

# Cloud Server Project

*Implementa un servicio cloud en opensource*



David G. & Paolo S.

---

# ubuntu<sup>®</sup>



Nextcloud



# 01

## Introducción

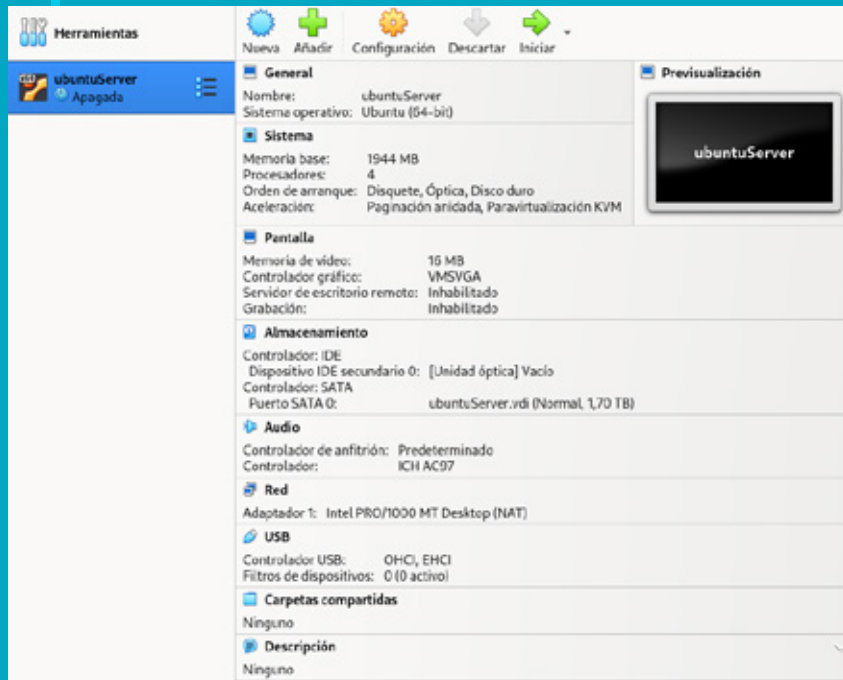
### *¿Qué vamos a ver?*

En este taller, aplicaremos los conocimientos adquiridos en los proyectos "**Born2beroot**" y "**Netpractice**" de **42** para configurar y desplegar un servicio **Nextcloud** completamente funcional, respaldado por **Ubuntu Server** y ejecutado sobre **VirtualBox**.

### *¿Qué aprenderás?*

**Configuración** de un servidor Ubuntu para soportar servicios en la nube. **Instalación y administración** de Nextcloud, una potente solución de almacenamiento y colaboración en la nube. **Implementación y manejo** de un servidor virtualizado con VirtualBox.

Aplicación de **buenas prácticas** en seguridad y redes basadas en el proyecto Born2beroot. **Gestión de redes** y servicios a nivel básico e intermedio, tomando como referencia el proyecto Netpractice.



# 02 VirtualBox

## ¿Qué es una máquina virtual?

Una **máquina virtual** (VM) es un software que **emula** un sistema informático completo, permitiendo ejecutar un **sistema operativo** (SO) dentro de otro SO de manera aislada. Por ejemplo, con **VirtualBox**, puedes instalar y ejecutar un **sistema operativo diferente** (como Linux, macOS o Windows) en tu computadora sin necesidad de reemplazar el sistema operativo principal.

Software de **código abierto** bajo la licencia **GPLv2** (GNU General Public License) para la mayor parte de su código.

## Aplicaciones de las MV

1. **Pruebas de software:** para probar aplicaciones en diferentes sistemas operativos.
2. **Entornos de desarrollo.**
3. **Seguridad:** probar software sospechoso o navegar por internet sin poner en riesgo el equipo principal.
4. **Ejecutar software incompatible.**

<https://www.virtualbox.org/>

## ¿Qué es Nextcloud?

Nextcloud, es una herramienta completa orientada a **empresas** y **particulares**, cuya función es, actuar como un servidor de almacenamiento en la **nube** de **fotos, datos, archivos**, etc.

Permite una gran **personalización** a través de la instalación de **apps** o **módulos**, que permite ampliar funcionalidades más completas, según las necesidades.

<https://nextcloud.com/es/>

# 03 NextCloud

## Open Source y Privacidad

**Nextcloud** es un software **open source**. Esto significa que su **código fuente** está disponible para el **público**, y cualquier persona puede acceder, modificar y distribuir el software de acuerdo con los términos de su licencia, que es la AGPL (GNU Affero General Public License).

<https://www.gnu.org/licenses/agpl-3.0.html>

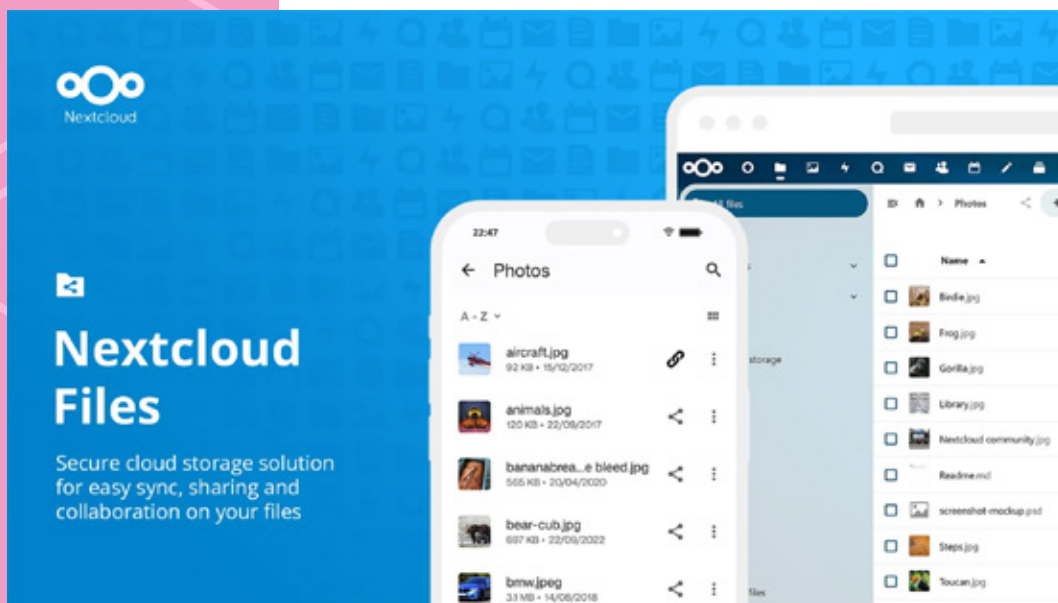




Foto de Pixabay. <https://www.pexels.com/es-es/foto/logotipo-de-seguridad-60504/>

# 04

## Privacidad

### *La importancia de la privacidad*

La **privacidad** en **Nextcloud** es fundamental porque te permite tener **control total** sobre tus datos.

A diferencia de los servicios en la nube comerciales, donde **tus archivos se almacenan en servidores de terceros**, con Nextcloud puedes hospedar **tu propia nube** en tus **propios servidores**.

Esto significa que:

- Tú decides quién tiene **acceso** a tus **datos**.
- **No** dependes de grandes empresas que pueden **compartir** o **utilizar** tu información para fines **comerciales**.
- **Cumples** con las normativas de protección de datos, como el RGPD, garantizando que la información de tu empresa o clientes esté **siempre segura y privada**.
- Es ideal para empresas que buscan proteger su **información sensible** sin comprometerla al almacenarla en servicios externos.

# 05

## Ubuntu Server

### ¿Qué es Ubuntu Server?

**Ubuntu Server** es una versión del sistema operativo **Ubuntu** diseñada específicamente para **servidores**.

Ofrece **estabilidad, seguridad y rendimiento** para administrar y ejecutar **aplicaciones y servicios** en la **nube, bases de datos, web hosting, y virtualización**.

Aplicaciones comunes:

- **Servidores web:** Apache, Nginx.
- **Bases de datos:** MySQL, PostgreSQL.
- **Correo electrónico**
- **Virtualización:** KVM, LXC.
- **Servicios en la nube:** Nextcloud, OpenStack.
- **Contenedores:** Docker, Kubernetes

Es ideal para empresas y particulares que necesitan un servidor **flexible, seguro** y de **código abierto**.

<https://ubuntu.com/download/server>

ubuntu<sup>®</sup>

Resumen de los recursos del host (ordenador físico):

RAM total: Al menos 8 GB para un rendimiento óptimo

(4 GB para el host y 4 GB para la VM).

Espacio en disco: Al menos 50 GB de espacio en disco en el host.

CPU: Procesador de 64 bits con soporte para virtualización.

# 06

## Requisitos

## Ordenador

Para instalar Ubuntu Server corriendo en Virtual-Box y luego ejecutar Nextcloud en el servidor, los requisitos mínimos del ordenador deberían ser los siguientes:

### Descargar e Instalar VirtualBox

- Instalar el software descargado desde la web  
<https://www.virtualbox.org/>
- Instalar el software desde el Terminal:

```
sudo dpkg -i virtualbox-7.0_7.0.16-162802-Debian-bookworm_amd64.deb
```

