
SUSE-Linux-Enterprise-15-SP6        SUSE Linux Enterprise 15 SP6
SUSE-Linux-Enterprise-15-SP7        SUSE Linux Enterprise 15 SP7
Ubuntu                              Ubuntu
Ubuntu-24.04                        Ubuntu 24.04 LTS
archlinux                           Arch Linux
kali-linux                          Kali Linux Rolling
openSUSE-Tumbleweed                 openSUSE Tumbleweed
openSUSE-Leap-16.0                  openSUSE Leap 16.0
Ubuntu-20.04                        Ubuntu 20.04 LTS
Ubuntu-22.04                        Ubuntu 22.04 LTS
OracleLinux_7_9                     Oracle Linux 7.9
OracleLinux_8_10                    Oracle Linux 8.10
OracleLinux_9_5                     Oracle Linux 9.5
openSUSE-Leap-15.6                  openSUSE Leap 15.6
```

4. En nuestro caso instalaremos Ubuntu, por lo que tendremos que poner, ***“wsl.exe –install Ubuntu-24.04”***.



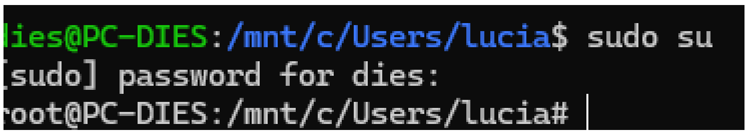
```
C:\Users\lucia>wsl.exe --install Ubuntu-24.04
Descargando: Ubuntu 24.04 LTS
Instalando: Ubuntu 24.04 LTS
Distribución instalada correctamente. Se puede iniciar a través de "wsl.exe -d Ubuntu-24.04"
Iniciando Ubuntu-24.04...
Provisioning the new WSL instance Ubuntu-24.04
This might take a while...
```

5. Una vez instalado nos pedirá que creemos una cuenta para Ubuntu.



```
Create a default Unix user account: dies
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dies@PC-DIES:/mnt/c/Users/lucia$
```

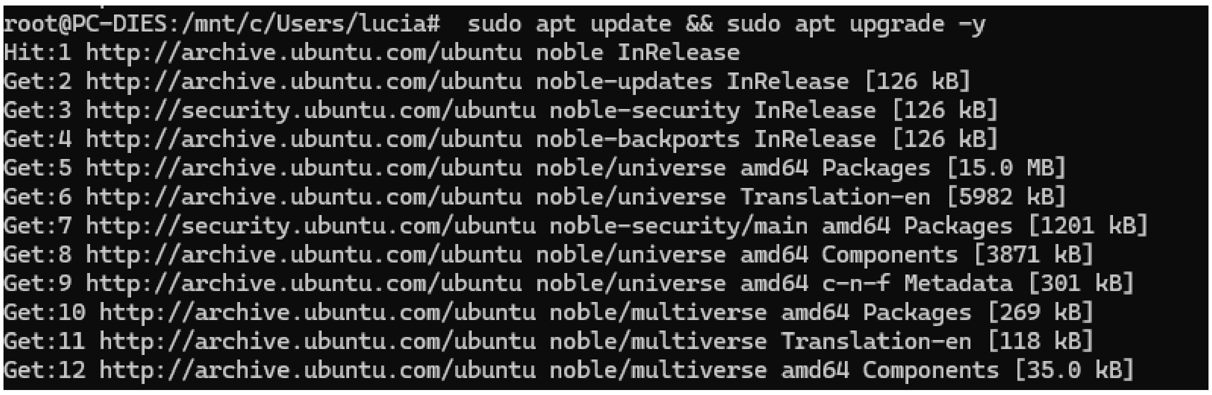
6. Pondremos un ***“sudo su”*** para tener permisos de administrador.



```
lies@PC-DIES:/mnt/c/Users/lucia$ sudo su
[sudo] password for dies:
root@PC-DIES:/mnt/c/Users/lucia#
```

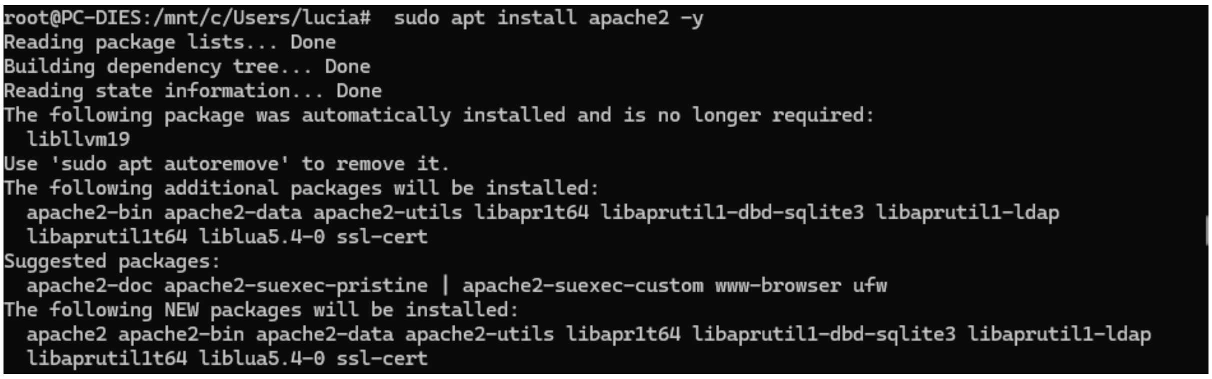
SERVIDOR APACHE CON PHP

1. Empezaremos actualizando el sistema con el comando ***“sudo apt update && sudo apt upgrade -y”***.



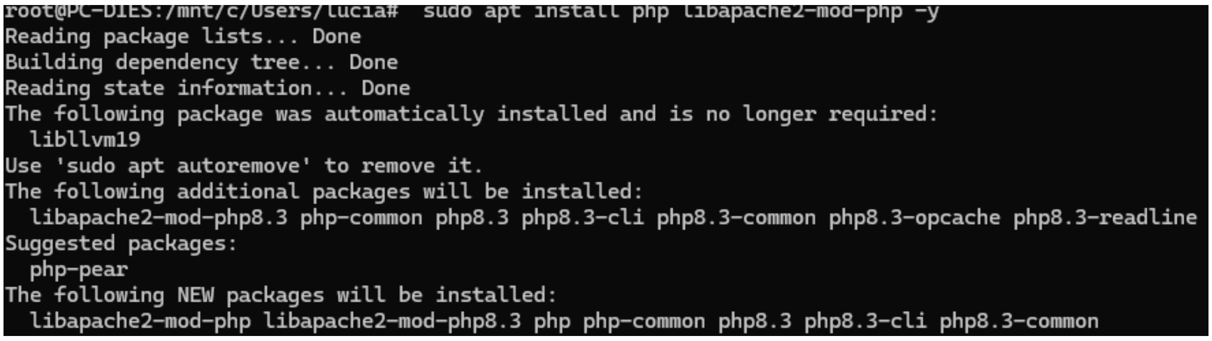
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo apt update && sudo apt upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1201 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

2. Instalamos Apache2 con el comando ***”sudo apt install apache2 -y”***.



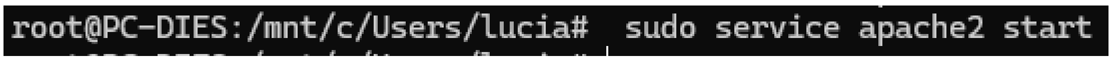
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 liblua5.4-0 ssl-cert
```

3. Instalamos PHP con el comando ***”sudo apt install php libapache2-mod-php -y”.***



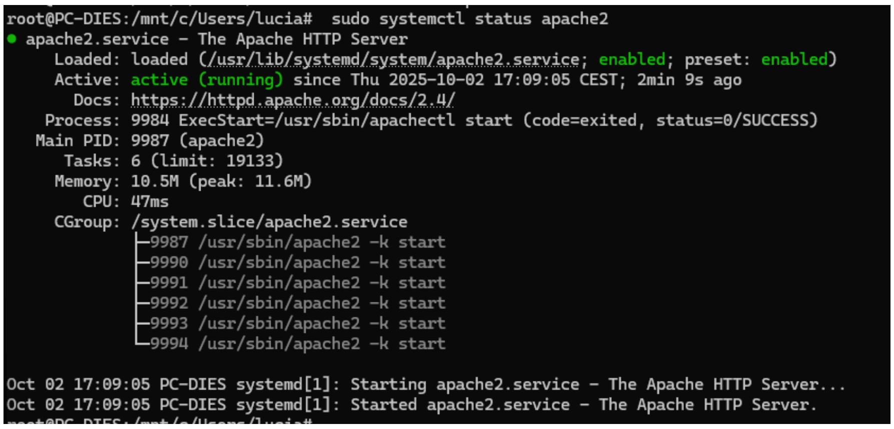
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo apt install php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libapache2-mod-php8.3 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 php php-common php8.3 php8.3-cli php8.3-common
```

4. Iniciaremos Apache2 mediante el comando ***“sudo service apache2 start”***.



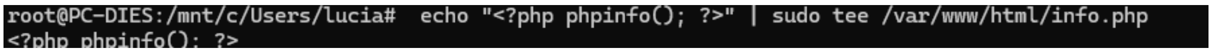
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo service apache2 start
```

