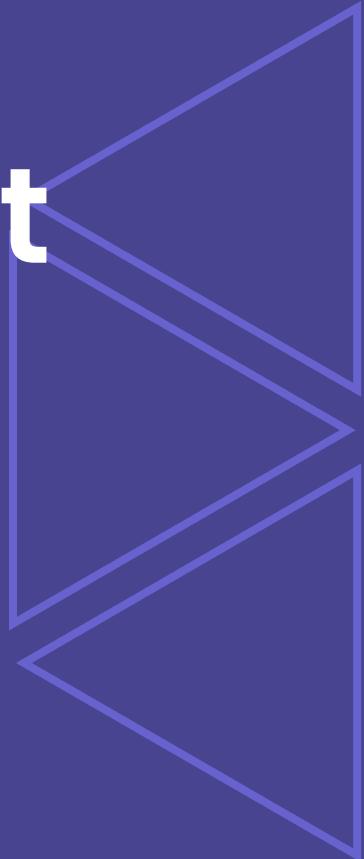


Cloud Server Project

Implementa un servicio cloud en opensource



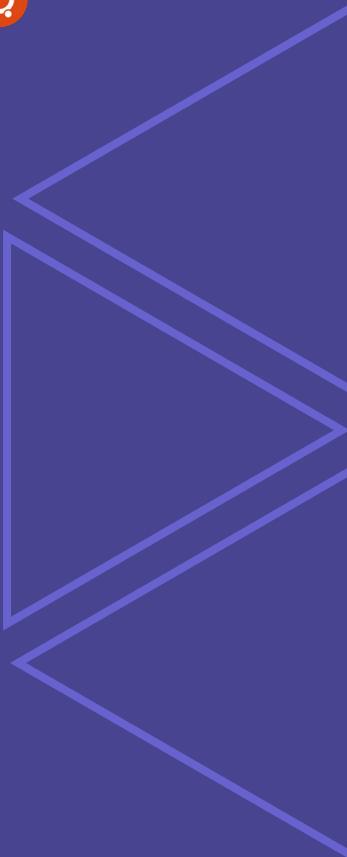
David G. & Paolo S.



01

Introducción

01



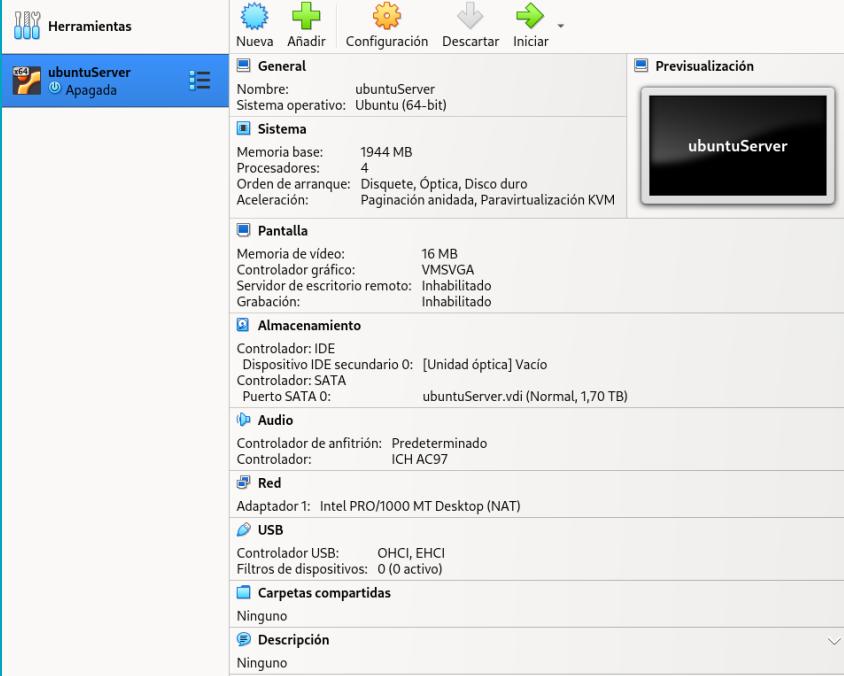
¿Qué vamos a ver?