### Descargar Ubuntu Server

Descargar desde la web

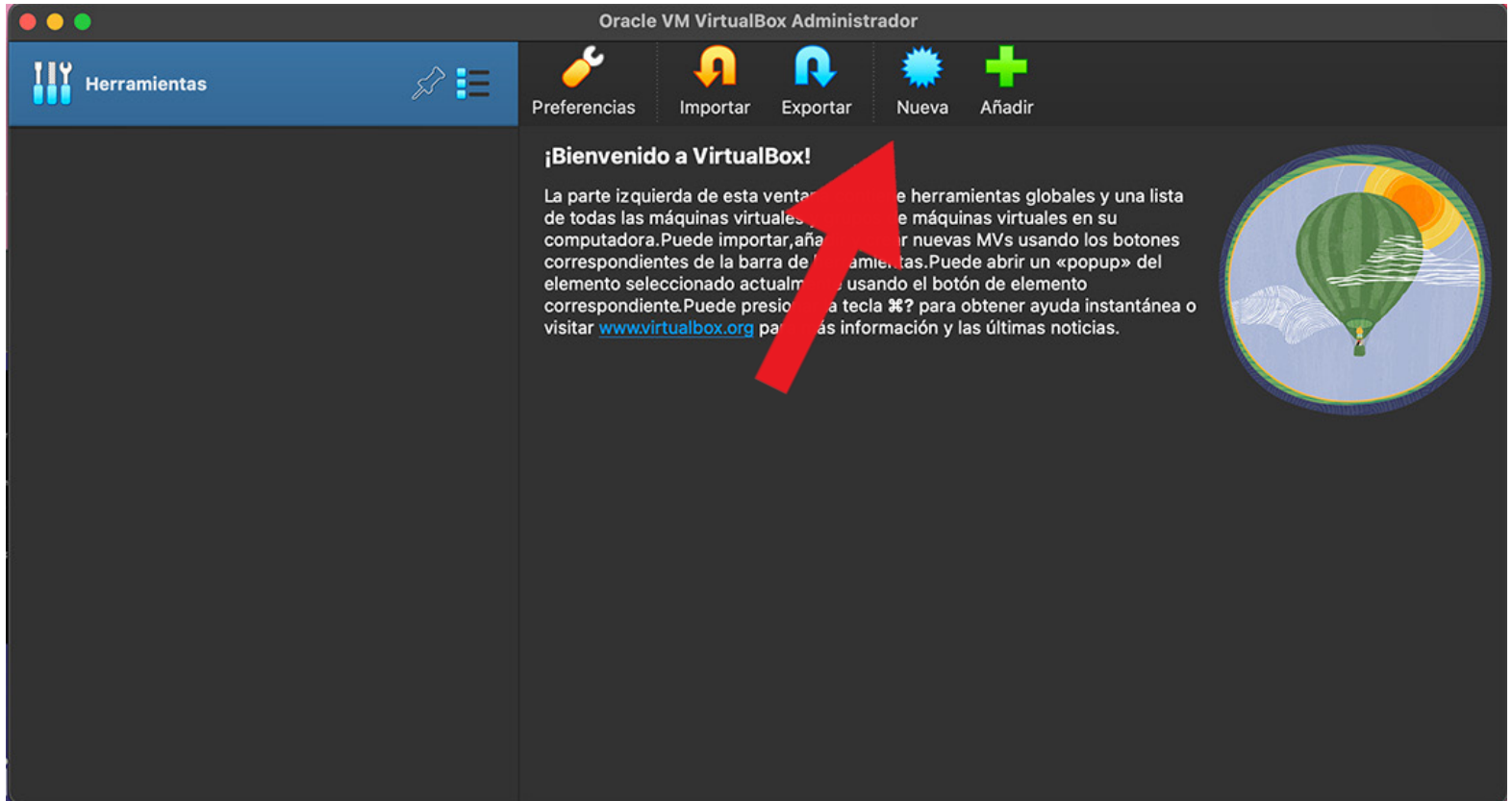
<https://www.virtualbox.org/>

### Descargar Nextcloud

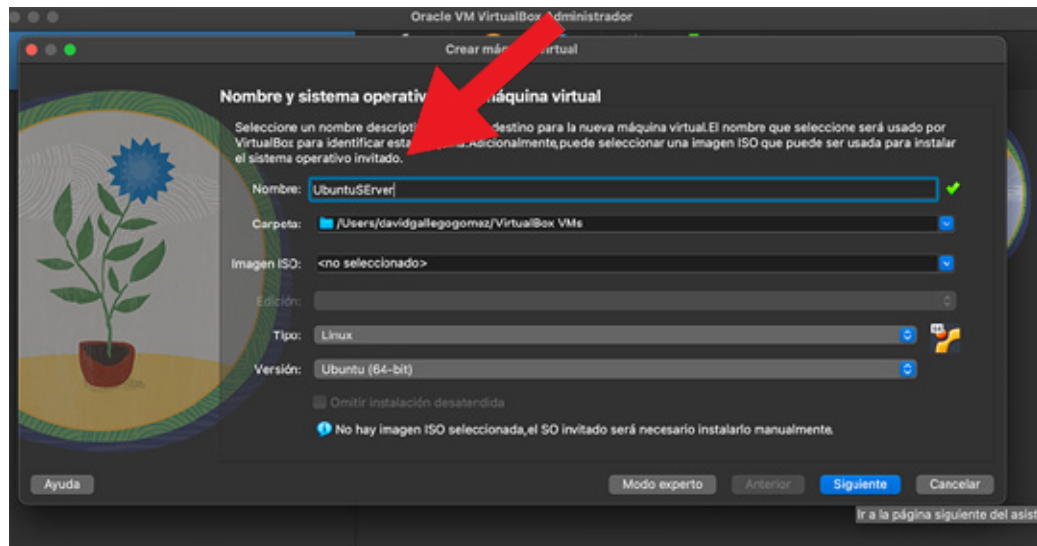
No es necesario puesto que se puede instalar desde el instalador de Ubuntu Server, pero si se prefiere, se puede descargar a través de la web o desde el mismo Terminal.

<https://nextcloud.com/es/>

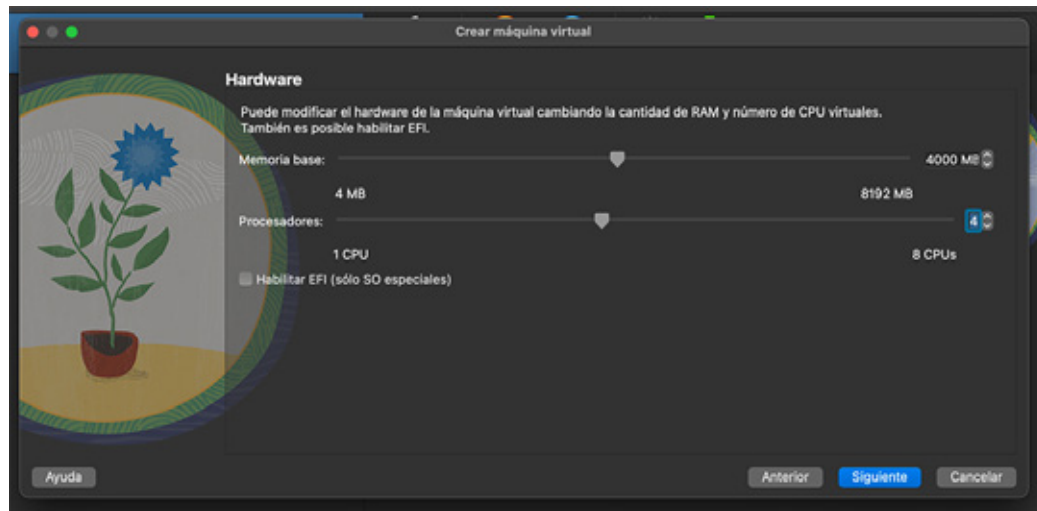
## 1. Abrimos VirtualBox y hacemos clic en nueva





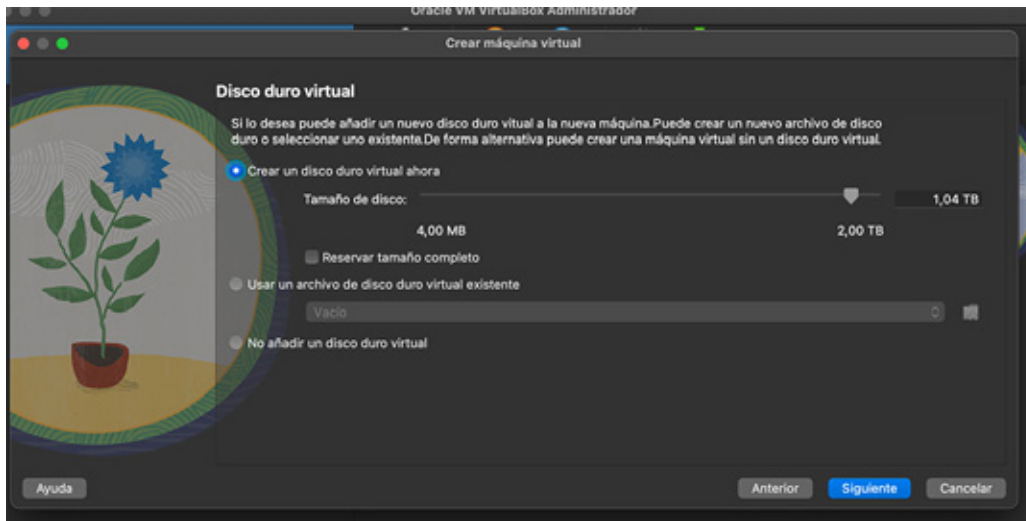


1. Ponemos nombre a la nueva máquina virtual.



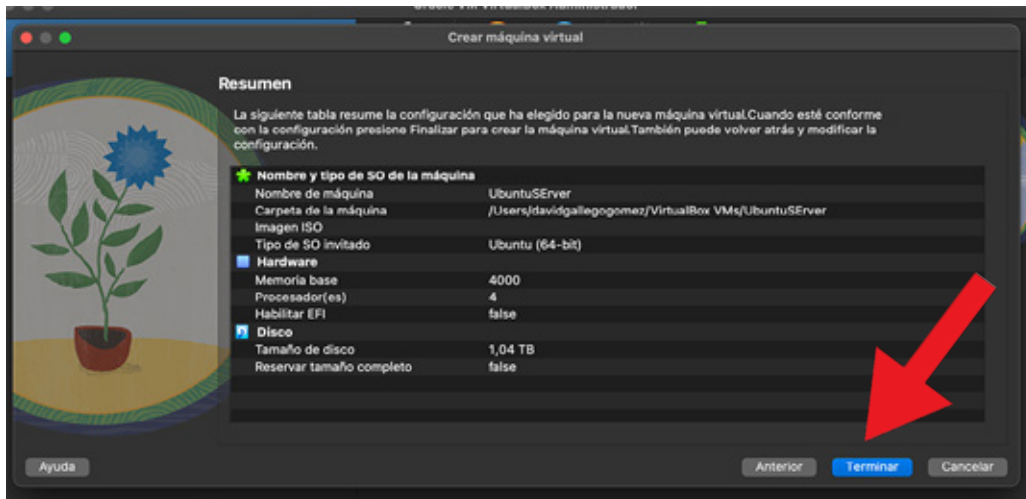
2. Seleccionamos la cantidad de RAM el número de CPU.

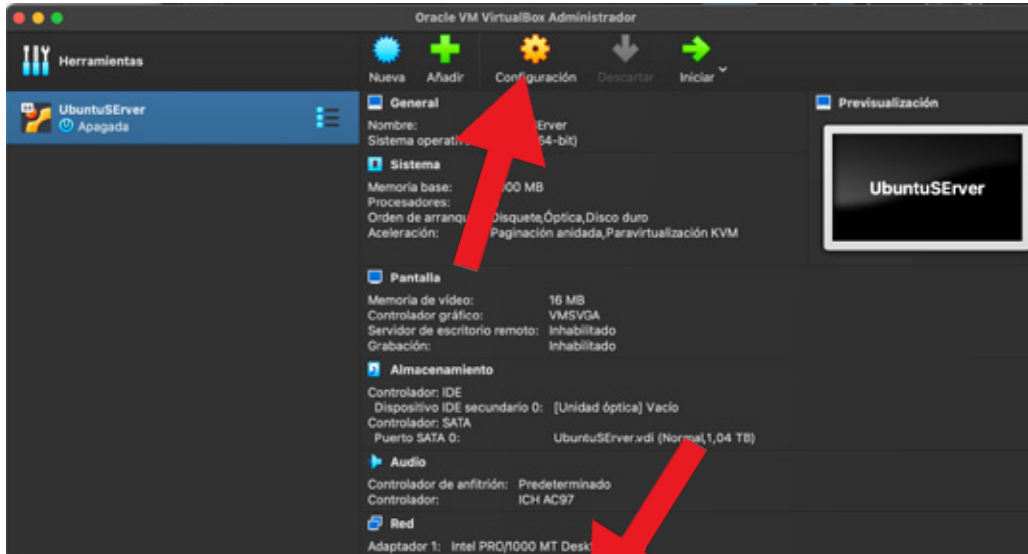
En este caso 4Gb de RAM y 4 CPU.



3. Creamos el disco duro y le damos la capacidad que creamos conveniente.

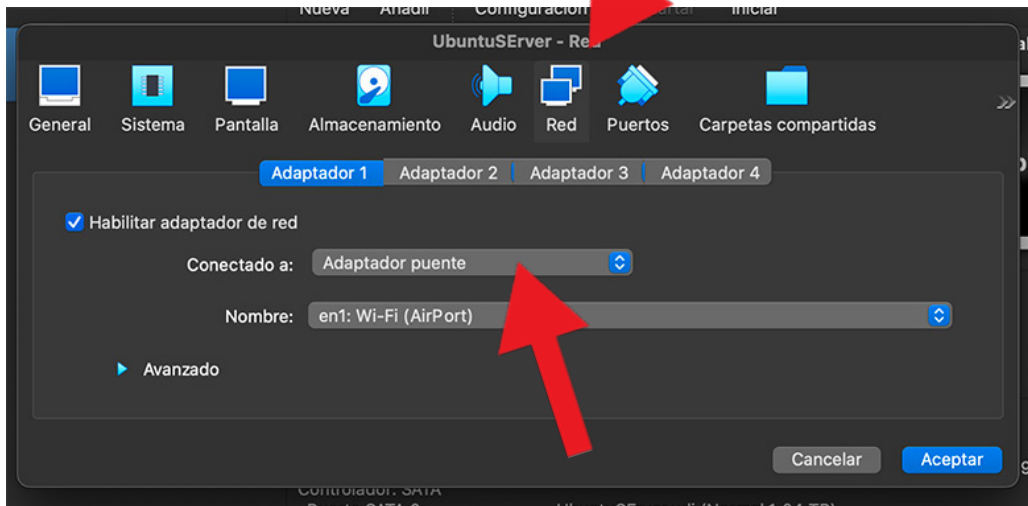
Para este caso 1Tb





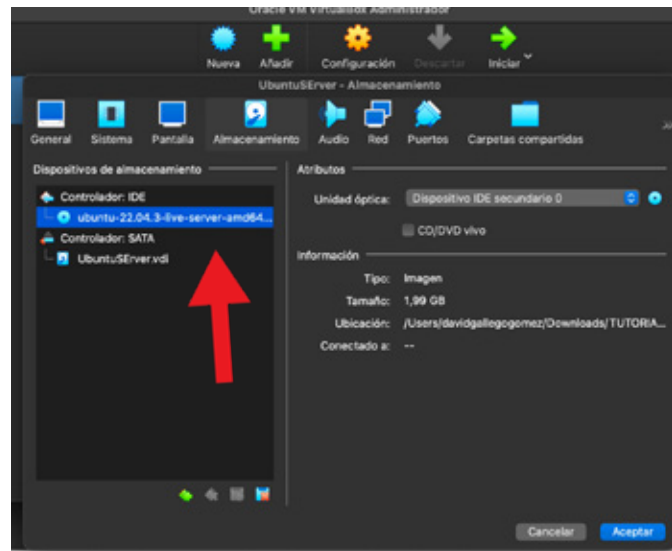
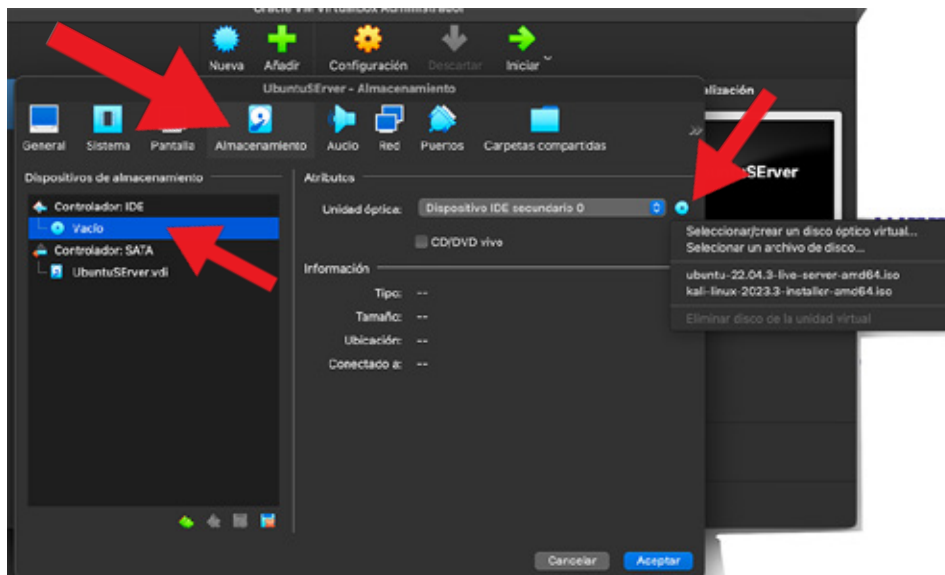
5. Ya tenemos máquina creada y ahora debemos de configurarla para instalar el Servidor.