5. Con el siguiente comando ***“sudo systemctl status apache2”*** veremos el estado apache.



```
root@PC-DIES:/mnt/c/Users/lucia#  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 Thu 2025-10-02 17:09:05 CEST; 2min 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 9984 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 9987 (apache2)
      Tasks: 6 (limit: 19133)
     Memory: 10.5M (peak: 11.6M)
        CPU: 47ms
     CGroup: /system.slice/apache2.service
             ├─9987 /usr/sbin/apache2 -k start
             ├─9990 /usr/sbin/apache2 -k start
             ├─9991 /usr/sbin/apache2 -k start
             ├─9992 /usr/sbin/apache2 -k start
             ├─9993 /usr/sbin/apache2 -k start
             └─9994 /usr/sbin/apache2 -k start

Oct 02 17:09:05 PC-DIES systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 02 17:09:05 PC-DIES systemd[1]: Started apache2.service - The Apache HTTP Server.
```

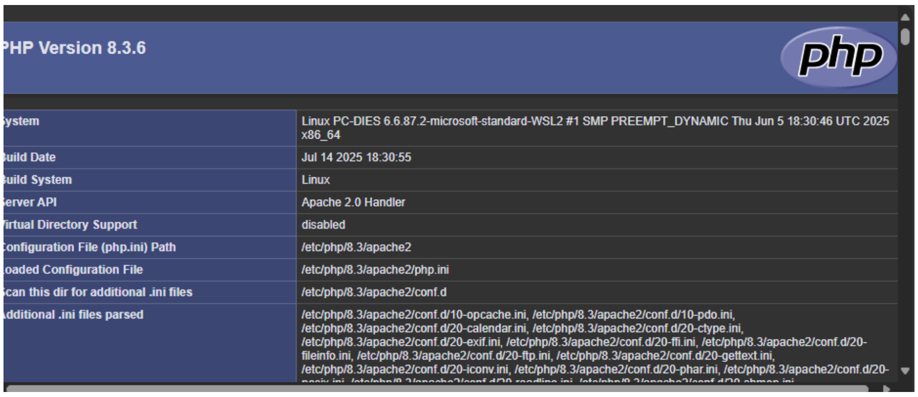
6. Crearemos un archivo PHP con el comando ***“echo "" | sudo tee /var/www/html/info.php ”.***



```
root@PC-DIES:/mnt/c/Users/lucia#  echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
```

7. Una vez creado el archivo verificaremos si se ha creado correctamente, lo podremos ver desde el navegador o desde la CMD.

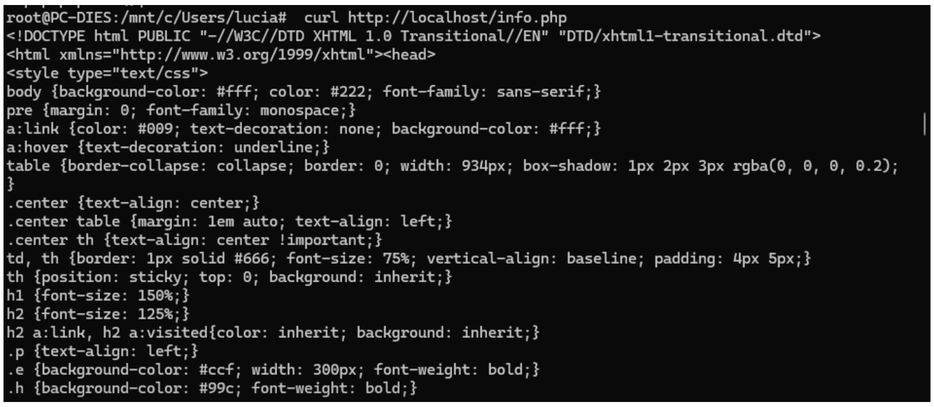
- GOOGLE:



PHP Version 8.3.6

ystem	Linux PC-DIES 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86_64
uild Date	Jul 14 2025 18:30:55
uild System	Linux
erver API	Apache 2.0 Handler
irtual Directory Support	disabled
onfiguration File (php.ini) Path	/etc/php/8.3/apache2
oaded Configuration File	/etc/php/8.3/apache2/php.ini
can this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
dditional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-

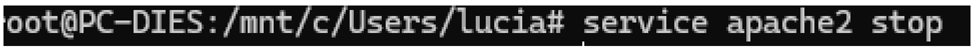
- CMD:



```
root@PC-DIES:/mnt/c/Users/lucia#  curl http://localhost/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);
}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
h2 a:link, h2 a:visited{color: inherit; background: inherit;}
.p {text-align: left;}
.e {background-color: #ccf; width: 300px; font-weight: bold;}
.h {background-color: #99c; font-weight: bold;}
```

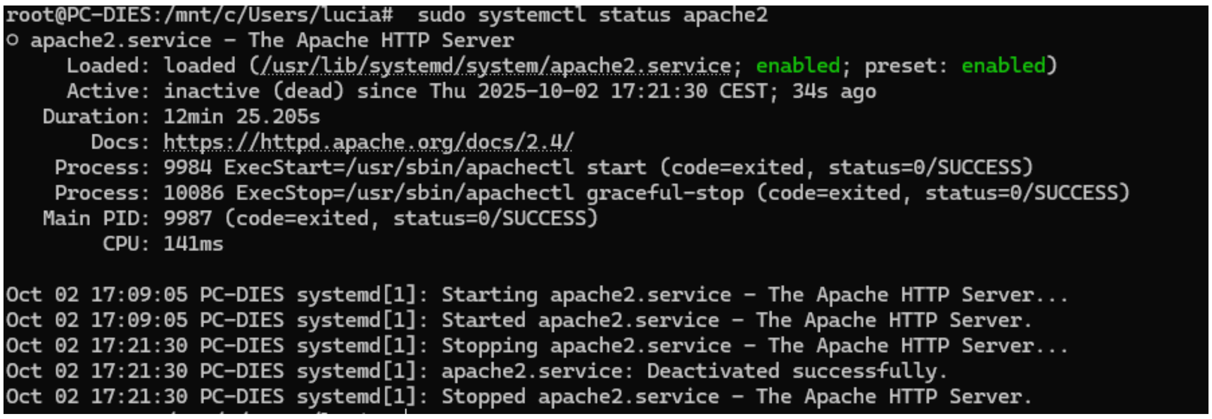
SERVIDOR NGINX CON HTML

1. Detendremos el Apache con el comando ***“service apache2 stop”.***



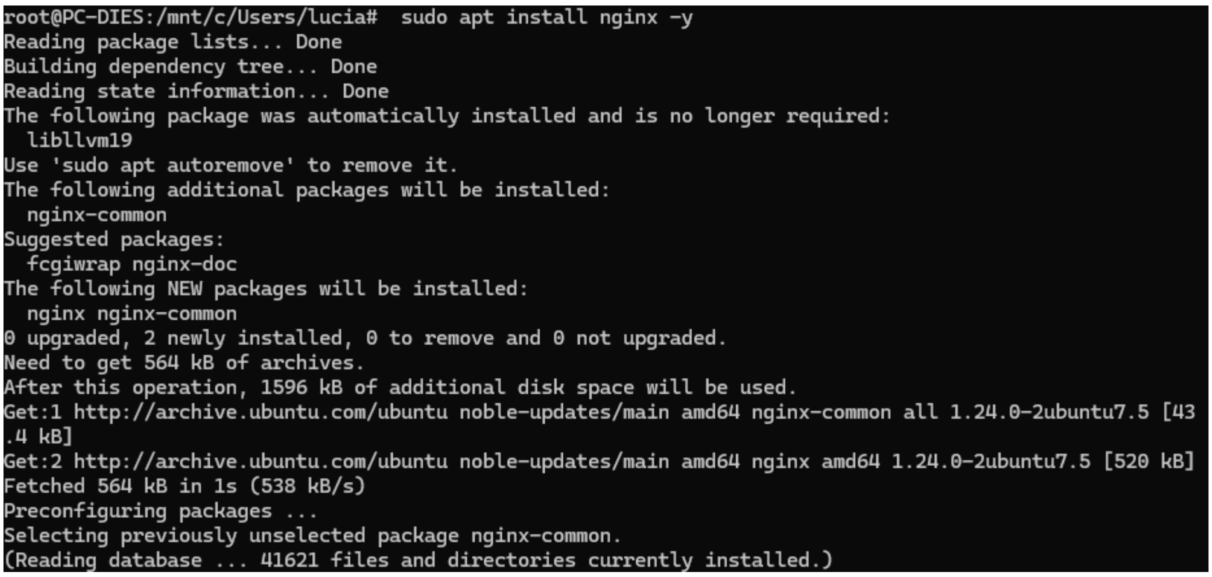
```
oot@PC-DIES:/mnt/c/Users/lucia# service apache2 stop
```

