

# Implementación de un servidor Linux para entornos empresariales

19/06/2025

Angel Moreno Garcia  
CodeArts Solutions  
Madrid

## Introducción

En este proyecto se ha instalado y configurado un servidor Linux (**Ubuntu Server 25.04 LTS**), con el objetivo de preparar una base estable para entornos empresariales. El servidor será utilizado para alojar servicios web, repositorios de código y documentación interna.

A lo largo del proceso se aplicaron medidas de seguridad básica, estructura de carpetas organizadas, y optimización de accesos remotos a través de SSH.

## Fase 1: Instalación del sistema base

- Hypervisor: Para la virtualización de las máquinas se ha empleado **VirtualBox**.
- ISO del sistema: La ISO del sistema es **Ubuntu Server 25.04 LTS**.
- Configuración hardware de la VM:
  - **CPU:** 3 núcleos
  - **RAM:** 4 GB
  - **Disco:** 50 GB

Adaptadores de red: Tipo **Bridge**

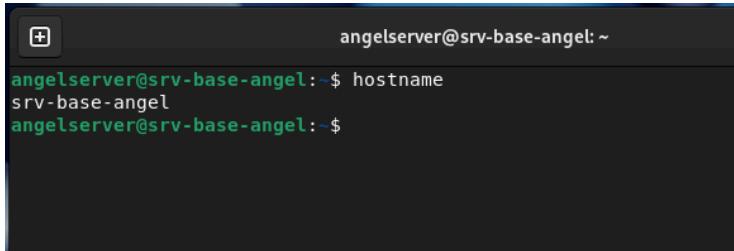
Conexión SSH habilitada: Sí

1. Lo primero es establecer la zona horaria con el comando: **timedatectl set-timezone Europe/Madrid**



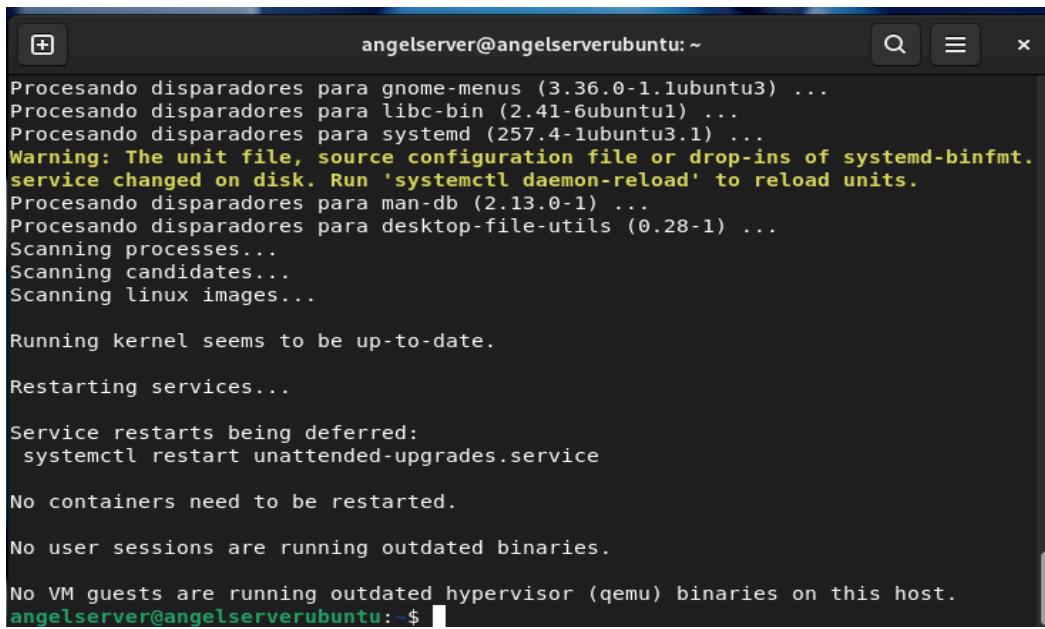
angelserver@angelserver:~\$ timedatectl status  
 Local time: mié 2025-06-18 14:46:07 CEST  
 Universal time: mié 2025-06-18 12:46:07 UTC  
 RTC time: mié 2025-06-18 12:41:46  
 Time zone: Europe/Madrid (CEST, +0200)  
 System clock synchronized: yes  
 NTP service: active  
 RTC in local TZ: no  
 angelserver@angelserver:~\$