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

- Pruebas de software:** para probar aplicaciones en diferentes sistemas operativos.
- Entornos de desarrollo.**
- Seguridad:** probar software sospechoso o navegar por internet sin poner en riesgo el equipo principal.
- 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.

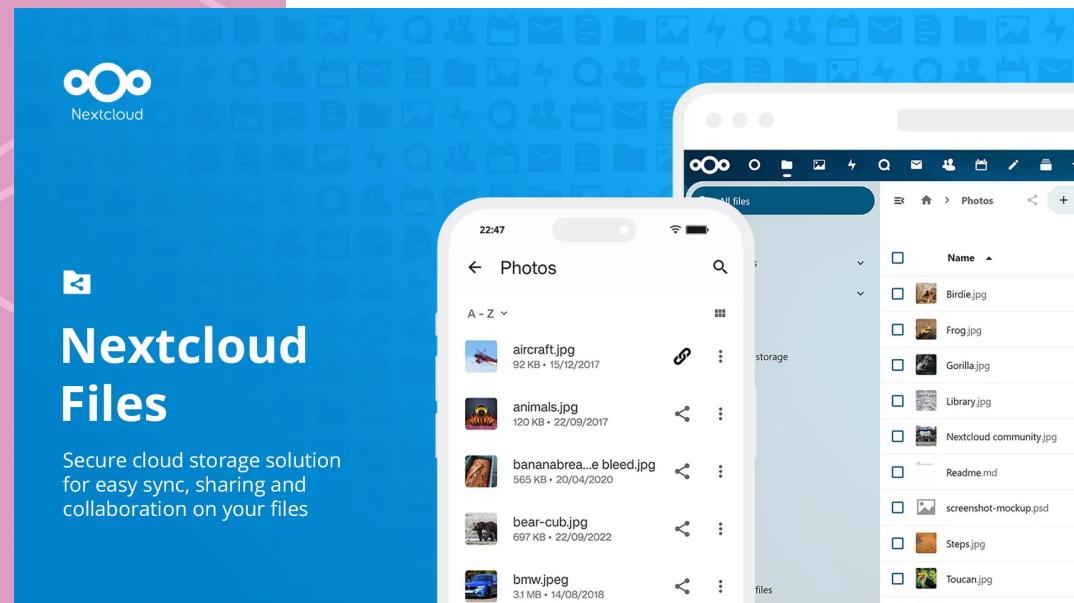
<https://nextcloud.com/es/>

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>



04

Privacidad

04



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

La importancia de la privacidad

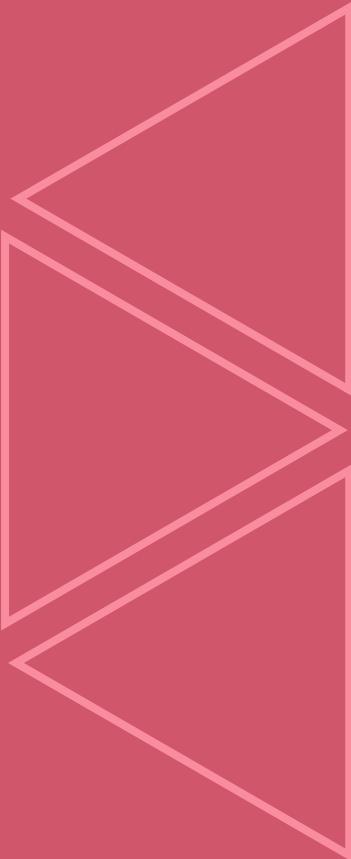
La **privacidad** en **Nextcloud** es fundamental porque te permite tener **control total** sobre tus datos.

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

06 Requisitos

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.