2. Y veremos el estado que está apagado.



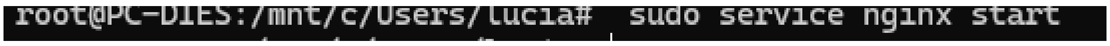
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo systemctl status apache2
○ apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: inactive (dead) since Thu 2025-10-02 17:21:30 CEST; 34s ago
   Duration: 12min 25.205s
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 9984 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Process: 10086 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
   Main PID: 9987 (code=exited, status=0/SUCCESS)
        CPU: 141ms

Oct 02 17:09:05 PC-DIES systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 02 17:09:05 PC-DIES systemd[1]: Started apache2.service - The Apache HTTP Server.
Oct 02 17:21:30 PC-DIES systemd[1]: Stopping apache2.service - The Apache HTTP Server...
Oct 02 17:21:30 PC-DIES systemd[1]: apache2.service: Deactivated successfully.
Oct 02 17:21:30 PC-DIES systemd[1]: Stopped apache2.service - The Apache HTTP Server.
```

3. Para instalar Nginx escribiremos ***“sudo apt install nginx -y”.***



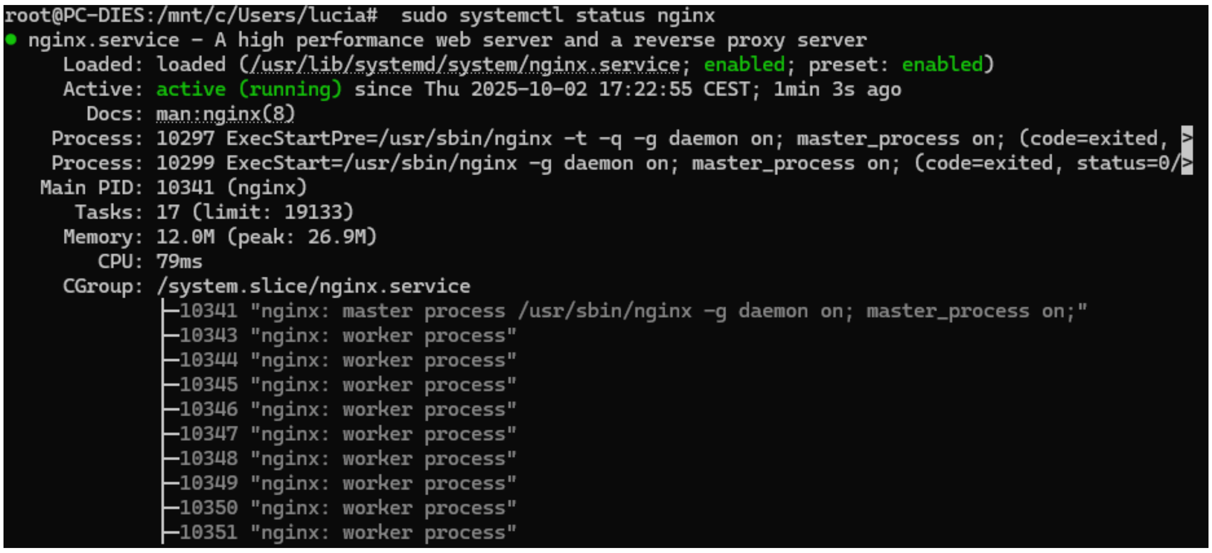
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 564 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43
.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Fetched 564 kB in 1s (538 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 41621 files and directories currently installed.)
```

4. Iniciaremos Nginx mediante el siguiente comando ***“sudo service nginx start”.***



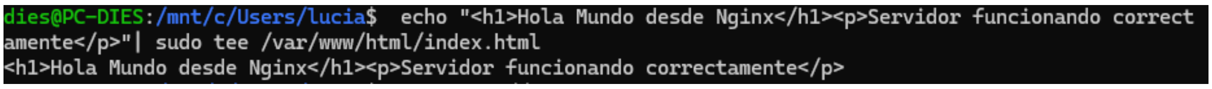
```
root@PC-DIES:/mnt/c/Users/lucia#  sudo service nginx start
```

5. Para ver el estado de Nginx escribimos ***"sudo systemctl status nginx"***.



```
root@PC-DIES:/mnt/c/Users/lucia#  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: enabled)
     Active: active (running) since Thu 2025-10-02 17:22:55 CEST; 1min 3s ago
       Docs: man:nginx(8)
    Process: 10297 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, >
    Process: 10299 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/>
   Main PID: 10341 (nginx)
      Tasks: 17 (limit: 19133)
     Memory: 12.0M (peak: 26.9M)
        CPU: 79ms
     CGroup: /system.slice/nginx.service
             ├─10341 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─10343 "nginx: worker process"
             ├─10344 "nginx: worker process"
             ├─10345 "nginx: worker process"
             ├─10346 "nginx: worker process"
             ├─10347 "nginx: worker process"
             ├─10348 "nginx: worker process"
             ├─10349 "nginx: worker process"
             ├─10350 "nginx: worker process"
             ├─10351 "nginx: worker process"
```

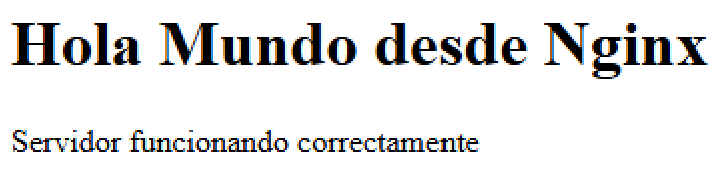
6. Ahora crearemos una página HTML, mediante ***echo " Hola Mundo desde Nginx Servidor funcionando correctamente" | sudo tee /var/www/html/index.html"***



```
dies@PC-DIES:/mnt/c/Users/lucia$  echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correct
amente</p>"| sudo tee /var/www/html/index.html
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

7. Accederemos y veremos si sale lo que hemos puesto en el HTML, ***"http://localhost".***

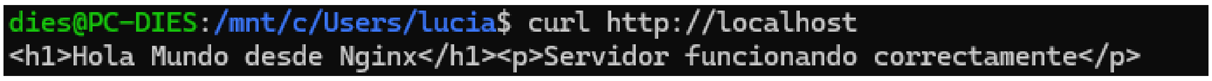
- <u>GOOGLE:</u>



Hola Mundo desde Nginx

Servidor funcionando correctamente

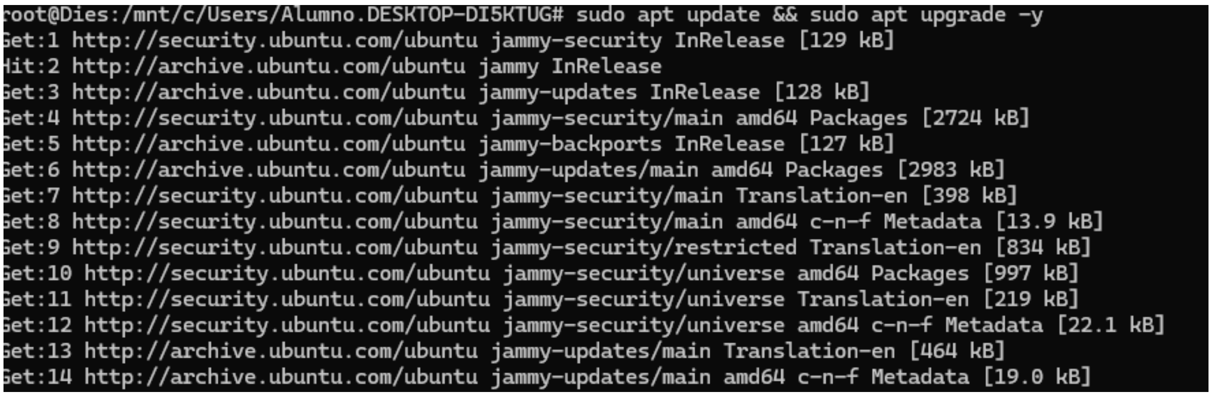
- <u>CMD:</u>



```
dies@PC-DIES:/mnt/c/Users/lucia$ curl http://localhost
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