2. Cambiamos el nombre del host modificando el archivo `/etc/hostname` con el comando: ***sudo nano /etc/hostname*** para que sea **srv-base-angel** y reiniciamos.



angelserver@srv-base-angel:~\$ hostname  
 srv-base-angel  
 angelserver@srv-base-angel:~\$

3. Usuario administrador configurado con contraseña segura.
4. Actualización de sistema con los comandos: ***sudo apt update && sudo apt upgrade***.



```
angelserver@angelserverubuntu:~$ Procesando disparadores para gnome-menus (3.36.0-1.1ubuntu3) ...
Procesando disparadores para libc-bin (2.41-6ubuntu1) ...
Procesando disparadores para systemd (257.4-1ubuntu3.1) ...
Warning: The unit file, source configuration file or drop-ins of systemd-binfmt.service changed on disk. Run 'systemctl daemon-reload' to reload units.
Procesando disparadores para man-db (2.13.0-1) ...
Procesando disparadores para desktop-file-utils (0.28-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
  systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
angelserver@angelserverubuntu:~$
```

## Fase 2: Configuración de red y acceso remoto

### Configuración de IP estática:

Para la configuración de la IP estática accederemos al plan de red a través del comando : **cd /etc/netplan** y entraremos (en mi caso) dentro del archivo llamado **50-cloud-init.yaml**. Una vez dentro modificaremos el archivo con estos parámetros.

Después usaremos el comando **sudo netplan apply** para aplicar los cambios y **ip route** para confirmar la información.