Acemos clic en Configuración.



6. Para poder tener conexión a internet:

1. Red.
2. Adaptador puente.



## 7. Hay que cargar el controlador:

1. Almacenamiento

2. Seleccionamos -> Vacío

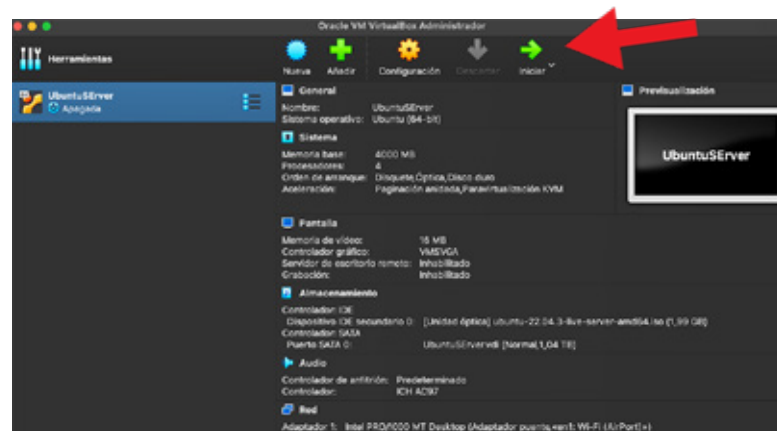
3. Clic en el "cd"

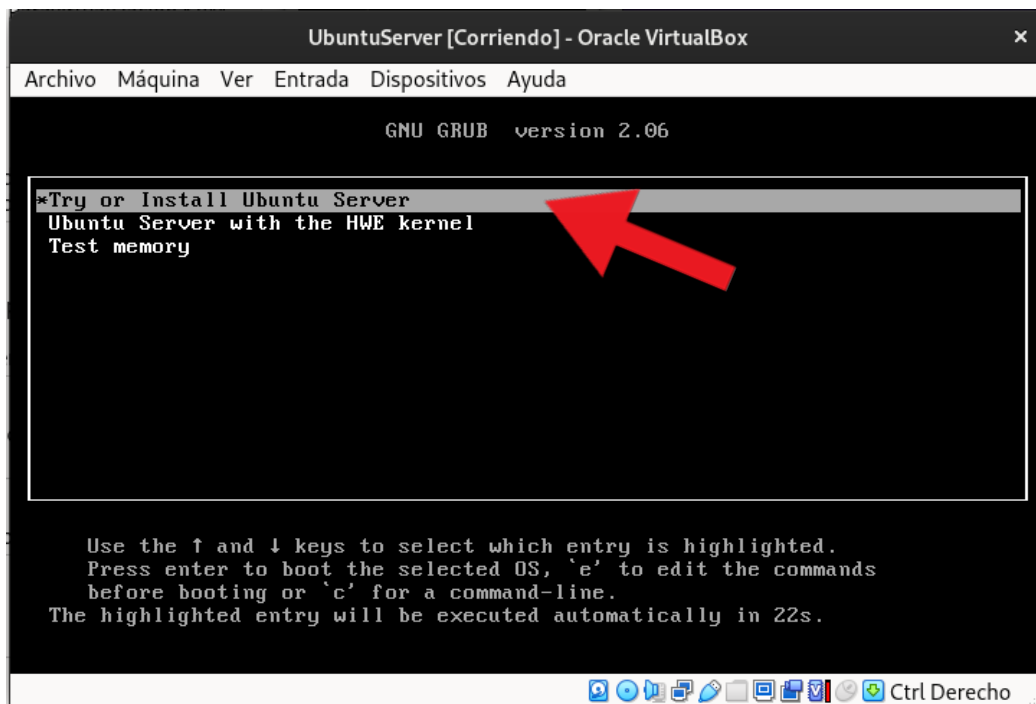
4. Cargamos la imagen .iso de Ubuntu Server.

5. Aceptar

8. Ya tenemos nuestro controlador cargado y aceptamos.

9. Pulsamos Iniciar y comenzará la instalación del servidor.

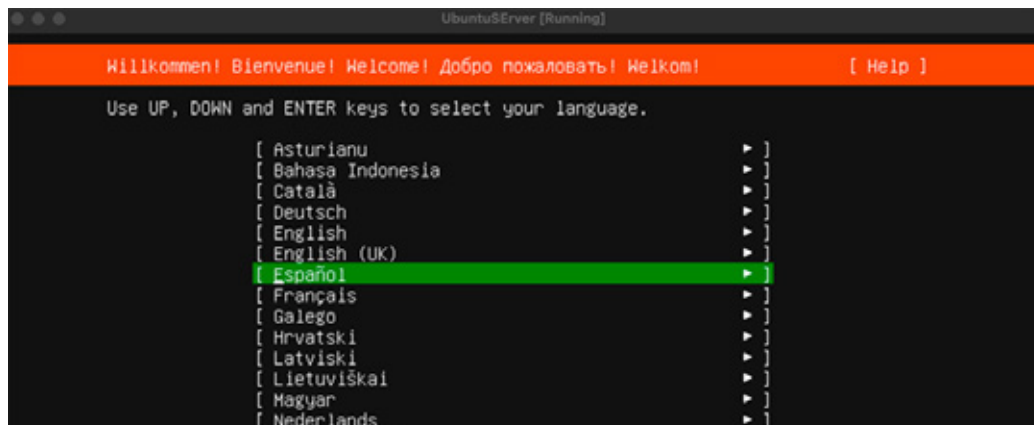




## 10. Install Ubuntu Server.

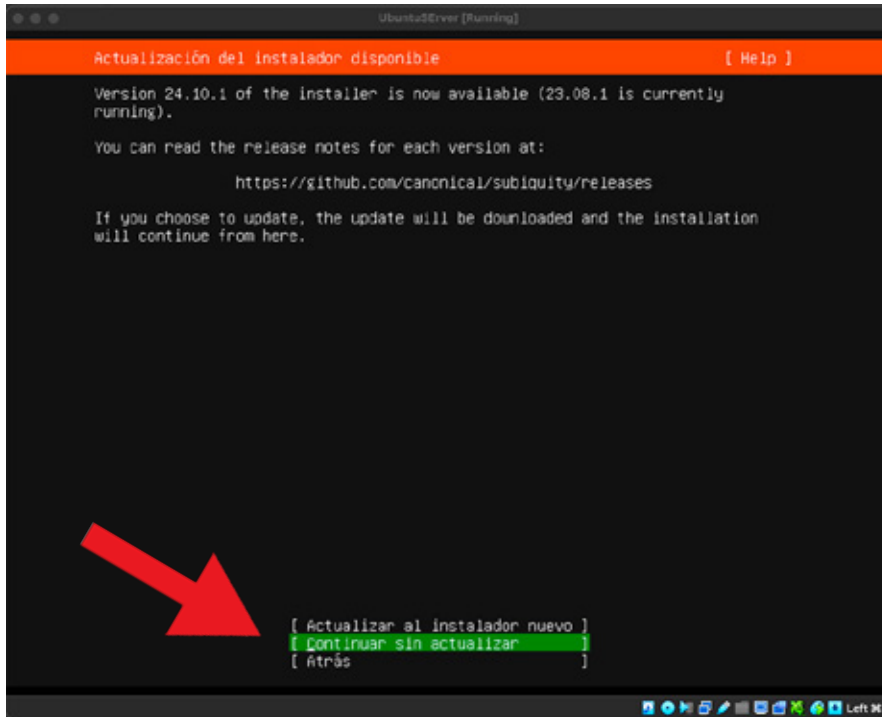
El tiempo de instalación del Servidor dependerá tanto del ordenador como de los paquetes que queramos instalar.

Esto rondará los 40 / 45 minutos.



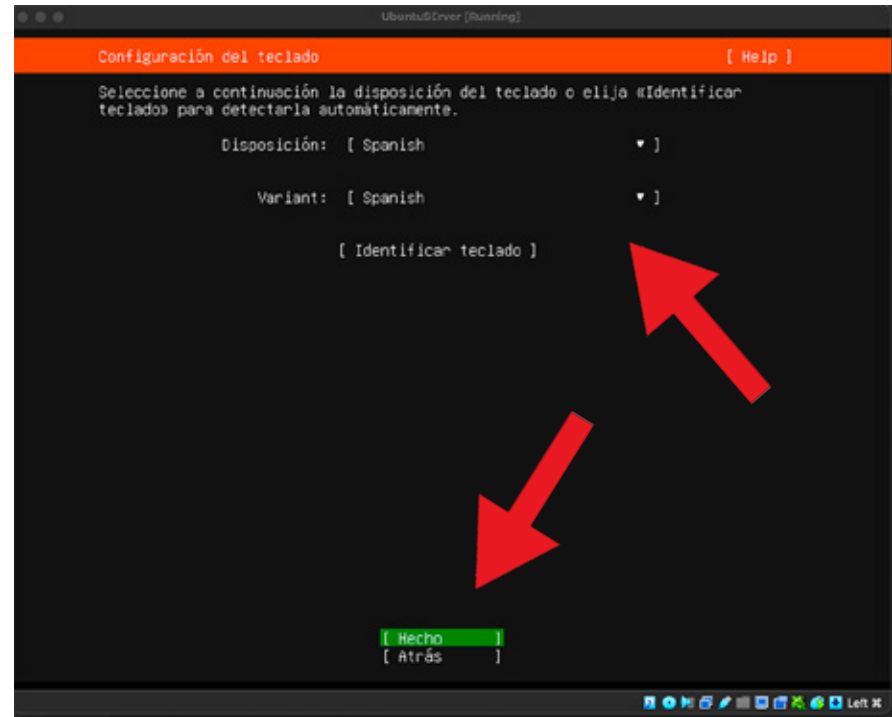
## 11. Seleccionamos el idioma.

12.



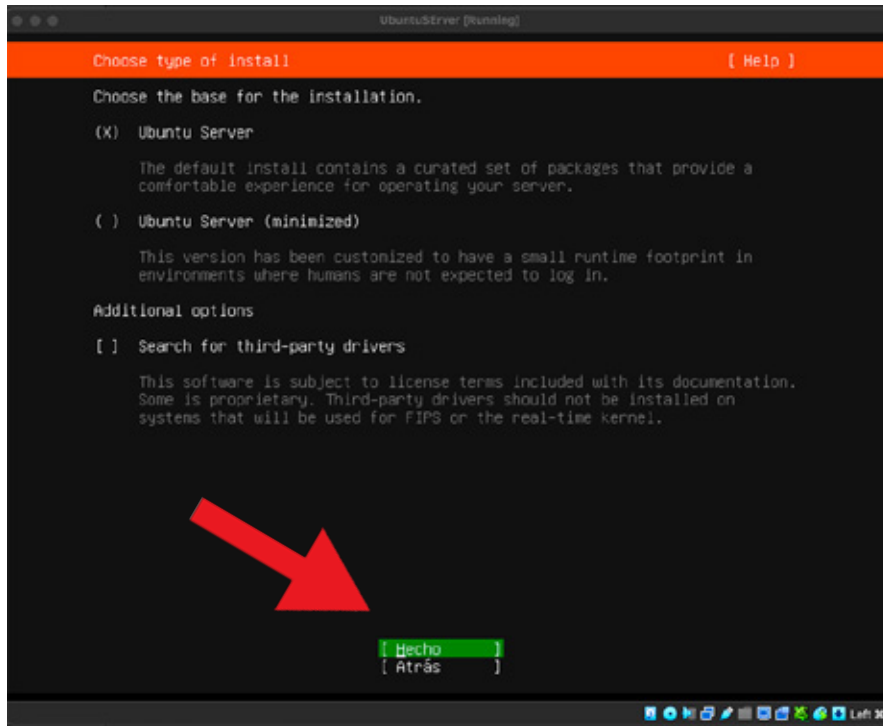
**No actualizamos al nuevo.  
Esta versión es muy estable**

13.