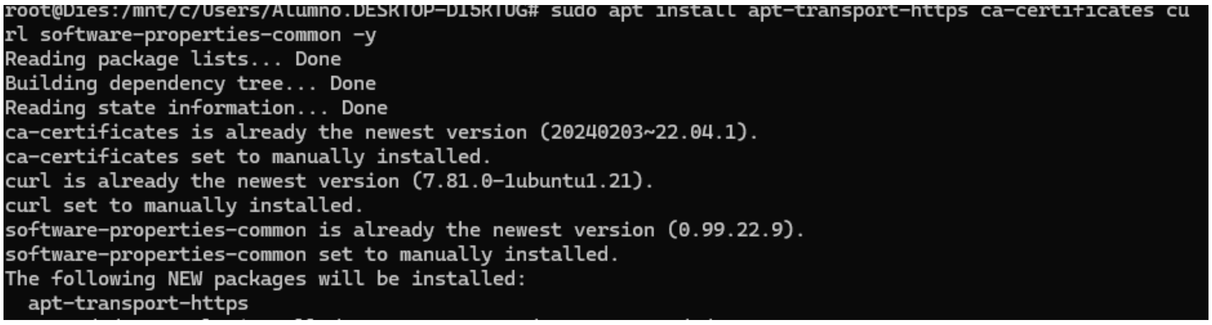
INSTALACIÓN DE DOCKER

1. Empezaremos con la actualización del sistema, “***sudo apt update && sudo apt upgrade -y***”.



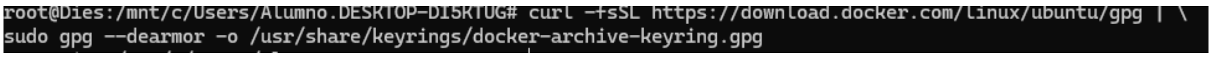
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo apt update && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2724 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2983 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [398 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.9 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [834 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [997 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [219 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [22.1 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [464 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [19.0 kB]
```

2. Posteriormente instalamos los paquetes necesarios para añadir los repositorios https, con el comando “***sudo apt install apt-transport-https ca-certificates curl software-properties-common -y***”.



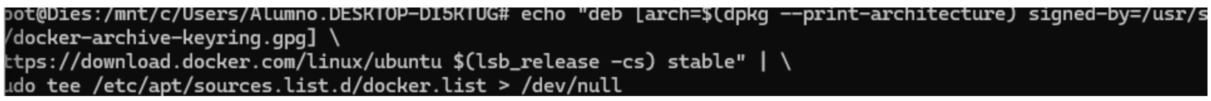
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo apt install apt-transport-https ca-certificates cu
rl software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.21).
curl set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
```

3. Ahora añadiremos las claves GPG de Docker, ***“curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \ sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg***”.



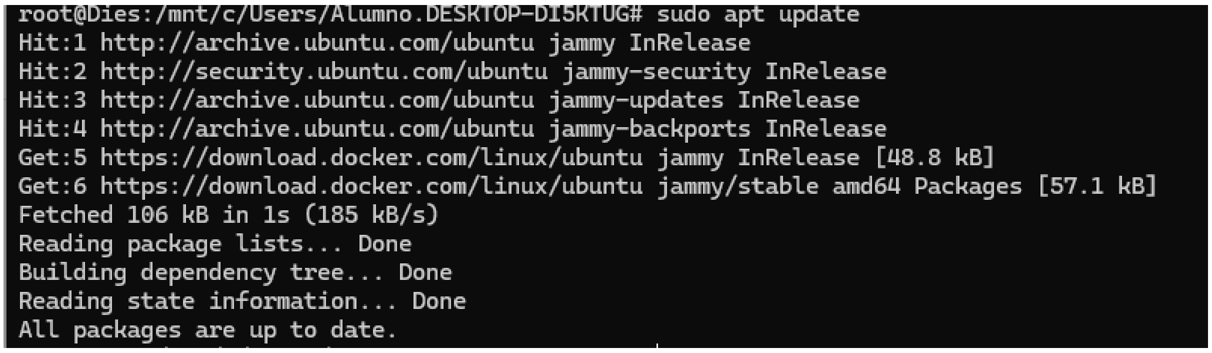
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

4. A continuación añadiremos el repositorio Docker a las fuentes APT, con el comando “***echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \ https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \ sudo tee /etc/apt/sources.list.d/docker.list > /dev/null*** ”.



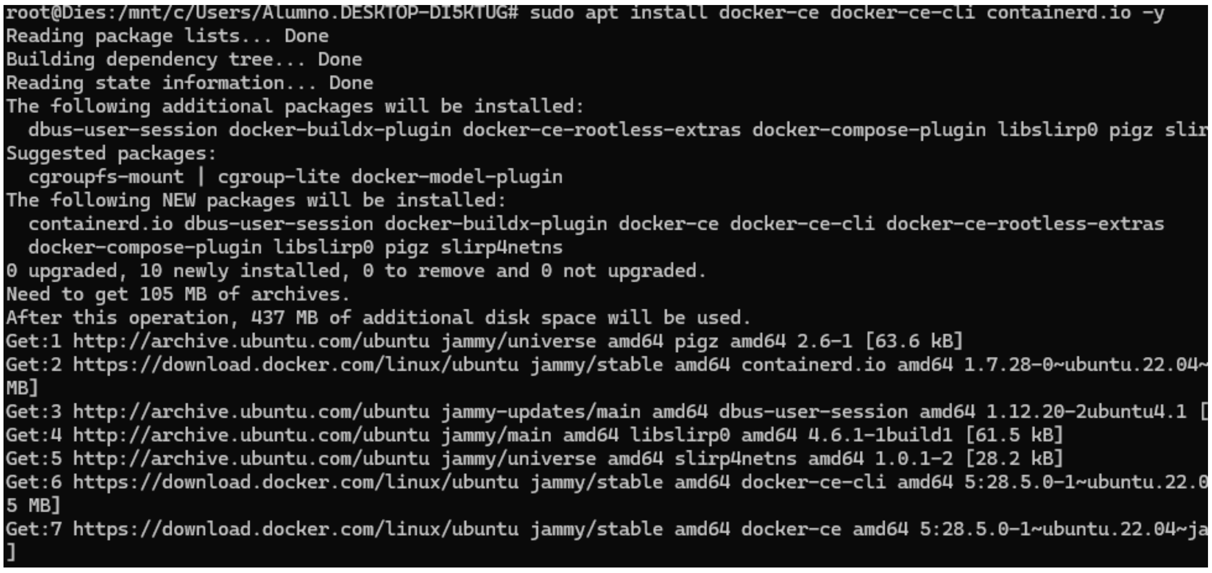
```
oot@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/s
/docker-archive-keyring.gpg] \
ttps://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
udo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

5. Actualizaremos los paquetes con el nuevo repositorio, para ello escribiremos "***sudo apt update***".



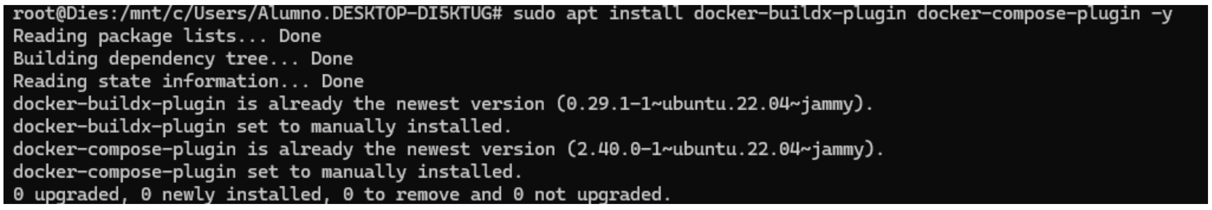
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [57.1 kB]
Fetched 106 kB in 1s (185 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

6. Instalaremos Docker en el sistema de Ubuntu con el comando "***sudo apt install docker-ce docker-ce-cli containerd.io -y***".



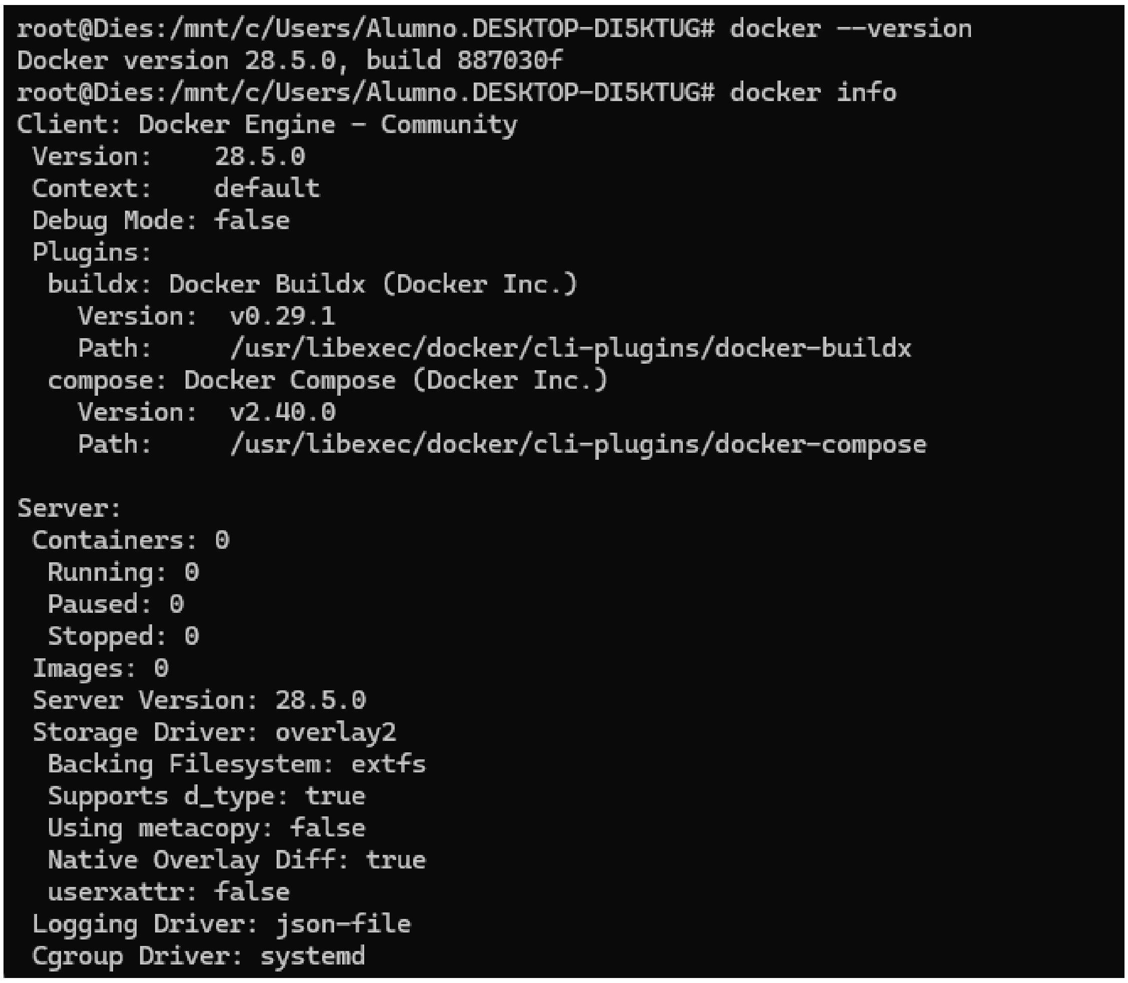
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo apt install docker-ce docker-ce-cli containerd.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus-user-session docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slir
Suggested packages:
  cgroupfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
  containerd.io dbus-user-session docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 105 MB of archives.
After this operation, 437 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.7.28-0~ubuntu.22.04~
MB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 dbus-user-session amd64 1.12.20-2ubuntu4.1 [
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:28.5.0-1~ubuntu.22.0
5 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:28.5.0-1~ubuntu.22.04~ja
]
```