**network:**

**version: 2**

**ethernets:**

**enp0s3:**

**dhcp4: no**

**addresses: [192.168.1.21/24]**

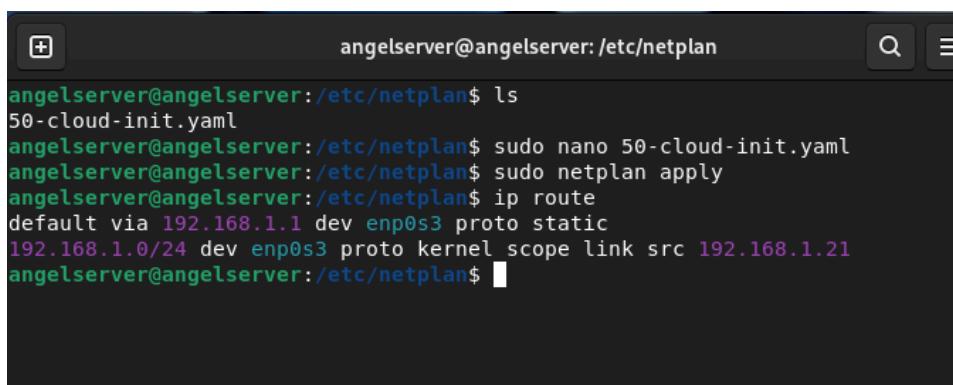
**routes:**

- to: 0.0.0.0/0

via: 192.168.1.1

**nameservers:**

**addresses: [8.8.8.8, 8.8.4.4]**

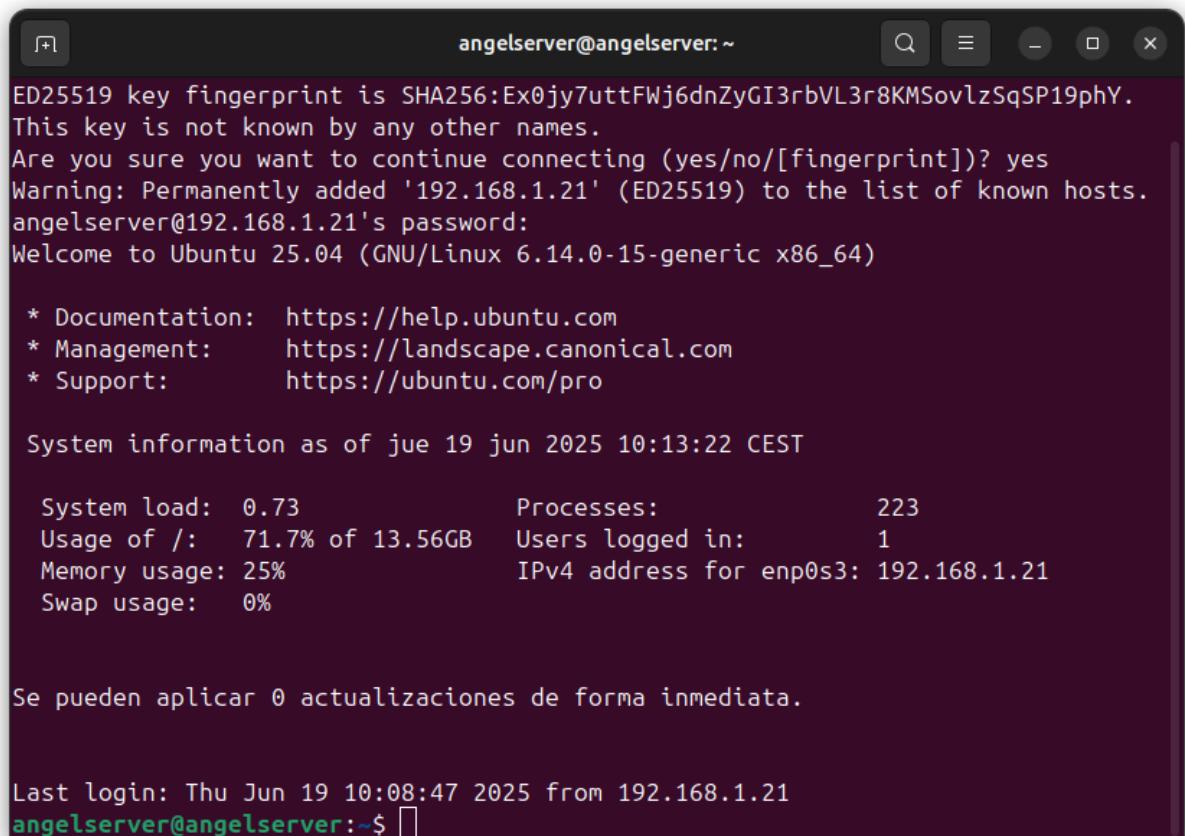


```
angelserv@angelserv:/etc/netplan$ ls
50-cloud-init.yaml
angelserv@angelserv:/etc/netplan$ sudo nano 50-cloud-init.yaml
angelserv@angelserv:/etc/netplan$ sudo netplan apply
angelserv@angelserv:/etc/netplan$ ip route
default via 192.168.1.1 dev enp0s3 proto static
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.21
angelserv@angelserv:/etc/netplan$
```

## Instalación del servicio SSH

Ahora instalaremos el servicio SSH con los comandos: ***sudo apt install openssh-server -y***, ***sudo systemctl enable ssh*** y ***sudo systemctl start ssh***.

Después de la instalación procederemos a la verificación de conexión remota desde otro sistema con el comando: ***ssh srv-base-angel@192.168.1.21***



```
angelsrv@angelsrv:~$ ED25519 key fingerprint is SHA256:Ex0jy7uttFWj6dnZyGI3rbVL3r8KMSovlzSqSP19phY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.21' (ED25519) to the list of known hosts.
angelsrv@192.168.1.21's password:
Welcome to Ubuntu 25.04 (GNU/Linux 6.14.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of jue 19 jun 2025 10:13:22 CEST

System load:  0.73           Processes:          223
Usage of /:   71.7% of 13.56GB  Users logged in:     1
Memory usage: 25%
Swap usage:   0%

Se pueden aplicar 0 actualizaciones de forma inmediata.

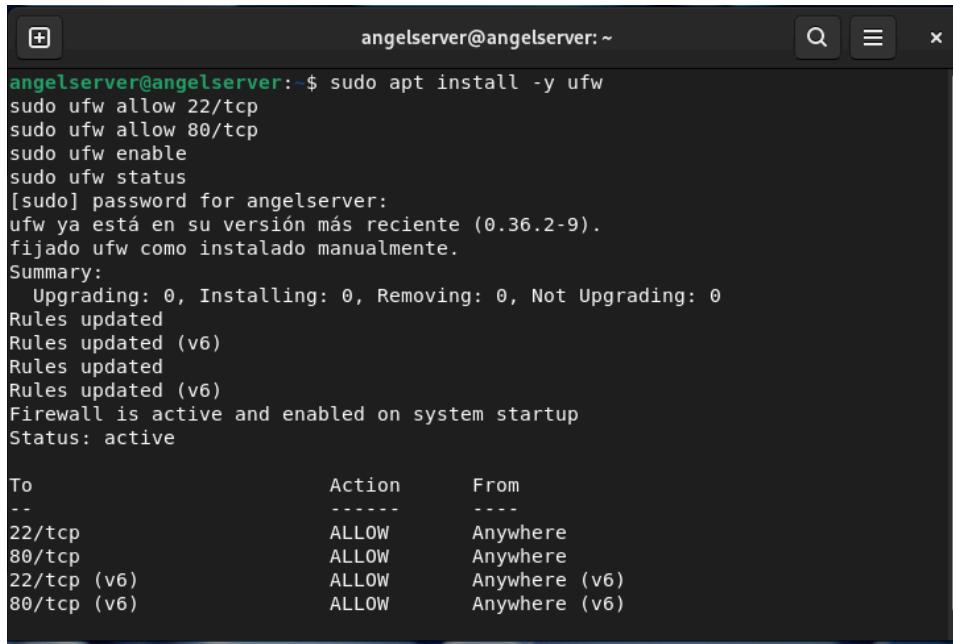
Last login: Thu Jun 19 10:08:47 2025 from 192.168.1.21
angelsrv@angelsrv:~$ 
```