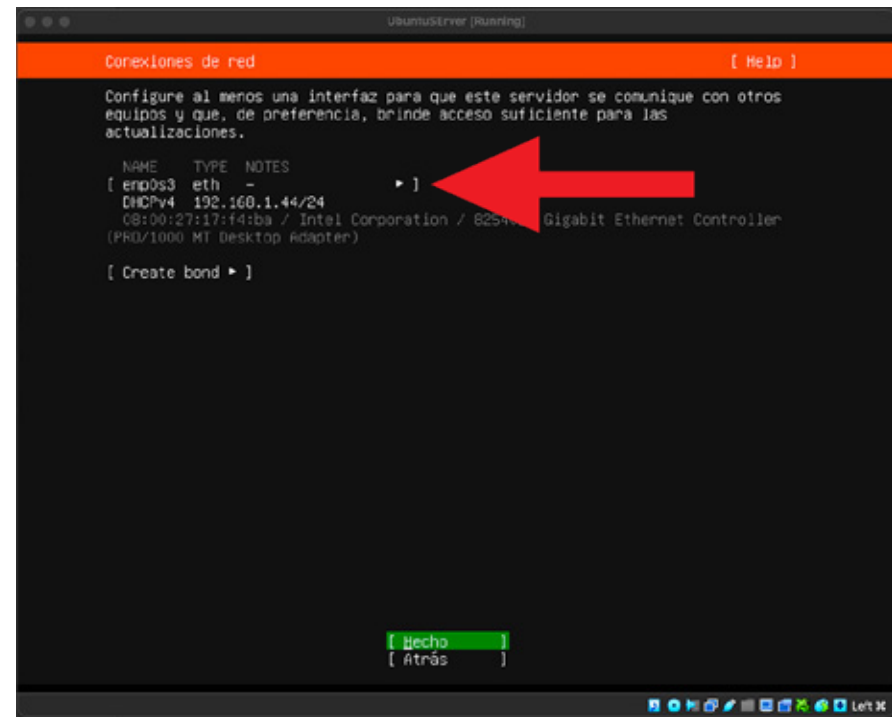
**Configuramos el idioma del teclado  
aunque Ubuntu lo hace automática-  
mente.**

14.



**Dejamos marcado Ubuntu Server**

15.



**Ahora es cuando vamos a configurar las conexiones de red del servidor. Hacemos clic en [ enp0s3 eth - > ]**

## ¿Qué es una red informática?

Una **red informática** es un conjunto de dispositivos; **ordenadores, servidores, impresoras**, etc. **conectados** entre sí.

Pueden **compartir** datos y recursos; como archivos o internet de manera eficiente. Estos dispositivos se **comunican mediante protocolos** y utilizan diferentes medios, como cables o conexiones inalámbricas, para transmitir información.

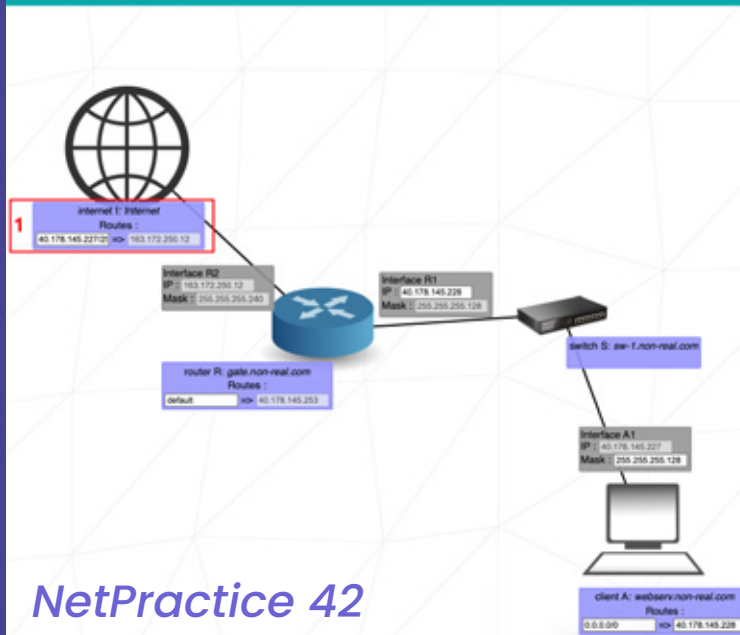
### Elementos clave de una red

- **Dispositivos:** Son los que forman la red, ordenadores, servidores, routers, etc.
- **Medios de transmisión:** Los canales a través de los cuales **viajan los datos**, ya sea mediante cables (redes cableadas) o señales inalámbricas (redes Wi-Fi).
- **Protocolos de comunicación:** Conjuntos de reglas que determinan cómo se envían y reciben los datos, como **TCP/IP**, que es el más común.
- **Dirección IP:** Cada dispositivo en la red tiene una **dirección IP** que lo identifica de manera **única** dentro de la **red**.

Level 6 :

Goal 1 : webserver-non-real.com need to communicate with Internet - Status : OK - Congratulations !!

Check again Get my config Next level



NetPractice 42

# 07

## Conexiones de Red



## ¿Qué significa TCP?

TCP significa **Protocolo de Control de Transmisión**. Es un estándar de comunicaciones que permite a programas y dispositivos **intercambiar mensajes** a través de una **red**. Se utiliza para enviar **paquetes** a través de **Internet**.

TCP **garantiza** la integridad de los datos. Antes de transmitir datos, TCP establece una **conexión** entre una **fuentes** y su **destino**, y permanece activa hasta que comienza la comunicación.

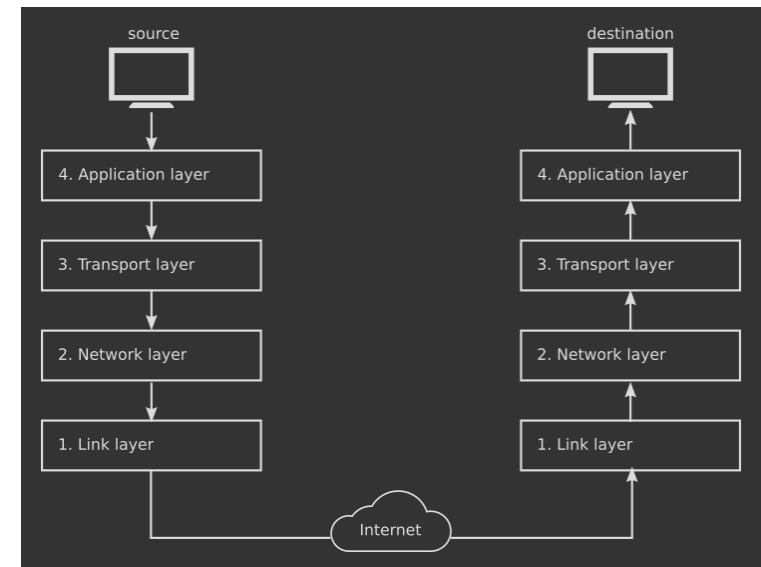
# 08

## Protocolos TCP/IP

## Protocolos IPv4 vs. IPv6

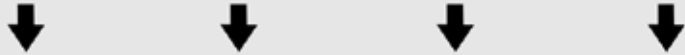
El **Protocolo de Internet** versión 4 (**IPv4**) define una **dirección IP** como un **número de 32 bits**.

Sin embargo, debido al crecimiento de Internet y al agotamiento de las direcciones IPv4 disponibles, en 1998 se estandarizó una nueva versión de IP (**IPv6**), que utiliza **128 bits** para la dirección IP.



Dirección IPv4 en notación punto-decimal

**172 . 16 . 254 . 1**



10101100.00010000.11111110.00000001



8 bits



32 bits (4 bytes)

# 09

## Configuración

### *IP y máscara subred*

En nuestro entorno del Servidor necesitamos la **IP** para que dispositivos fuera de tu red local **puedan acceder** a tus servicios, como un servidor web o correo electrónico.

**La máscara de subred** es una dirección de 32 bits que define qué **parte** de una **IP corresponde a la red** y qué parte corresponde a los **dispositivos** (hosts) dentro de esa red.

IP: 104.198.241.125

Máscara de subred: 255.255.255.128

*Esto indica que los primeros 25 bits (de 32) son para la red y los restantes para los hosts.*



## Explicación Desglosada

La máscara de subred, en este caso, es **255.255.255.128**, que en binario sería:

```
11111111.11111111.11111111.10000000
```

Los primeros **25 bits** (11111111.11111111.11111111.1) están reservados para la **red**.

Los últimos **7 bits** (0000000) están reservados para los **hosts**.

Si tomamos la dirección **IP 104.198.241.125** y la convertimos en binario, se ve así:

```
01101000.11000110.11110001.01111101
```

Cada grupo de 8 dígitos binarios (octetos) corresponde **a una parte** de la **dirección IP**.

# 10 Cálculos

## Explicación Desglosada

La máscara de subred, en este caso, es **255.255.255.128**, que en binario sería:

```
11111111.11111111.11111111.10000000
```

Los primeros **25 bits** (11111111.11111111.11111111.1) están reservados para la **red**.

Los últimos **7 bits** (0000000) están reservados para los **hosts**.

Si tomamos la dirección **IP 104.198.241.125** y la convertimos en binario, se ve así:

```
01101000.11000110.11110001.01111101
```

Cada grupo de 8 dígitos binarios (octetos) corresponde **a una parte** de la **dirección IP**.

# 10 Cálculos

## Aplicación de la Máscara Subred

Cuando aplicamos la máscara de **subred 255.255.255.128** a la **IP 104.198.241.125**, obtenemos lo siguiente:

```
IP           : 01101000.11000110.11110001.01111101
Máscara      : 11111111.11111111.11111111.10000000
-----
Dirección Red : 01101000.11000110.11110001.00000000
```

Se aplica la puerta lógica **AND**

(su salida es 1 solo si ambos bits son 1. Si alguno de los bits es 0, la salida es 0)

La dirección de red es **104.198.241.0**, que **no** puede ser utilizada por ningún dispositivo.

## Cálculo rango del host.

Direcciones de **hosts disponibles** en esta red:

**Primer** Host Disponible: **104.198.241.1**

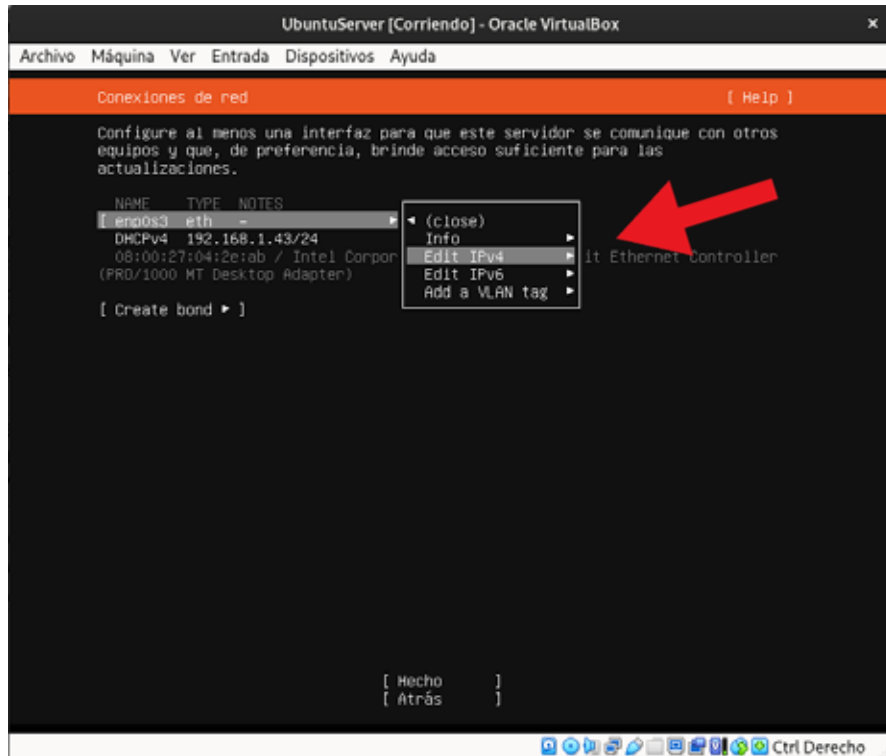
**Último** Host Disponible: **104.198.241.126**