Ordenador

Para instalar Ubuntu Server corriendo en VirtualBox 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

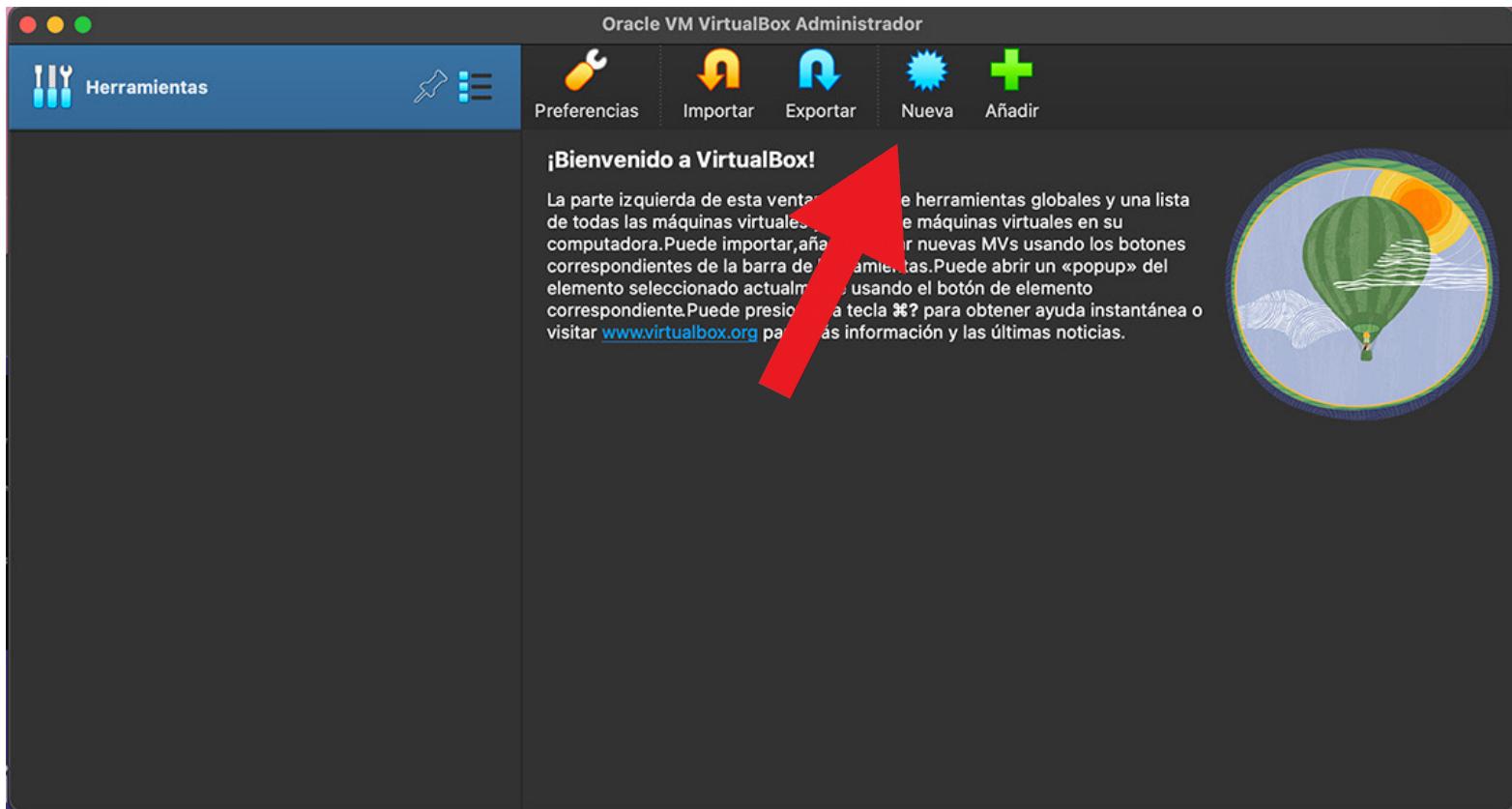
<https://www.virtualbox.org/>

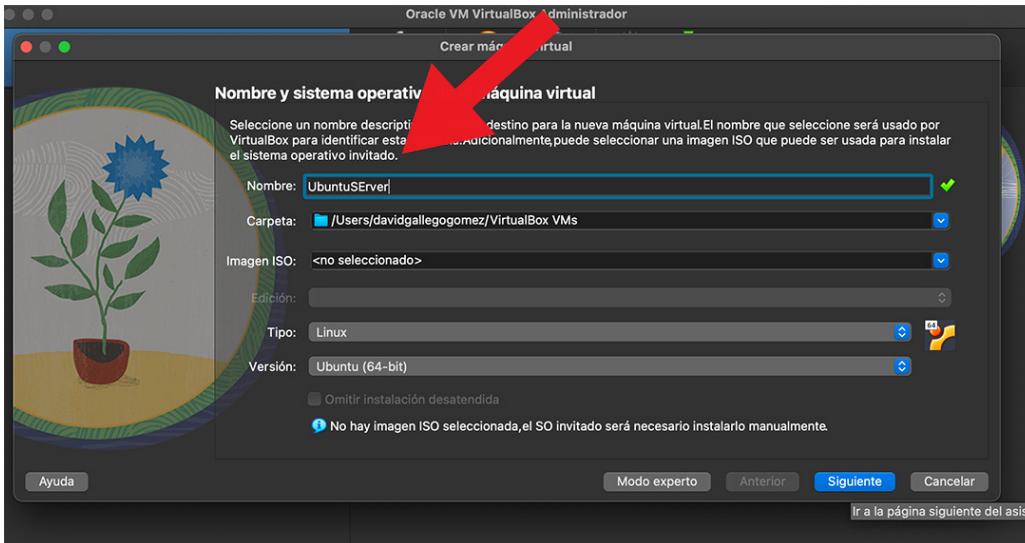