## Fase 3: Seguridad mínima obligatoria

### Instalación y configuración de Firewall + Prueba

Procederemos a instalar y configurar el UFW(Firewall), para ello usaremos los comandos:

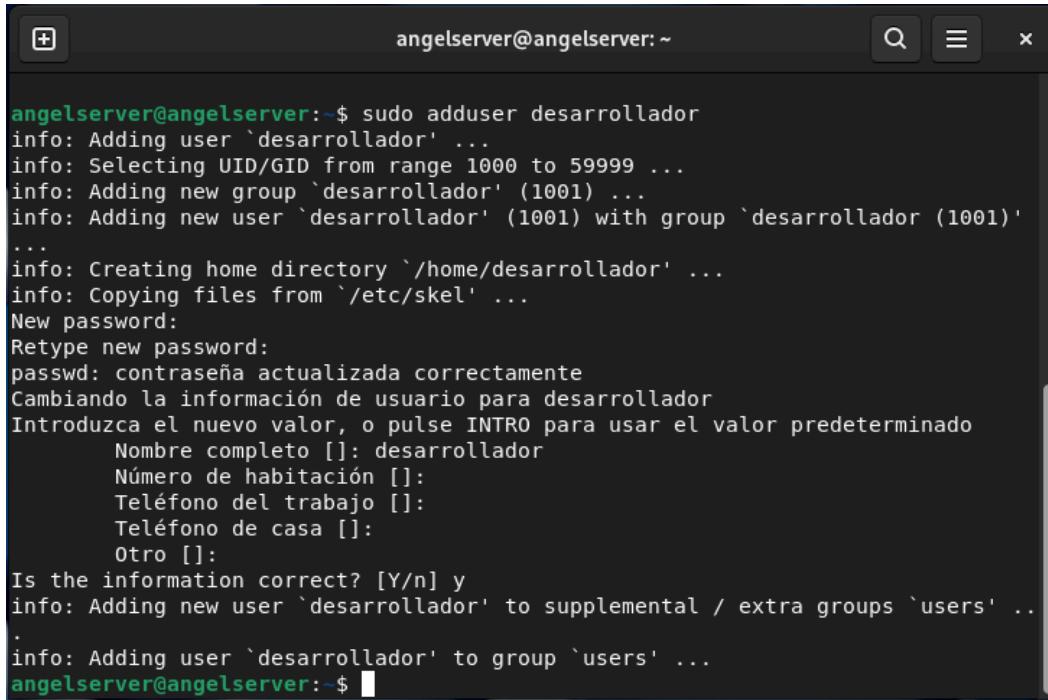
***sudo apt install ufw -y, sudo ufw allow 22/tcp, sudo ufw allow 80/tcp y sudo ufw enable***



A terminal window titled "angelserver@angelserver: ~" showing the output of the command "sudo apt install -y ufw". The output includes the installation of ufw, enabling it, and updating rules. It also shows that ufw is active and enabled on system startup. A table at the bottom lists the current firewall rules.

To	Action	From
--	-----	-----
22/tcp	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
80/tcp (v6)	ALLOW	Anywhere (v6)

Crearemos un usuario llamado “desarrollador” sin permisos de superusuario con el comando: ***sudo adduser desarrollador***.