7. Para terminar la instalación, descargamos los complementos restantes, para ello escribiremos "***sudo apt install docker-buildx-plugin docker-compose-plugin -y***".



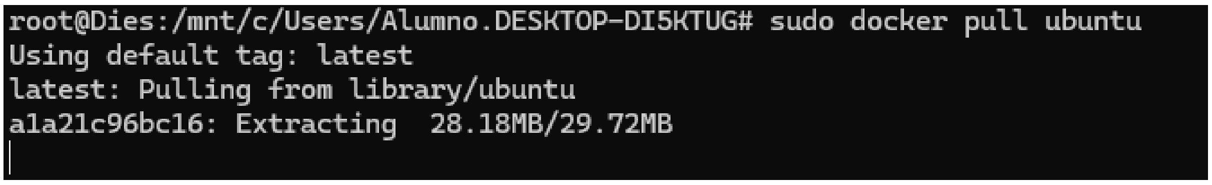
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo apt install docker-buildx-plugin docker-compose-plugin -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-buildx-plugin is already the newest version (0.29.1-1~ubuntu.22.04~jammy).
docker-buildx-plugin set to manually installed.
docker-compose-plugin is already the newest version (2.40.0-1~ubuntu.22.04~jammy).
docker-compose-plugin set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

8. Verificaremos la instalación, para estar seguros de que todo esté correctamente “***docker --version***” y “***docker info***”.



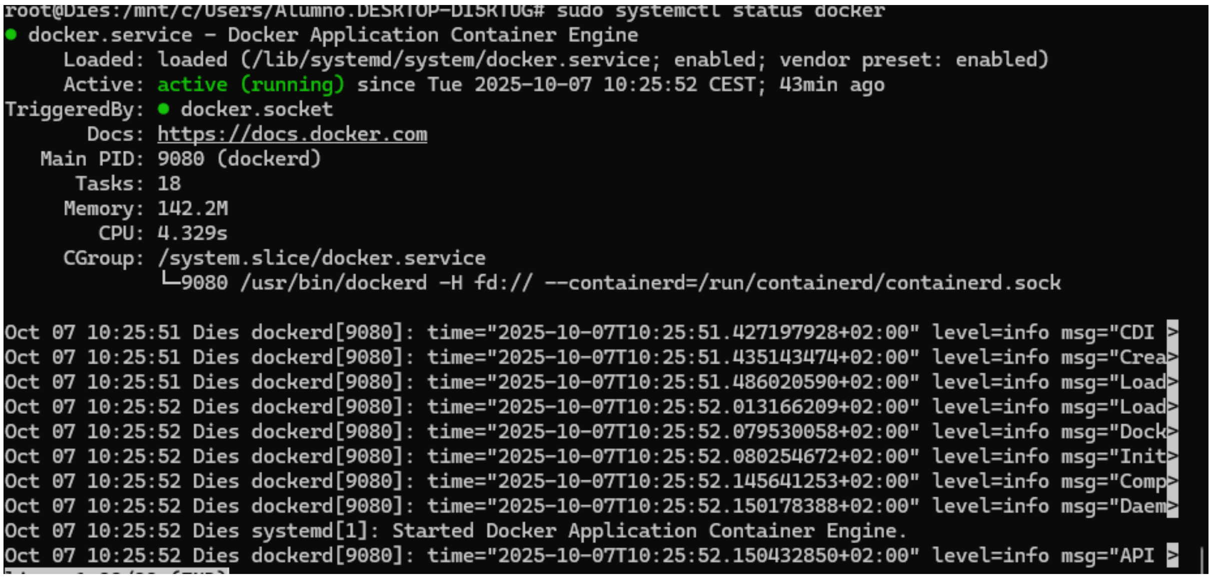
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# docker --version
Docker version 28.5.0, build 887030f
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# docker info
Client: Docker Engine - Community
 Version:    28.5.0
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.29.1
    Path:     /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.40.0
    Path:     /usr/libexec/docker/cli-plugins/docker-compose

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 28.5.0
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
```

9. Instalamos la última versión de ubuntu con Docker “***sudo docker pull ubuntu***”.



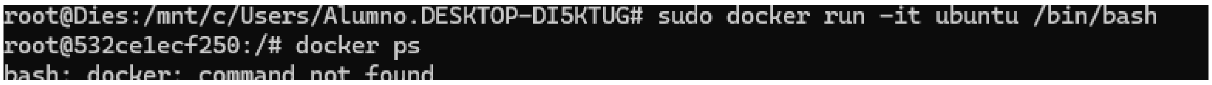
```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
a1a21c96bc16: Extracting  28.18MB/29.72MB
```

10. Revisamos por si acaso el estado de Docker, “***sudo systemctl status docke***r”.



```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-10-07 10:25:52 CEST; 43min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 9080 (dockerd)
      Tasks: 18
     Memory: 142.2M
        CPU: 4.329s
     CGroup: /system.slice/docker.service
             └─9080 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 07 10:25:51 Dies dockerd[9080]: time="2025-10-07T10:25:51.427197928+02:00" level=info msg="CDI >
Oct 07 10:25:51 Dies dockerd[9080]: time="2025-10-07T10:25:51.435143474+02:00" level=info msg="Crea>
Oct 07 10:25:51 Dies dockerd[9080]: time="2025-10-07T10:25:51.486020590+02:00" level=info msg="Load>
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.013166209+02:00" level=info msg="Load>
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.079530058+02:00" level=info msg="Dock>
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.080254672+02:00" level=info msg="Init>
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.145641253+02:00" level=info msg="Comp>
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.150178388+02:00" level=info msg="Daem>
Oct 07 10:25:52 Dies systemd[1]: Started Docker Application Container Engine.
Oct 07 10:25:52 Dies dockerd[9080]: time="2025-10-07T10:25:52.150432850+02:00" level=info msg="API >
```

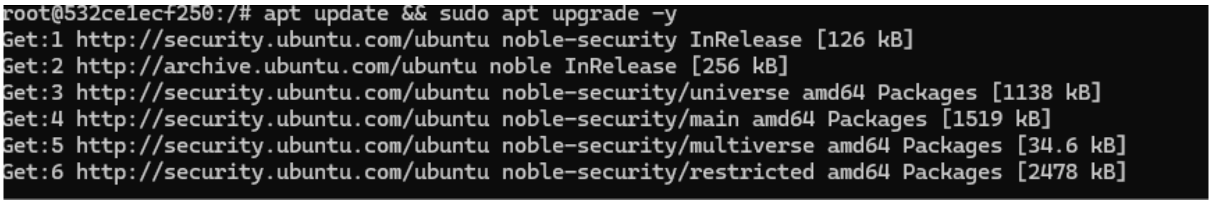
11. Crearemos un contenedor con el comando “***sudo docker run -it ubuntu /bin/bash***”.