Los extremos de la red reservados para usos específicos:

**104.198.241.0** es la dirección de **red**.

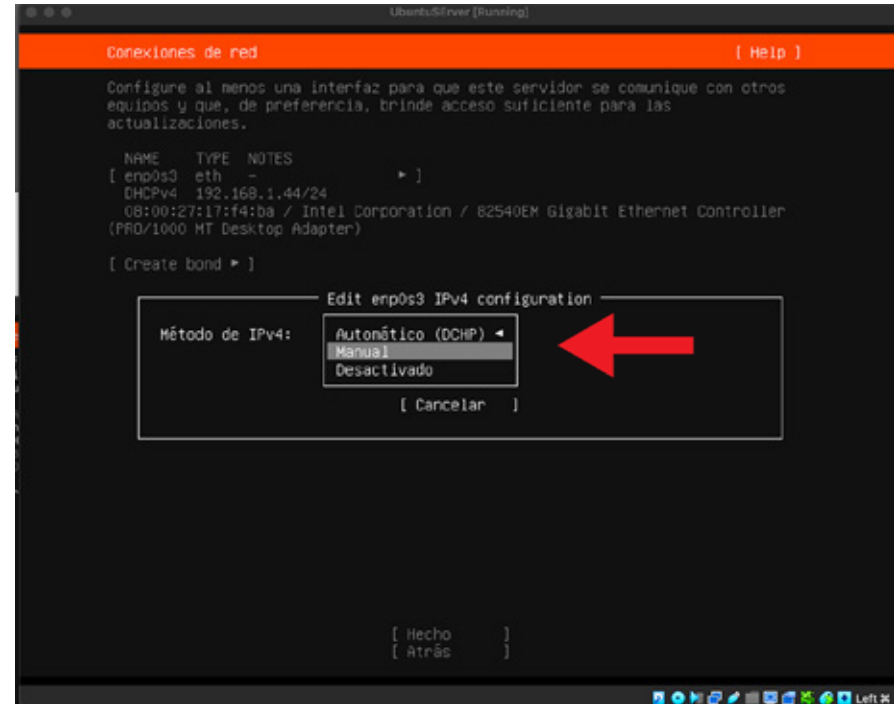
**104.198.241.127** es la dirección de **broadcast** (utilizada para enviar paquetes a todos los dispositivos en la red).

16.



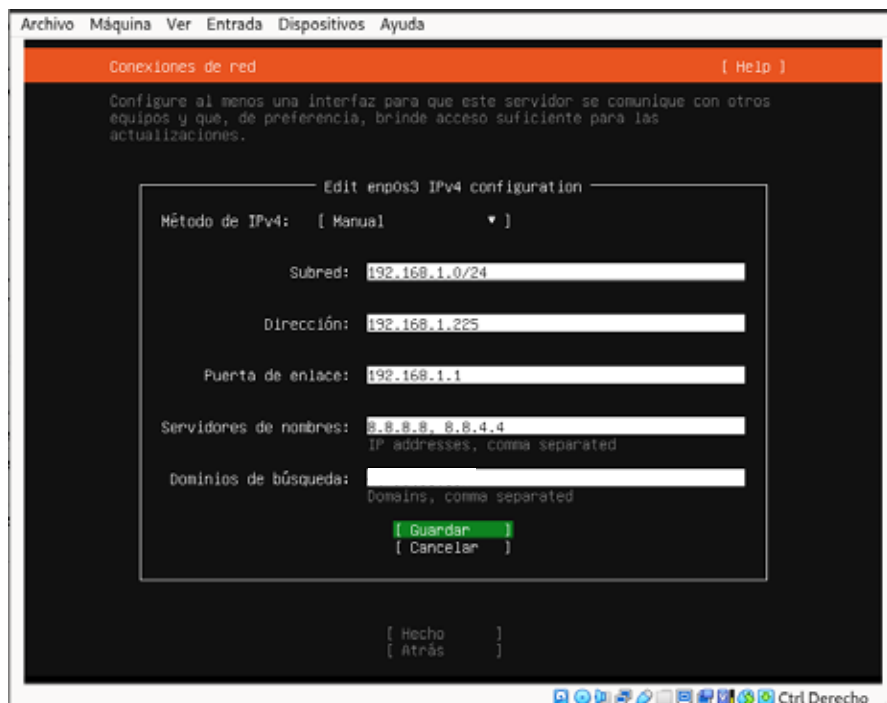
**Creamos una IP estática usando e  
protocolo de 32bits  
IPv4**

17.



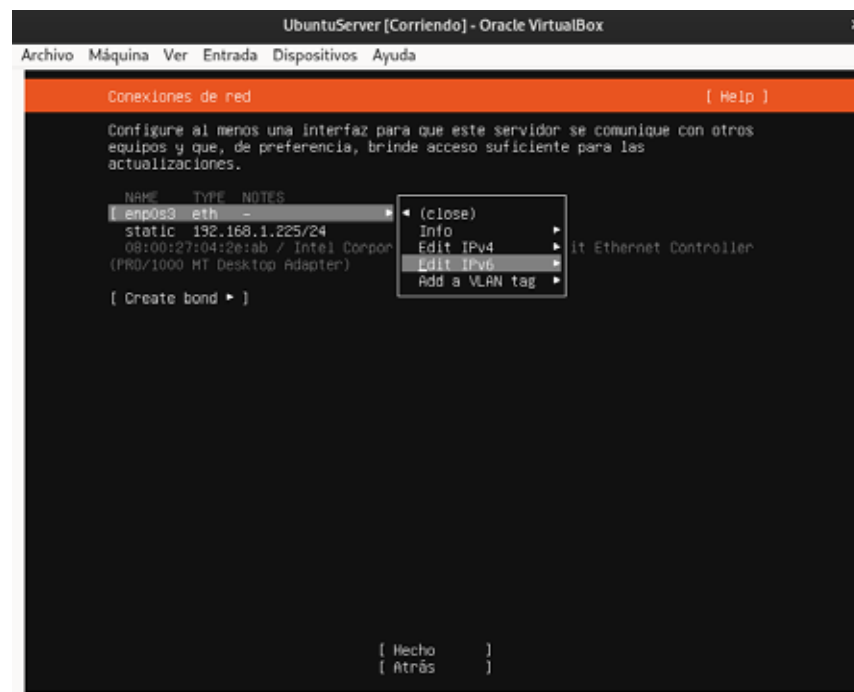
**Seleccionamos la configuración  
manual.**

18.



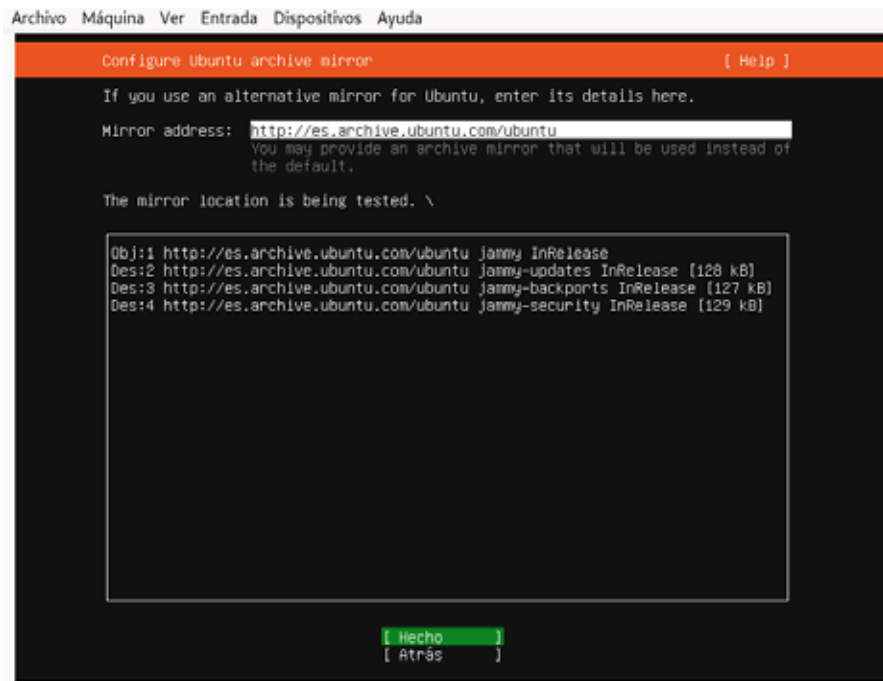
**Subred:** 192.168.1.0/24  
**IP:** 192.168.1.42 *Rango del 1 al 254 usando prefijo 24*  
**Puerta de enlace:** 192.168.1.1 *La IP del Router*  
**Servidores:** 8.8.8.8, 8.8.4.4 *Servidores de Google y Yahoo*  
**Dominios:** *No es necesario.*

19.



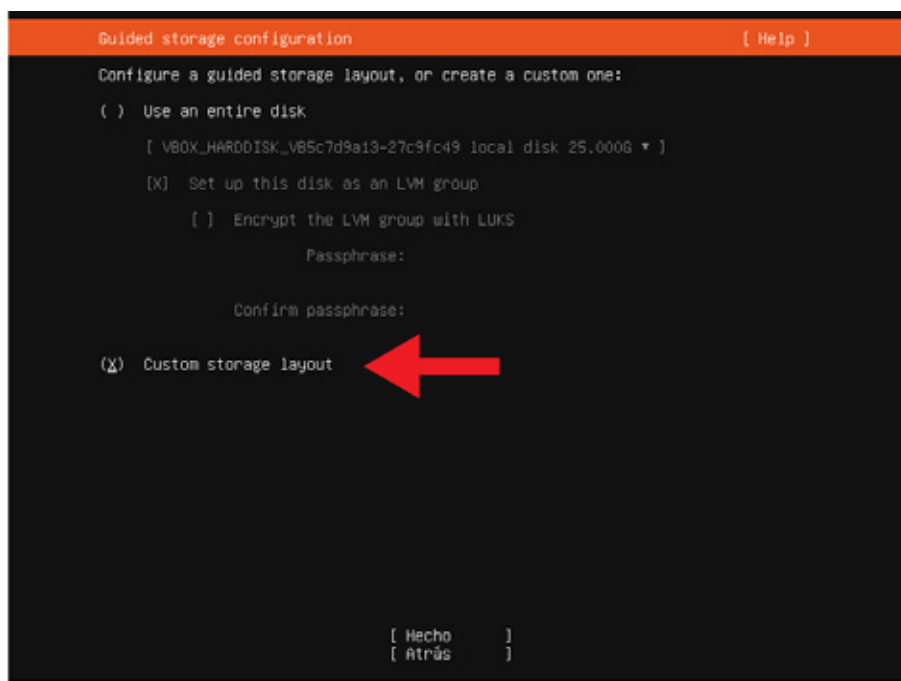
**Protocolo IPv6 de 128bits**  
**lo dejamos en automático.**  
**Esperamos a que se apliquen los**  
**cambios y continuamos**

20.



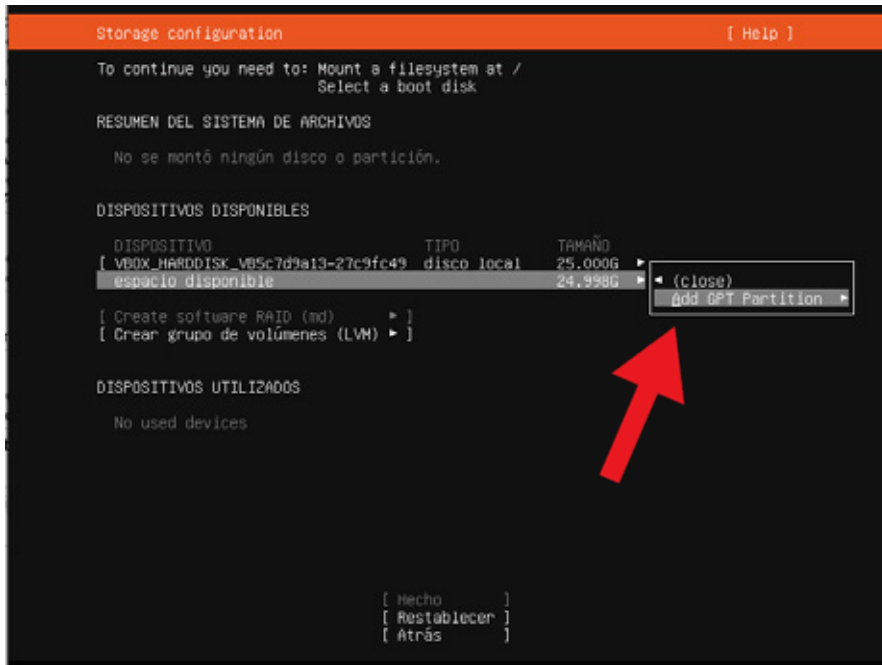
**Esperamos a que termine el test de localización de Ubuntu.**

21.