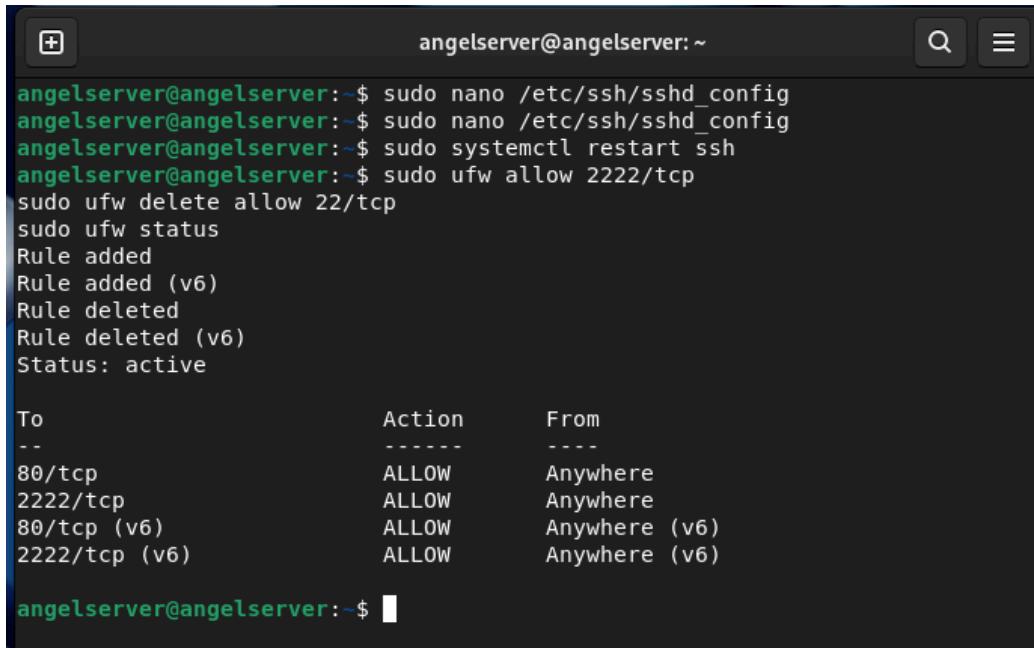
The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "angelserver@angelserver:~". The command entered is "sudo adduser desarrollador". The output shows the system adding a new user with UID 1001 and GID 1001, creating a home directory at "/home/desarrollador", and copying files from "/etc/skel". It prompts for a new password, which is confirmed by retying it. It also asks for additional user information like name, address, work phone, home phone, and other details, all of which are left blank. Finally, it asks if the information is correct, and the user responds with "y" (yes). The command concludes by adding the user to the "users" group.

```
angelserv...$ sudo adduser desarrollador
info: Adding user `desarrollador' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `desarrollador' (1001) ...
info: Adding new user `desarrollador' (1001) with group `desarrollador (1001)'
...
info: Creating home directory `/home/desarrollador' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para desarrollador
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
    Nombre completo []: desarrollador
    Número de habitación []:
    Teléfono del trabajo []:
    Teléfono de casa []:
    Otro []:
Is the information correct? [Y/n] y
info: Adding new user `desarrollador' to supplemental / extra groups `users' ...
.
info: Adding user `desarrollador' to group `users' ...
angelserv...$
```

Cambiaremos el puerto SSH de 22 a 2222 editando el archivo correspondiente con el comando: ***sudo nano /etc/ssh/sshd\_config*** y descomentamos para modificar el parámetro

Port 22 → Port 2222

Después reiniciamos el servidor SSH con: ***sudo systemctl restart ssh*** y actualizamos el UFW para puerto 2222 con : ***sudo ufw allow 2222/tcp ,sudo ufw delete allow 22/tcp*** y ***sudo ufw status***.



```

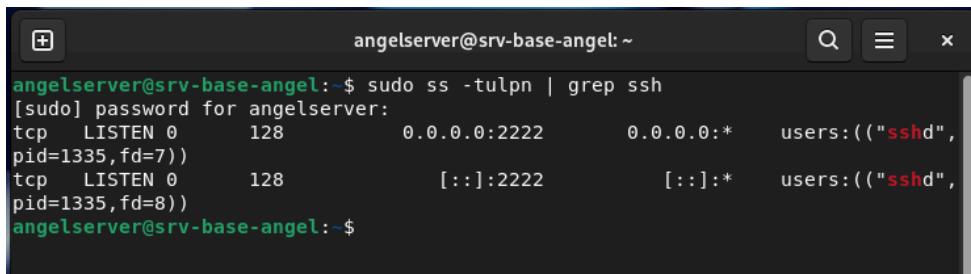
angelserver@angelserver:~$ sudo nano /etc/ssh/sshd_config
angelserver@angelserver:~$ sudo nano /etc/ssh/sshd_config
angelserver@angelserver:~$ sudo systemctl restart ssh
angelserver@angelserver:~$ sudo ufw allow 2222/tcp
sudo ufw delete allow 22/tcp
sudo ufw status
Rule added
Rule added (v6)
Rule deleted
Rule deleted (v6)
Status: active

To           Action    From
--           ----     ---
80/tcp        ALLOW     Anywhere
2222/tcp      ALLOW     Anywhere
80/tcp (v6)   ALLOW     Anywhere (v6)
2222/tcp (v6) ALLOW     Anywhere (v6)

angelserver@angelserver:~$ █