```
root@Dies:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG# sudo docker run -it ubuntu /bin/bash
root@532ce1ecf250:/# docker ps
bash: docker: command not found
```

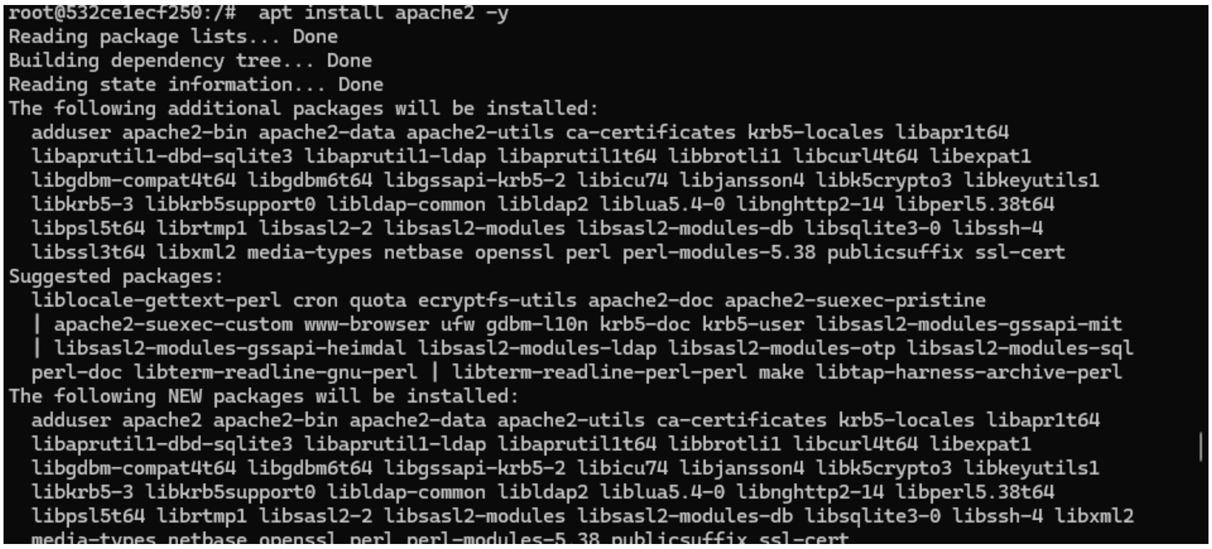
PHP EN DOCKER

1. Actualizaremos todo, ***“apt update”***.



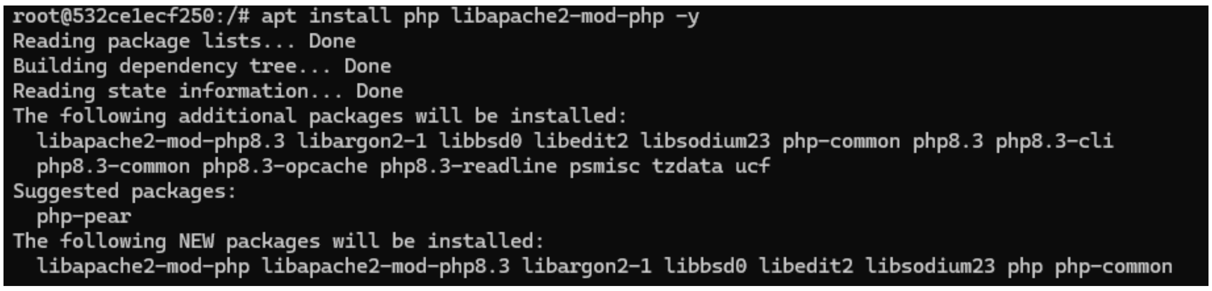
```
root@532ce1ecf250:/# apt update && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1138 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1519 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [34.6 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2478 kB]
```

2. Instalaremos nuevamente apache, “***apt install apache2 -y***”.



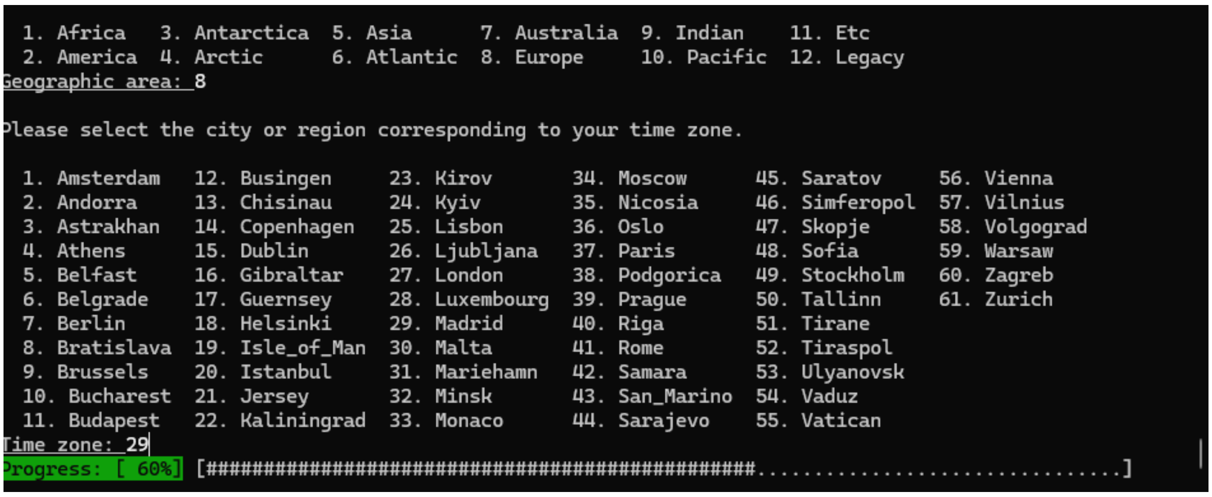
```
root@532ce1ecf250:/#  apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 libbrotli1 libcurl4t64 libexpat1
  libgdbm-compat4t64 libgdbm6t64 libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1
  libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp2-14 libperl5.38t64
  libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsqlite3-0 libssh-4
  libssl3t64 libxml2 media-types netbase openssl perl perl-modules-5.38 publicsuffix ssl-cert
Suggested packages:
  liblocale-gettext-perl cron quota ecryptfs-utils apache2-doc apache2-suexec-pristine
  | apache2-suexec-custom www-browser ufw gdbm-l10n krb5-doc krb5-user libsasl2-modules-gssapi-mit
  | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql
  perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  adduser apache2 apache2-bin apache2-data apache2-utils ca-certificates krb5-locales libapr1t64
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 libbrotli1 libcurl4t64 libexpat1
  libgdbm-compat4t64 libgdbm6t64 libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1
  libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp2-14 libperl5.38t64
  libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db libsqlite3-0 libssh-4 libxml2
  media-types netbase openssl perl perl-modules-5.38 publicsuffix ssl-cert
```

3. También instalaremos el PHP, “***apt install php libapache2-mod-php -y*** ”.



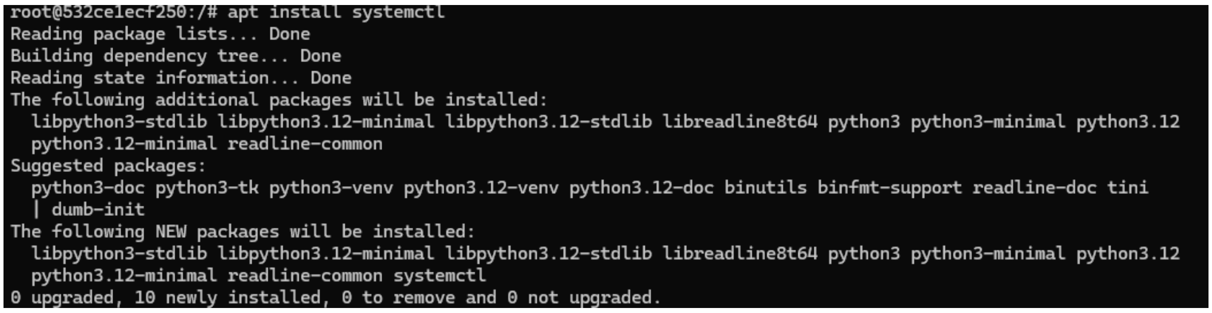
```
root@532ce1ecf250:/# apt install php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.3 libargon2-1 libbsd0 libedit2 libsodium23 php-common php8.3 php8.3-cli
  php8.3-common php8.3-opcache php8.3-readline psmisc tzdata ucf
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.3 libargon2-1 libbsd0 libedit2 libsodium23 php php-common
```

4. Escogeremos Europa y Madrid en nuestro caso.



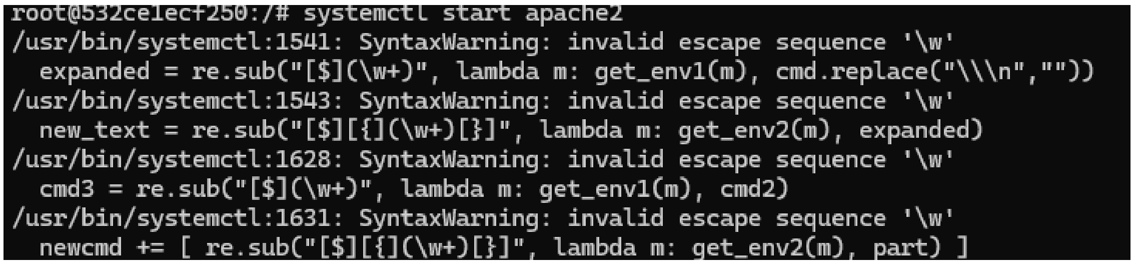
```
  1. Africa   3. Antarctica  5. Asia      7. Australia  9. Indian    11. Etc
  2. America  4. Arctic      6. Atlantic  8. Europe     10. Pacific  12. Legacy
Geographic area: 8

Please select the city or region corresponding to your time zone.