**Configuramos manualmente la partición del disco duro.**

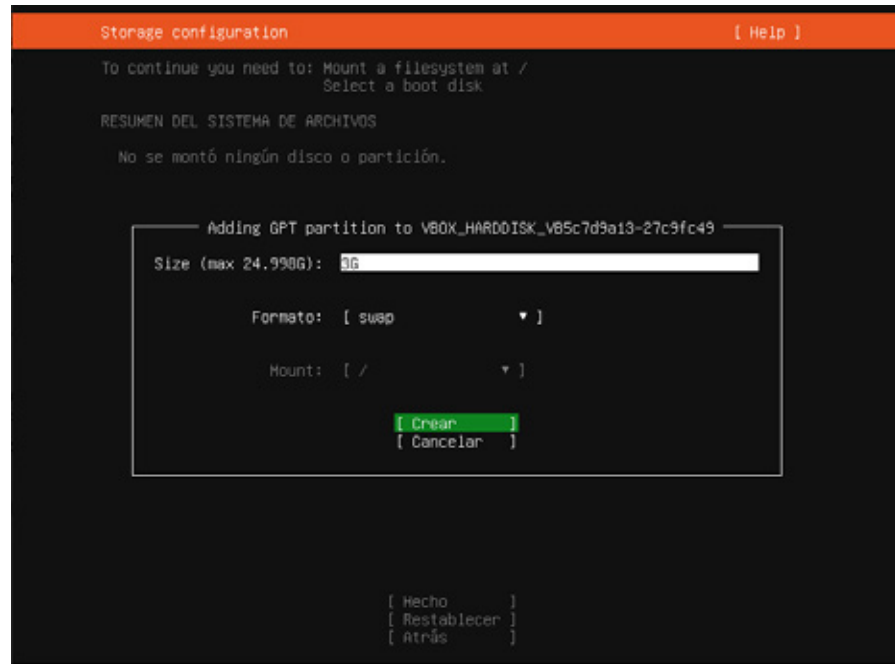
22.



**Elegimos la parte de espacio disponible que es la hemos configurado en nuestra MV.**

***espacio vacio => Add GPT Partition***

23.



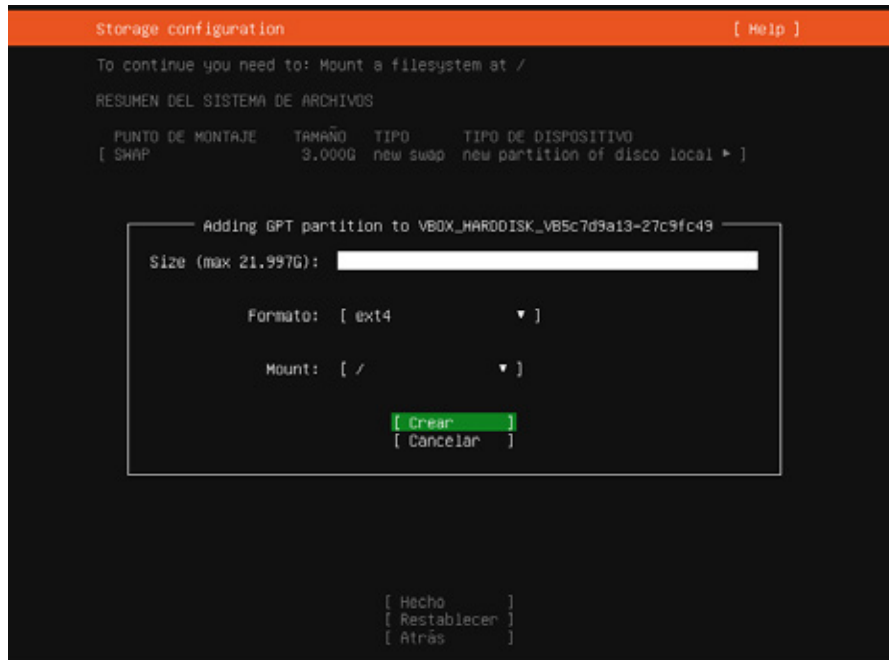
**Creamos la partición SWAP con un tamaño algo menor o igual a la RAM que hemos configurado en la MV.**

**4G => 3G**

La partición swap es un espacio en disco que se usa como "memoria virtual" cuando la RAM se llena. Ayuda a evitar bloqueos temporales al mover temporalmente datos de la memoria RAM menos usados al disco



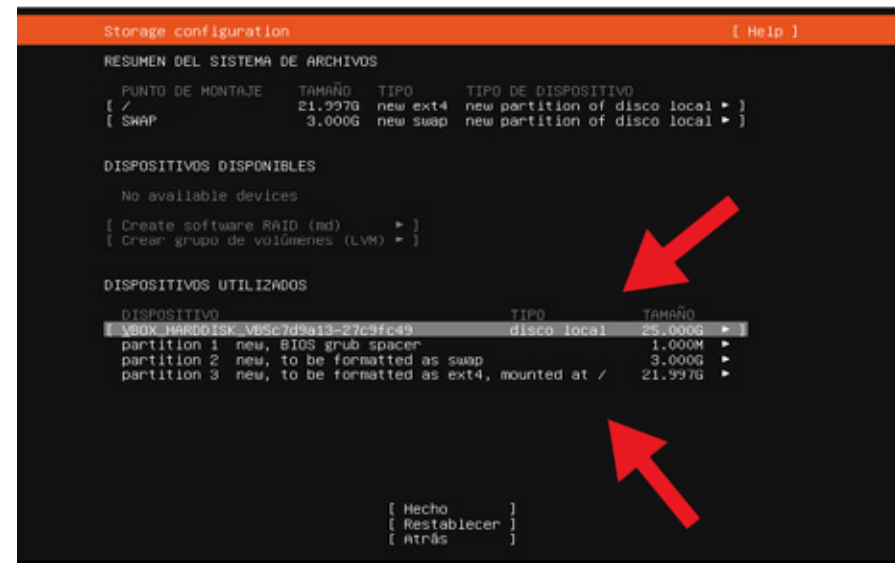
26.



**Creamos otra partición del resto del disco y se Mount de deja en '/' que es la raíz.**

*Size en blanco para max.*

27.



**Vemos las 3 particiones que tiene nuestro servidor.**

28.

Configuración de perfil [ Help ]

Proporcione el nombre de usuario y la contraseña que utilizará para acceder al sistema. Puede configurar el acceso SSH en la pantalla siguiente, pero aun se necesita una contraseña para sudo.

Su nombre:

El nombre del servidor:

The name it uses when it talks to other computers.

Elija un nombre de usuario:

Elija una contraseña:

Confirme la contraseña:

[ Hecho ]

## Configuración de Usuario Admin

Nombre: xxxxx xxx  
 Nombre del servidor: xxxxxx  
 Nombre Usuario xxxxxx  
 Contraseña: xxxxxxxx  
 Rep. Contraseña: xxxxxxxx

29.

Upgrade to Ubuntu Pro [ Help ]

Upgrade this machine to Ubuntu Pro for security updates on a much wider range of packages, until 2032. Assists with FedRAMP, FIPS, STIG, HIPAA and other compliance or hardening requirements.

[ About Ubuntu Pro ► ]

( ) Enable Ubuntu Pro

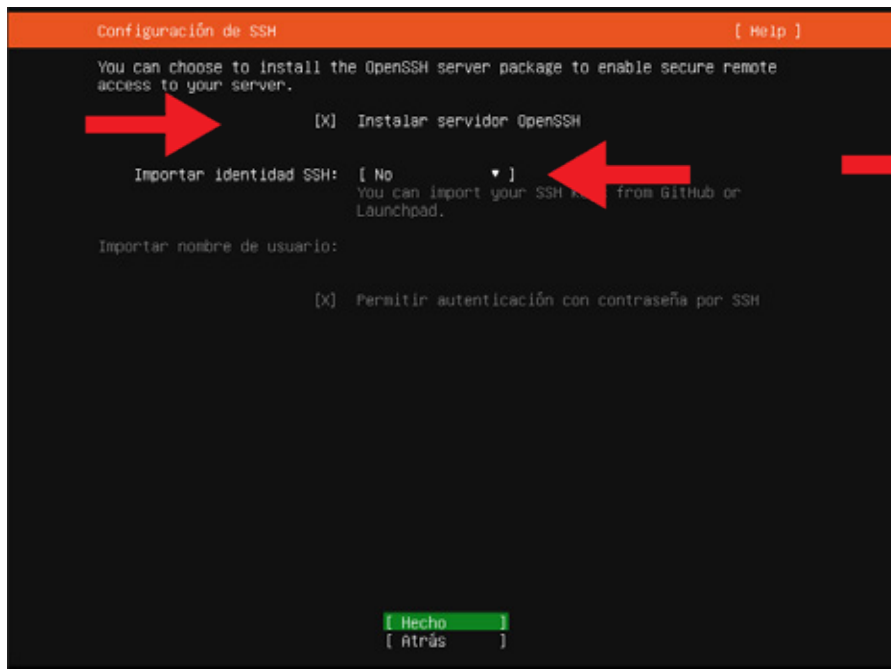
(X) Skip for now

You can always enable Ubuntu Pro later via the 'pro attach' command.

[ Continuar ]  
[ Atrás ]

No vamos a usar la versión Pro  
 seleccionamos [x]Skip for now

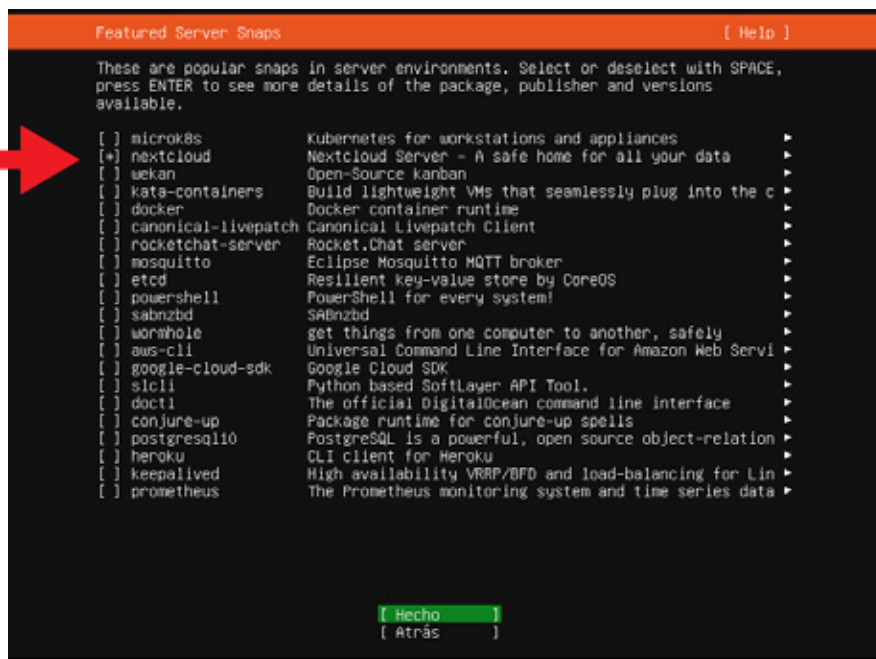
30.



## Instalamos SSH Server sin importar una identidad SSH

El servicio SSH (Secure Shell) permite conectarse de forma segura a un servidor o dispositivo remoto a través de una red, utilizando cifrado para proteger la comunicación.

31.



## Seleccionamos el servicio que vamos a instalar en nuestro servidor.

En nuestro caso es:  
**NextCloud**

32.


```

Ha finalizado la instalación. [ Help ]

running 'curtin curthooks'
curtin command curthooks
  configuring apt
  configuring apt
  installing missing packages
  configuring iscsi service
  configuring raid (mdadm) service
  installing kernel
  setting up swap
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating initramfs configuration
  configuring target system bootloader
  installing grub to target devices
final system configuration
  configuring cloud-init
  calculating extra packages to install
  installing openssh-server
  retrieving openssh-server
  curtin command system-install
  unpacking openssh-server
  curtin command system-install
  downloading and installing security updates
  curtin command in-target
  restoring apt configuration
  curtin command in-target -

[ View full log ]
[ Reiniciar ahora ]

```



**Esperamos a que se instale todo, haga las particiones y las actualizaciones.**

- \* Antes de reboot retirar la imagen del disco:
- \* Dispositivos => Unidades Ópticas => Remove from virtual drive.