```

(En mi caso el servicio que tenía activo era `ssh.socket` y no directamente `sshd.service` por lo que seguía cogiendo el puerto 22, para solucionarlo hay que detener el servicio `ssh.socket` con el comando: **`sudo systemctl disable --now ssh.socket`**. Esto desactiva la escucha al puerto 22, intentaremos iniciar el servicio manualmente con: **`sudo systemctl start ssh.service`** y verificaremos que escucha al puerto 2222)



```

angelserver@srv-base-angel:~$ sudo ss -tulpn | grep ssh
[sudo] password for angelserver:
tcp  LISTEN  0      128          0.0.0.0:2222        0.0.0.0:*      users:(("sshd",
pid=1335,fd=7))
tcp  LISTEN  0      128          [::]:2222         [::]:*       users:(("sshd",
pid=1335,fd=8))
angelserver@srv-base-angel:~$ █

```

Para desactivar acceso SSH del usuario root volveremos a modificar y descomentar con el comando : **`sudo nano /etc/ssh/sshd_config`**

`PermitRootLogin no`

```

GNU nano 8.3                               angelserver@angelserver: ~

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

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

```

Para terminar, comprobaremos la conexión desde otro equipo de la red interna con :  
**ssh desarrollador@192.168.1.21 -p 2222.**

```

desarrollador@angelserver: ~

System information as of jue 19 jun 2025 12:14:32 CEST

System load:  0.14          Processes:            234
Usage of /:   71.8% of 13.56GB  Users logged in:       2
Memory usage: 26%
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Se pueden aplicar 0 actualizaciones de forma inmediata.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

desarrollador@angelserver:~$ 

```

## Fase 4: Estructura de carpetas y servicios iniciales

# Crear estructura de carpetas

Crearemos una estructura de carpetas organizadas para después otorgarles permisos de desarrollador y root con: ***sudo mkdir -p /srv/www***, ***sudo mkdir -p /srv/repositorios*** y ***sudo mkdir -p /srv/docs***.

## Permisos

Daremos los permisos a las carpetas de *desarrollador* y *root* con los comandos :

**sudo chown desarrollador:desarrollador /srv/www, sudo chmod 755 /srv/www, sudo chown root:root /srv/repositorios y sudo chmod 700 /srv/repositorios.**

## Instalar Apache

En este ejemplo instalaremos **Apache2** (también se puede utilizar **NGINX**), para ello usaremos en la consola: ***sudo apt install -y apache2***, ***sudo systemctl enable apache2*** y ***sudo systemctl start apache2***.

```
angelserv@angelserv:~$ sudo chown root:root /srv/repositorios
angelserv@angelserv:~$ sudo chmod 700 /srv/repositorios
angelserv@angelserv:~$ sudo apt install -y apache2
sudo systemctl enable apache2
sudo systemctl start apache2
Installing:
 apache2

Installing dependencies:
 apache2-data apache2-utils

Paquetes sugeridos:
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom

Summary:
 Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 0
 Download size: 354 kB
 Space needed: 1.684 kB / 3.345 MB available

Des:1 http://es.archive.ubuntu.com/ubuntu plucky/main amd64 apache2-data all 2.4.63-1ubuntu1 [1
 3 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu plucky/main amd64 apache2-utils amd64 2.4.63-1ubuntu1
 [100 kB]
```



## Crear y configurar página de prueba:

Crearemos una página de prueba escribiendo: ***echo "<h1>Servidor operativo - \$hostname</h1>" | sudo tee /srv/www/index.html***