  1. Amsterdam   12. Busingen     23. Kirov       34. Moscow      45. Saratov     56. Vienna
  2. Andorra     13. Chisinau     24. Kyiv        35. Nicosia     46. Simferopol  57. Vilnius
  3. Astrakhan   14. Copenhagen   25. Lisbon      36. Oslo        47. Skopje      58. Volgograd
  4. Athens      15. Dublin       26. Ljubljana   37. Paris       48. Sofia       59. Warsaw
  5. Belfast     16. Gibraltar    27. London      38. Podgorica   49. Stockholm   60. Zagreb
  6. Belgrade    17. Guernsey     28. Luxembourg  39. Prague      50. Tallinn     61. Zurich
  7. Berlin      18. Helsinki     29. Madrid      40. Riga        51. Tirane
  8. Bratislava  19. Isle_of_Man  30. Malta       41. Rome        52. Tiraspol
  9. Brussels    20. Istanbul     31. Mariehamn   42. Samara      53. Ulyanovsk
  10. Bucharest  21. Jersey       32. Minsk       43. San_Marino  54. Vaduz
  11. Budapest   22. Kaliningrad  33. Monaco      44. Sarajevo    55. Vatican
Time zone: 29
Progress: [ 60%] [################################################..............................]
```

5. Instalaremos el systemctl, ya que no está instalado en Docker, usaremos el comando, "***apt install systemctl***".



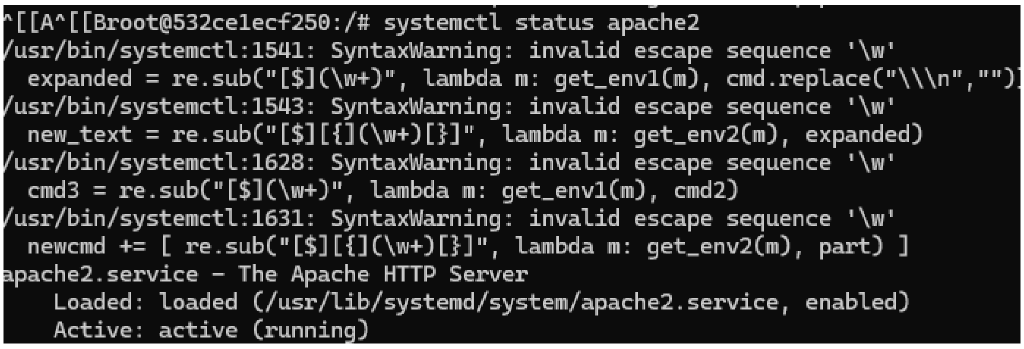
```
root@532ce1ecf250:/# apt install systemctl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib libreadline8t64 python3 python3-minimal python3.12
  python3.12-minimal readline-common
Suggested packages:
  python3-doc python3-tk python3-venv python3.12-venv python3.12-doc binutils binfmt-support readline-doc tini
  | dumb-init
The following NEW packages will be installed:
  libpython3-stdlib libpython3.12-minimal libpython3.12-stdlib libreadline8t64 python3 python3-minimal python3.12
  python3.12-minimal readline-common systemctl
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
```

6. Iniciaremos apache2 con "***systemctl start apache2***".



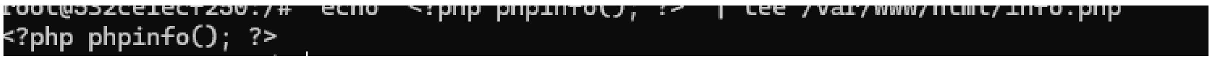
```
root@532ce1ecf250:/# systemctl start apache2
/usr/bin/systemctl:1541: SyntaxWarning: invalid escape sequence '\w'
  expanded = re.sub("[$](\w+)", lambda m: get_env1(m), cmd.replace("\\\n",""))
/usr/bin/systemctl:1543: SyntaxWarning: invalid escape sequence '\w'
  new_text = re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), expanded)
/usr/bin/systemctl:1628: SyntaxWarning: invalid escape sequence '\w'
  cmd3 = re.sub("[$](\w+)", lambda m: get_env1(m), cmd2)
/usr/bin/systemctl:1631: SyntaxWarning: invalid escape sequence '\w'
  newcmd += [ re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), part) ]
```

7. Verificamos que apache esté encendido, “***systemctl status apache2*** ”.



```
^[[A^[[Broot@532ce1ecf250:/# systemctl status apache2
/usr/bin/systemctl:1541: SyntaxWarning: invalid escape sequence '\w'
  expanded = re.sub("[$](\w+)", lambda m: get_env1(m), cmd.replace("\\\n",""))
/usr/bin/systemctl:1543: SyntaxWarning: invalid escape sequence '\w'
  new_text = re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), expanded)
/usr/bin/systemctl:1628: SyntaxWarning: invalid escape sequence '\w'
  cmd3 = re.sub("[$](\w+)", lambda m: get_env1(m), cmd2)
/usr/bin/systemctl:1631: SyntaxWarning: invalid escape sequence '\w'
  newcmd += [ re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), part) ]
apache2.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/apache2.service, enabled)
    Active: active (running)
```

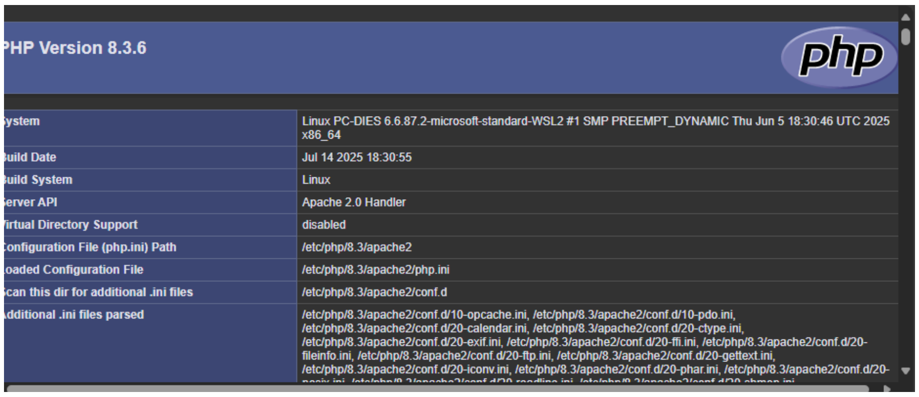
8. Crearemos el archivo PHP info, “ ***echo "" | tee /var/www/html/info.php*** ”.