**Dependiendo de las características del ordenador, conexión y paquetes elegidos, la instalación durará en torno a los 45 minutos.**

33.

```

Ubuntu 22.04.3 LTS bioaire tty1
bioaire login: [ 166.930069] cloud-init[812]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 running 'modules:config' at Tue, 22 Oct 2024 05:59:15 +0000. Up 166.12 seconds.
[ 175.680092] cloud-init[823]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 running 'modules:final' at Tue, 22 Oct 2024 05:59:24 +0000. Up 175.50 seconds.
[ 175.914877] cloud-init[823]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 finished at Tue, 22 Oct 2024 05:59:24 +0000. Datasource DataSourceNone. Up 175.88 seconds
[ 175.917329] cloud-init[823]: 2024-10-22 05:59:24.901 - cc_final_message.py[WARNING]: Used fallback datasource
Hint: Num Lock on
bioaire login: davgalle
Password: _

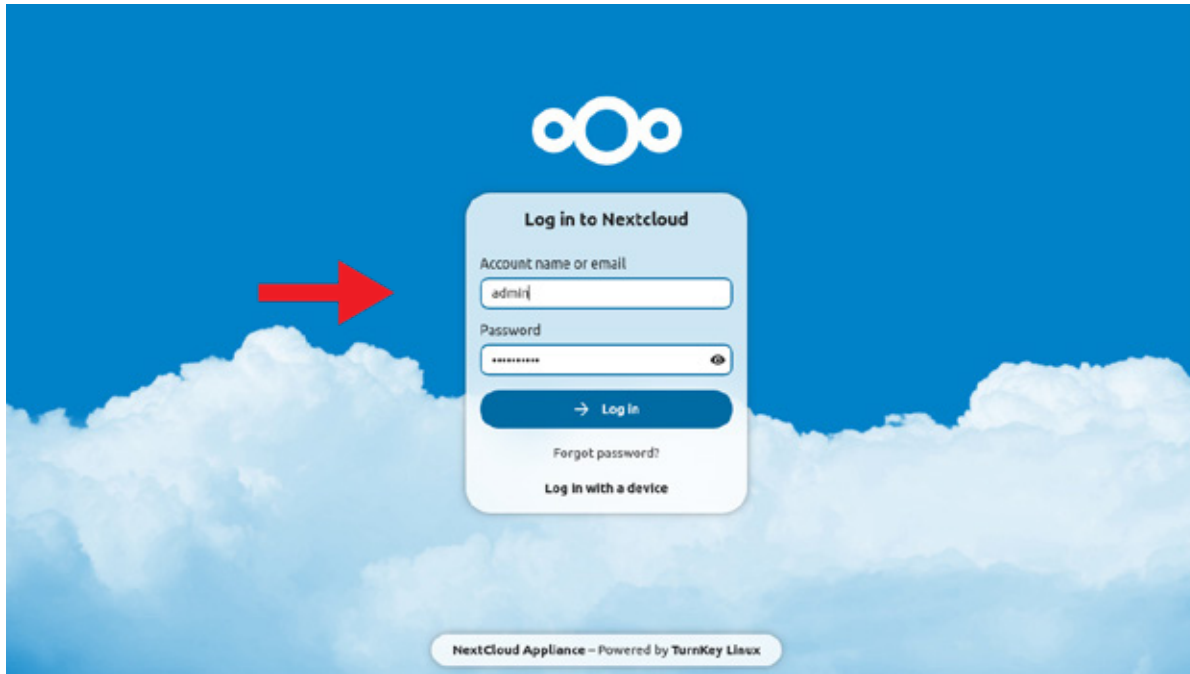
```



**El reboot tardará porque tiene que descargar unos paquetes.**

**Una vez reiniciado nos logueamos.**

34.



**Abrimos un navegador y si todo ha ido como debe, nos tiene que aparecer nuestra nube desplegada.**

**Nos logueamos y se creara la base de datos de nuestro perfil Admin.**

# 11

## Conf. ssh y ufw

### ¿Qué es SSH y UFW?

**SSH** permite acceder al servidor de forma **remota** y **segura** para administrar y configurar **Nextcloud** desde cualquier lugar.

**UFW** (Uncomplicated **Firewall**) se utiliza para proteger el servidor, **permitiendo** o **bloqueando** conexiones a puertos específicos.

Como el puerto **22** (SSH) y los puertos **80/443** (HTTP/HTTPS) para el acceso web de nuestro **Nextcloud**.



```
davgalle@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:06:ce:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.42/26 brd 192.168.1.63 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe06:ce54/64 scope link
        valid_lft forever preferred_lft forever
davgalle@ubuntu:~$
```

The image displays two terminal windows from a Ubuntu server, both titled 'dougall@ubuntuserver: ~'. The top window shows a successful ping to 'google.es' (142.250.280.67) with 56(84) bytes of data. It lists 10 successful ping attempts with varying times (e.g., 4.80 ms, 4.72 ms, 4.29 ms, 4.78 ms, 4.50 ms, 4.49 ms, 4.55 ms, 4.67 ms, 4.35 ms, 4.05 ms) and a final summary: '0/0 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp\_seq=11 ttl=116 time=5.12 ms'. Below this, it shows '64 bytes from mad7a25-in-f3.1e100.net' for the next 10 attempts, followed by '--- google.es ping statistics ---' and '35 packets transmitted, 35 received, 0% packet loss, time 4007ms', and 'rtt min/avg/max/ndev = 3.683/4.586/5.120 ms'. The bottom window shows a failed ping to 'google.es' (142.250.280.67) with 56(84) bytes of data. It lists 10 successful ping attempts with times (e.g., 5.59 ms, 4.47 ms, 4.15 ms, 4.15 ms, 4.27 ms) and a final summary: '5 packets transmitted, 5 received, 8% packet loss, time 4007ms', and 'rtt min/avg/max/ndev = 4.130/4.724/5.566/0.393 ms'. The terminal text is as follows:

```
dougall@ubuntuserver:~$ ping google.es
PING google.es (142.250.280.67) 56(84) bytes of data:
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=1 ttl=116 time=4.80 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=2 ttl=116 time=4.72 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=3 ttl=116 time=4.29 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=4 ttl=116 time=4.78 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=5 ttl=116 time=4.50 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=6 ttl=116 time=4.49 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=7 ttl=116 time=4.55 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=8 ttl=116 time=4.67 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=9 ttl=116 time=4.35 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=10 ttl=116 time=4.05 ms
0/0 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=11 ttl=116 time=5.12 ms
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
--- google.es ping statistics ---
35 packets transmitted, 35 received, 0% packet loss, time 4007ms
rtt min/avg/max/ndev = 3.683/4.586/5.120 ms
dougall@ubuntuserver:~$
```

```
dougall@ubuntuserver:~$ ping -4 google.es
PING google.es (142.250.280.67) 56(84) bytes of data:
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=1 ttl=116 time=5.59 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=2 ttl=116 time=4.47 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=3 ttl=116 time=4.15 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=4 ttl=116 time=4.15 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=5 ttl=116 time=4.27 ms
64 bytes from mad7a25-in-f3.1e100.net (142.250.280.67): icmp_seq=6 ttl=116 time=5.27 ms
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
64 bytes from mad7a25-in-f3.1e100.net:
--- google.es ping statistics ---
5 packets transmitted, 5 received, 8% packet loss, time 4007ms
rtt min/avg/max/ndev = 4.130/4.724/5.566/0.393 ms
dougall@ubuntuserver:~$
```

```
davgalle@ubuntuserver: ~$ ping -4 google.es
```

```
davgalle@ubuntuserver: ~$ ip a
```

37.

```

davgalle@ubuntu:~$ sudo -i
[sudo] password for davgalle:
root@ubuntu:~# service ssh status
ssh.service - OpenSSH Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
Active: active (running) since Tue 2024-10-22 09:26:57 UTC; 3 days ago
Docs: man:ssh(8)
      man:ssh_config(5)
Main PID: 878 (sshd)
Tasks: 1 (limit: 5636)
Memory: 7.0M
CPU: 2.826s
CGroup: /system.slice/ssh.service
        └─878 "sshd" /usr/sbin/sshd -D [listener] 0 of 10-100 startups

oct 24 17:55:21 ubuntu: sshd[358863]: Accepted password for davgalle from 192.168.1.1 port 58806 ssh2
oct 24 17:55:21 ubuntu: sshd[358863]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
oct 24 20:35:45 ubuntu: sshd[374849]: error: key_exchange_identification: banner line contains invalid characters
oct 24 20:35:45 ubuntu: sshd[374849]: banner exchange: Connection from 109.94.145.189 port 55520: invalid format
oct 24 20:36:10 ubuntu: sshd[374863]: Connection closed by 167.91.145.189 port 81395 [preauth]
oct 25 10:32:38 ubuntu: sshd[460552]: Accepted password for davgalle from 195.55.210.130 port 42303 ssh2
oct 25 10:32:38 ubuntu: sshd[460552]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
oct 25 12:38:18 ubuntu: sshd[474180]: Connection closed by 45.84.89.3 port 30316 [preauth]
oct 25 15:48:40 ubuntu: sshd[492966]: Accepted password for davgalle from 192.168.1.1 port 49005 ssh2
oct 25 15:48:40 ubuntu: sshd[492966]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
root@ubuntu:~# grep "Port" /etc/ssh/sshd_config
Port 4242
root@ubuntu:~#

```

38.

```

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/bin:/bin:/sbin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

# Include /etc/ssh/sshd_config.d/*.conf

#Port 22
Port 4242
#AddressFamily
#ListenAddress
#ListenAddress

#Identity
#IdentityAgent
#IdentityFile
#KexAlgorithms
#MACs
#PubkeyAcceptedKeyTypes
#TrustedUserCAKeys
#UsePrivilegeSeparation
#X11
#X11Forwarding yes
#XAuthLocation /usr/libexec/openssh/xauth

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 3m
#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
#AuthorizedKeysFile none

```

Deberíamos de tener instalado sudo.  
Accedemos como usuario root.

root:

```
davgalle@ubuntu:~$ sudo -i
```

Mostramos el estado de ssh:

```
davgalle@ubuntu:~$ service ssh status
```

Mostramos el puerto sshd:

```
davgalle@ubuntu:~$ grep "^Port" /etc/ssh/sshd_config
```

Cambiamos el puerto 22 del SSHD por el que queramos (4242)

Modificamos el archivo sshd\_config:

```
davgalle@ubuntu:~$ vim /etc/ssh/sshd_config
```

```
#Port 22 -> Port 4242
```

```
#PermitRootLogin prohibit-password -> PermitRootLogin no
```



**39.**

```

root@ubuntu-server: ~
# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# KexMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519

Port 4242
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-sha1,hmac-sha2,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalForward no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1h
# UserKnownHostsFile ~/.ssh/known_hosts.d/%k
# SendEnv LANG LC_*
# HostKnownHosts yes
# GSSAPIAuthentication yes

"/etc/ssh/ssh_config" 518, 16488

```

## Cambiamos el puerto 22 del SSH por el que queramos (4242)

## Modificamos ssh\_config y reiniciamos ssh:

```
davgalle@ubuntuserver: ~$ vim /etc/ssh/ssh_config
```

```
#Port 22 -> Port 4242
```

```
davgalle@ubuntuuser:~$ sudo service ssh restart
```

**40.**

```

root@ubuntuuser:~# sftp status
Status: active

To          Action     From
--          -
root@ubuntuuser:~#

```

## Verificamos si UFW (firewall) está instalado y si no lo está los instalamos:

## Verificamos, instalamos y habilitamos:

```
davgalle@ubuntuuser: ~$ sudo ufw status
```

```
davgalle@ubuntuserver: ~$ sudo install ufw
```

```
davqalle@ubuntuserver: ~$ sudo ufw enable
```

41.


```

root@ubuntu:~# ufw status
Status: active

To Action From
--
4242 ALLOW Anywhere
443/tcp ALLOW Anywhere
80/tcp ALLOW Anywhere
4242 (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)

root@ubuntu:~#

```



**Tenemos que habilitar los puertos en el firewall.**

**Habilitamos el puerto ssh:**

```
davgalle@ubuntu:~$ sudo ufw allow 4242
```

**Habilitamos 80 y 443 de http y https:**

```
davgalle@ubuntu:~$ sudo ufw allow 80, 443/tcp
```

42.

```

davgalle@ubuntu:~$ ssh -p 4242
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-124-generic x86_64)

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

System information as of 25 oct 2024 16:46:29 UTC

System load: 0.87      Processes:    141
Usage of /:  2.0% of 1.8TB    Users logged in: 1
Memory usage: 15%      IPv4 address for enp2s3: 192.168.1.42
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

El mantenimiento de seguridad expandido para aplicaciones está desactivado.
Se pueden aplicar 20 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESX Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esx o ejecute «sudo pro status»

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Oct 25 15:46:41 2024 from 192.168.1.1
davgalle@ubuntu:~$

```

**Abrimos Terminal de nuestro pc y accedemos al servidor por el puerto ssh**

```
ssh nombreusuario@192.168.1.42 -p 4242
```

**Se nos mostrará información de interés cada vez que entremos.**

# 12

## Configurar la Red

Necesitamos poder acceder a nuestra nube a través de internet.