Configuraremos **Apache** para que sirva /srv/www. Puedes cambiar el DocumentRoot en /etc/apache2/sites-available/000-default.conf. Escribiremos en la consola: ***sudo nano /etc/apache2/sites-available/000-default.conf***

Dentro del archivo modificaremos:

***DocumentRoot /srv/www***

Reinicia **Apache** después de modificar el archivo con el comando: ***sudo systemctl restart apache2***

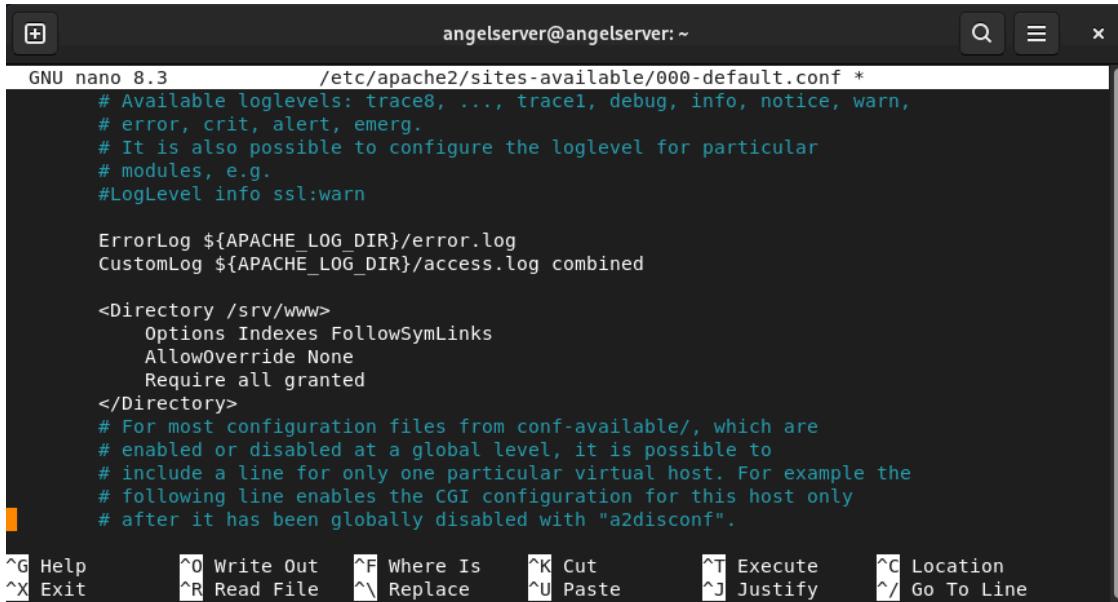
*(En mi caso faltaba el Bloque <Directory> que le diga a Apache que puede servir contenido de /srv/www, Por defecto, si no tienes un bloque <Directory /srv/www>, Apache rechaza el acceso por seguridad, y es por eso que te sale. Para solucionar este error tendremos que editar el archivo /etc/apache2/sites-available/000-default.conf como en el paso anterior y añadir el bloque <Directory> justo antes de </VirtualHost> quedandote el archivo en este orden:*

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /srv/www

    # 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

    <Directory /srv/www>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

</VirtualHost>
```



```
GNU nano 8.3          /etc/apache2/sites-available/000-default.conf *
# 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

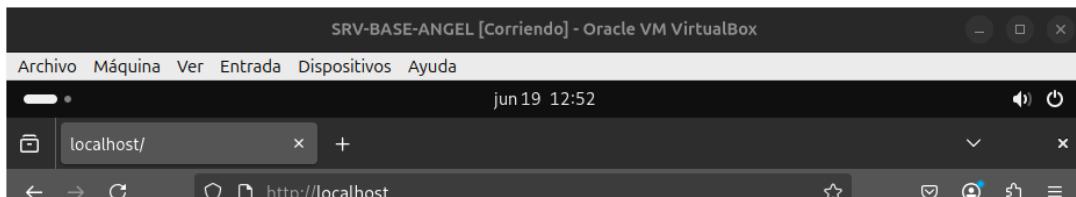
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /srv/www>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
# 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".

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

Una vez corregido el error, ahora solo falta reiniciar Apache2 con el comando: ***sudo systemctl restart apache2***.

Por último, probaremos en el navegador la dirección: <http://localhost> y comprobaremos el contenido de la página web de prueba.



**Servidor operativo - angelserver**