```
root@532ce1ecf250:/# echo "<?php phpinfo(); ?>" | tee /var/www/html/info.php
<?php phpinfo(); ?>
```

9. Verificaremos que todo esté correctamente:

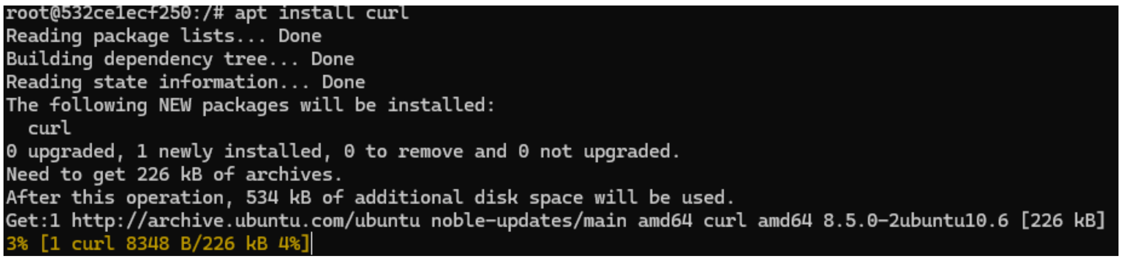
- <u>GOOGLE</u>



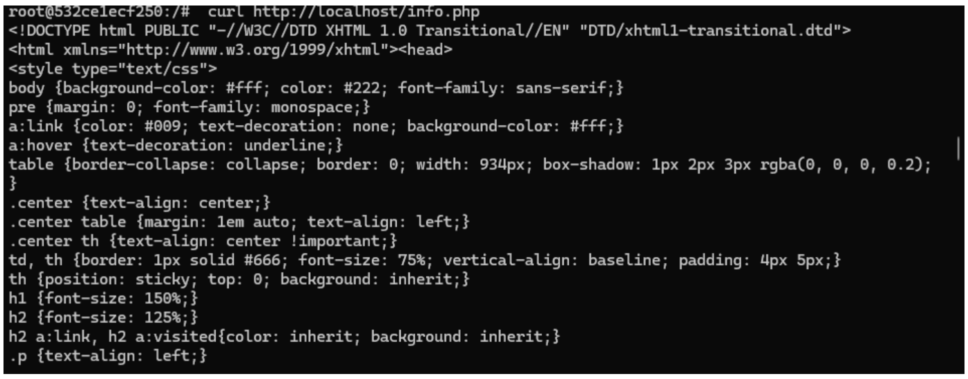
PHP Version 8.3.6

ystem	Linux PC-DIES 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86_64
uild Date	Jul 14 2025 18:30:55
uild System	Linux
erver API	Apache 2.0 Handler
irtual Directory Support	disabled
onfiguration File (php.ini) Path	/etc/php/8.3/apache2
oaded Configuration File	/etc/php/8.3/apache2/php.ini
can this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
dditional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-

- <u>CMD:</u> Para ello necesitaremos instalar el curl con “***apt install curl***”



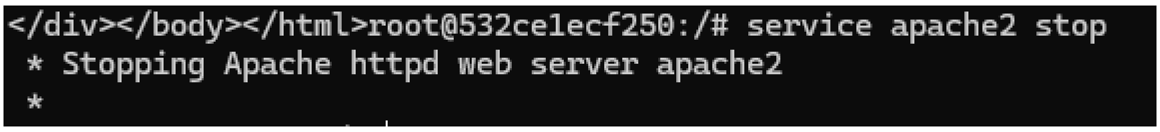
```
root@532ce1ecf250:/# apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 226 kB of archives.
After this operation, 534 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.6 [226 kB]
3% [1 curl 8348 B/226 kB 4%]
```



```
root@532ce1ecf250:/#  curl http://localhost/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);
}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
h2 a:link, h2 a:visited{color: inherit; background: inherit;}
.p {text-align: left;}
```

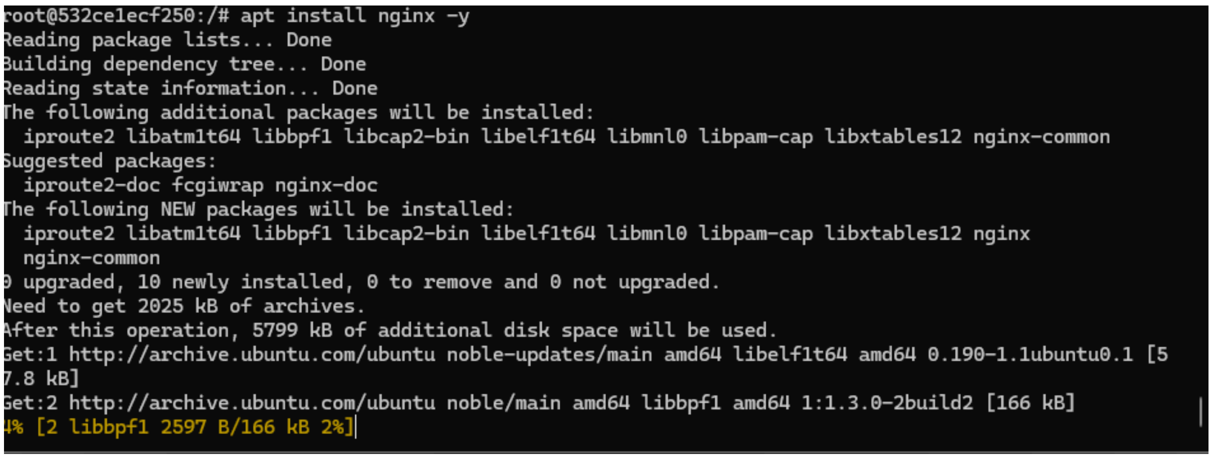
NGINX EN DOCKER

1. Pararemos el apache con “service apache2 stop”.



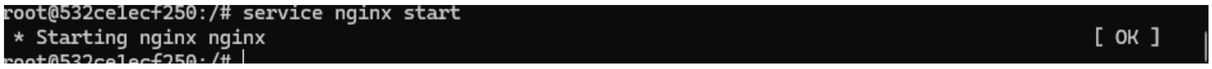
```
</div></body></html>root@532ce1ecf250:/# service apache2 stop
 * Stopping Apache httpd web server apache2
 *
```

2. Instalaremos el Nginx “apt install nginx -y ”.



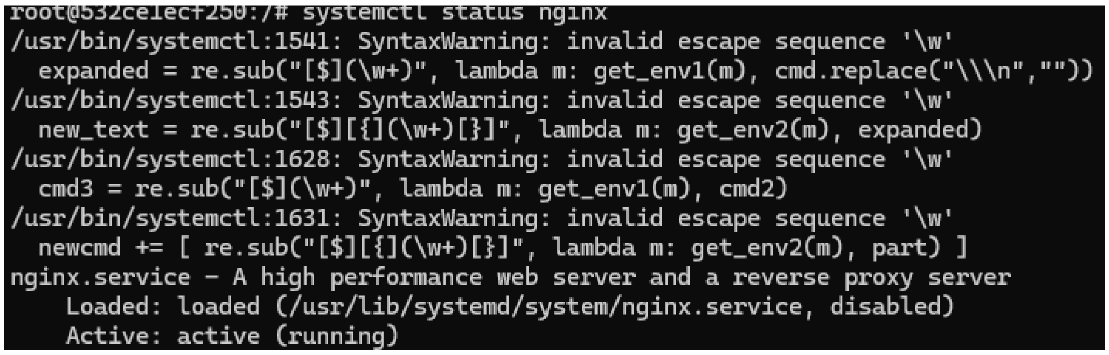
```
root@532ce1ecf250:/# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  iproute2 libatm1t64 libbpf1 libcap2-bin libelf1t64 libmnl0 libpam-cap libxtables12 nginx-common
Suggested packages:
  iproute2-doc fcgiwrap nginx-doc
The following NEW packages will be installed:
  iproute2 libatm1t64 libbpf1 libcap2-bin libelf1t64 libmnl0 libpam-cap libxtables12 nginx
  nginx-common
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2025 kB of archives.
After this operation, 5799 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libelf1t64 amd64 0.190-1.1ubuntu0.1 [5
7.8 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libbpf1 amd64 1:1.3.0-2build2 [166 kB]
4% [2 libbpf1 2597 B/166 kB 2%]
```

3. Iniciaremos Nginx “service nginx start ”.



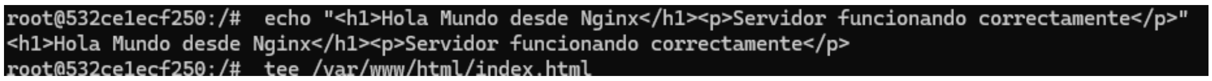
```
root@532ce1ecf250:/# service nginx start
 * Starting nginx nginx                                                              [ OK ]
root@532ce1ecf250:/#
```

4. Verificaremos el estado de Nginx ”systemctl status nginx ”.



```
root@532ce1ecf250:/# systemctl status nginx
/usr/bin/systemctl:1541: SyntaxWarning: invalid escape sequence '\w'
  expanded = re.sub("[$](\w+)", lambda m: get_env1(m), cmd.replace("\\\n",""))
/usr/bin/systemctl:1543: SyntaxWarning: invalid escape sequence '\w'
  new_text = re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), expanded)
/usr/bin/systemctl:1628: SyntaxWarning: invalid escape sequence '\w'
  cmd3 = re.sub("[$](\w+)", lambda m: get_env1(m), cmd2)
/usr/bin/systemctl:1631: SyntaxWarning: invalid escape sequence '\w'
  newcmd += [ re.sub("[$][{](\w+)[}]", lambda m: get_env2(m), part) ]
nginx.service - A high performance web server and a reverse proxy server
    Loaded: loaded (/usr/lib/systemd/system/nginx.service, disabled)
    Active: active (running)
```

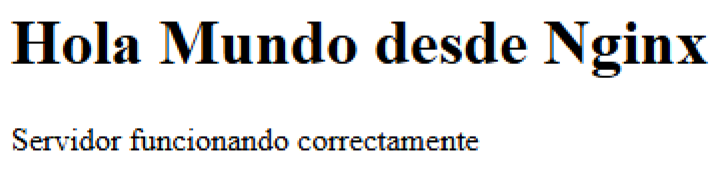
5. Creamos el archivo HTML con el comando ***echo " Hola Mundo desde Nginx Servidor funcionando correctamente" | sudo tee /var/www/html/index.html”***



```
root@532ce1ecf250:/#  echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>"
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
root@532ce1ecf250:/#  tee /var/www/html/index.html
```

6. Verificaremos en el navegador:

- GOOGLE:



Hola Mundo desde Nginx

Servidor funcionando correctamente