NetPractice 42

```
devgalle@ubuntu:server: ~  
,  
'supportedDatabases' =>  
array (  
  0 => 'mysql',  
,  
'memcache.locking' => '\\OC\\Memcache\\Redis',  
'memcache.local' => '\\OC\\Memcache\\Redis',  
'redis' =>  
array (  
  'host' => '/tmp/sockets/redis.sock',  
  'port' => 0,  
,  
'log_type' => 'file',  
'logfile' => '/var/snap/nextcloud/current/logs/nextcloud.log',  
'logfilemode' => 416,  
'maintenance_window_start' => 1,  
'instanceid' => 'oc1ptz4csnq',  
'passwordsalt' => [REDACTED],  
'secret' => [REDACTED],  
'trusted_domains' =>  
array (  
  0 => [REDACTED],  
  1 => [REDACTED],  
  2 => [REDACTED],  
,  
'datadirectory' => '/var/snap/nextcloud/common/nextcloud/data',  
'dbtype' => 'mysql',  
'version' => '29.0.8.1',  
'overwrite.cli.url' => 'http://192.168.1.42',  
'dbname' => 'nextcloud',  
'dbhost' => 'localhost:/tmp/sockets/mysql.sock',  
'dbport' => '',  
'databseprefix' => 'oc_',  
'mysql.utf8mb4' => true,  
'dbuser' => 'nextcloud',  
'dbpassword' => [REDACTED],  
'installed' => true,  
'maintenance' => false,  
'loglevel' => 2,  
,  
'trusted_domains' =>  
array (  
  0 => '192.168.1.43',  
  1 => 'xx.xx.xx.xx',  
  2 => 'cloud.mydominio.-com',  
,  
'trusted_domains' =>  
array (  
  0 => IP servidor,  
  1 => Ip Router ,  
  2 => Dominio.com`,  
)
```

## Hay que configurar las excepciones de nuestra nube en 'config.php'.

Accedemos al archivo siguiendo este ruta:

```
vim /var/snap/nextcloud/current/nextcloud/config/config.php
```

Comando para buscar directorios, etc:

```
sudo find / -type d -name (nombre a buscar)
```

```
`trusted_domains` =>  
array (  
  0 => `192.168.1.43`,  
  1 => `xx.xx.xx.xx`,  
  2 => `cloud.mydominio.-com`,  
,
```

```
`trusted_domains` =>  
array (  
  0 => IP servidor,  
  1 => Ip Router ,  
  2 => Dominio.com`,  
)
```

## 44.

**Accedemos a nuestro Router desde el navegador web.**

Tabla actual de mapeo de puertos						
	Nombre	Protocolo	Puerto/Rango Externo	Puerto/Rango Interno	Dirección IP	Activar
	cloud	TCP	80	80	192.168.1.42	
	https	TCP	443	443	192.168.1.42	
	Exter	TCP	4242	4242	192.168.1.42	
	NUEVA	TCP	90	90	192.168.1.13	

**Mapeamos los puertos para http, https y ssh**

<b>Nombre:</b>	<b>http</b>	<b>https</b>
<b>Dirección IP:</b>	<b>192.168.1.43</b>	<b>192.168.1.43</b>
<b>Protocolo:</b>	<b>TCP</b>	<b>TCP</b>
<b>WLAN:</b>	<b>80</b>	<b>443</b>



# 13

## https y dominio

### *¿Qué es un certificado https?*

Un **certificado HTTPS** es un archivo digital que asegura la **conexión** entre un **navegador** web y un **servidor**, garantizando que los datos transmitidos estén **cifrados** y **protegidos**.

Esto evita que terceros puedan interceptar o modificar la información.

Además, **autentica** la identidad del sitio web, demostrando a los usuarios que están en el sitio **correcto** y que es **seguro**.

### *¿Cómo tener un certificado válido?*

Descargando **Let's Encrypt**, que es una autoridad certificadora que nos permitirá **generar** certificados **HTTPS válidos** y **gratuitos**.

Estos certificados son **reconocidos** por la mayoría de navegadores, asegurando **conexiones seguras** y **encriptadas** en nuestro **servidor**.





**45.**

**Los requisitos para poder generar certificados válidos son:**







<b>Firewall:</b>	✓
<b>ssh:</b>	✓
<b>80,443/tcp:</b>	✓
<b>Dominio:</b>	✗

45.











Los requisitos para poder generar certificados válidos son:

Firewall:   
ssh:   
80,443/tcp:   
Dominio: 

En el Hosting donde tengamos nuestro dominio, crearemos un subdominio y configuraremos las DNS para que apunten a la dirección IP de nuestro Router.

Añadir subdominio							
Subdominio		Carpeta de destino				Acciones	
cloud		/web/cloud					
placas-solares		/web/placas_solares					
server		/web/lbserver					

<input type="checkbox"/>	A	letter	Apunta a:		900	<a href="#">Editar registro</a>	
<input type="checkbox"/>	A	smtp	Apunta a:		900	<a href="#">Editar registro</a>	
<input type="checkbox"/>	A	cloud	Apunta a:		900	<a href="#">Editar registro</a>	
<input type="checkbox"/>	A	imap	Apunta a:		900	<a href="#">Editar registro</a>	
<input type="checkbox"/>	A	gestionaria	Apunta a:		900	<a href="#">Editar registro</a>	



46.

## Generar certificados

### Accedemos al Terminal desde ssh

```
ssh username@192.168.1.42 -p 4242
```

```
cd /usr/local
```

### Descargamos el repositorio de github

```
sudo git clone https://github.com/letsencrypt/letsencrypt
```

### Si hacemos 'ls' debemos de ver el directorio

```
bin etc games include letsencrypt lib man sbin share ...
```

```
cd letsencrypt
```

### Dentro de letsencrypt generamos el certificado

```
sudo ./letsencrypt-auto --apache -d cloud.4trastos.com
```

47.

## Nos pedirán una cuenta de correo a cambio del certificado.

### Y no aceptaremos nada más.

```
Enter email address: xxxx@xxx.com
```

```
(A)gree / (C)ancel => A
```

```
(Y)es / (N)o => N
```

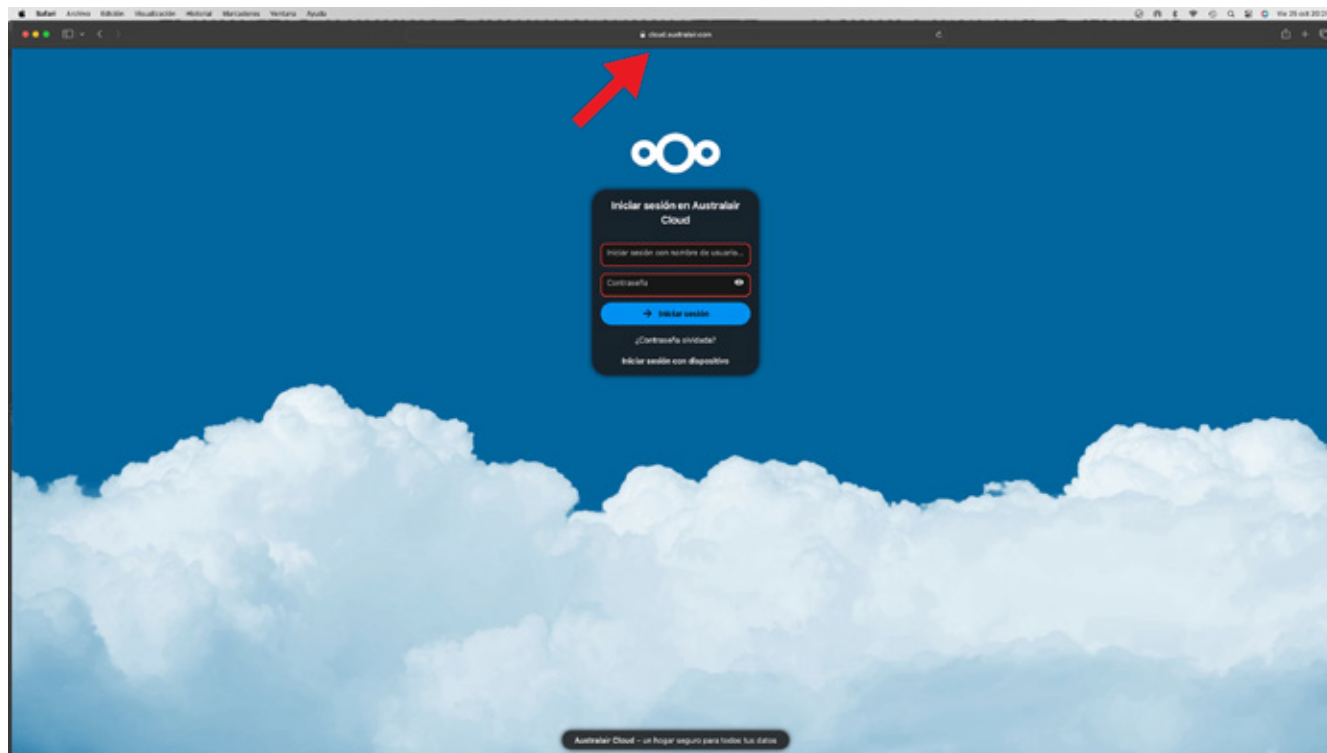
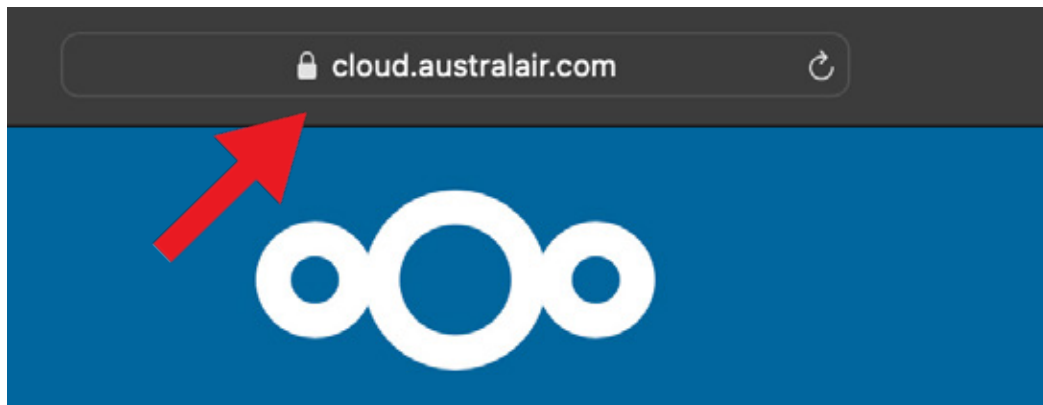
```
Select the appropriate number[ 1 - 2 ] => 2
```

### Reiniciaremos apache.

```
sudo systemctl restart apache2
```

### Volvemos a nuestro navegador y cambiamos la IP por el subdominio que habiamos creado.





# 14

## Conf. SSH Key

### *¿Qué es el acceso con SSH Key?*

El **acceso SSH** con **clave** pública se basa en **criptografía** asimétrica y reemplaza el uso de contraseñas.

- **Mayor seguridad**
- **Acceso automatizado y controlado**
- **Registro y auditoría individual**

Se evita completamente el uso de contraseña en el servidor.

Las claves privadas son **casi imposibles** de **romper** si están bien protegidas



## 47.

### Generar claves SSH en el cliente

En nuestro host local ejecutamos:

```
ssh-keygen -t ed25519 -C "clave acceso cloud" -f ~/.ssh/id_ed25519
```

Usa una passphrase para mayor seguridad.

Copiar clave pública al servidor:

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub -p 4242 usuario@IP-o-dominio-del-servidor
```

También se puede hacer manualmente:

```
# En tu máquina local  
cat ~/.ssh/id_ed25519.pub
```

Y luego en el servidor:

```
mkdir -p ~/.ssh  
nano ~/.ssh/authorized_keys  
# Pega la clave pública aquí  
chmod 600 ~/.ssh/authorized_keys  
chmod 700 ~/.ssh
```

## 48.

### Configurar el servidor SSH

Edita el archivo de configuración

```
sudo nano /etc/ssh/sshd_config
```

Asegúrate de tener lo siguiente:

```
Port 4242  
PermitRootLogin no  
PubkeyAuthentication yes  
PasswordAuthentication no  
KbdInteractiveAuthentication no
```

Reinicia el servicio SSH:

```
sudo systemctl restart ssh
```

Probar acceso desde el cliente:

```
ssh -i ~/.ssh/id_ed25519 -p 4242 usuario@cloud.dominio.com
```

Salir y volver a conectar:

```
ssh -p 4242 usuario@cloud.dominio.com
```

# Cloud Server Project

*Implementa un servicio cloud en opensource*

# ¡Muchas gracias!



David G. & Paolo S.

---