Descargar Nextcloud

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.

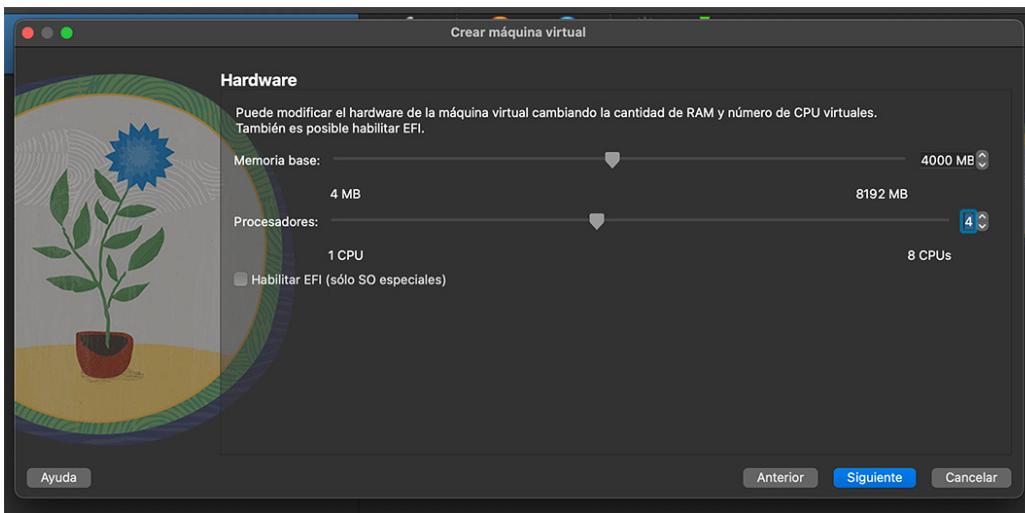
<https://nextcloud.com/es/>

1. Abrimos VirtualBox y hacemos clic en nueva



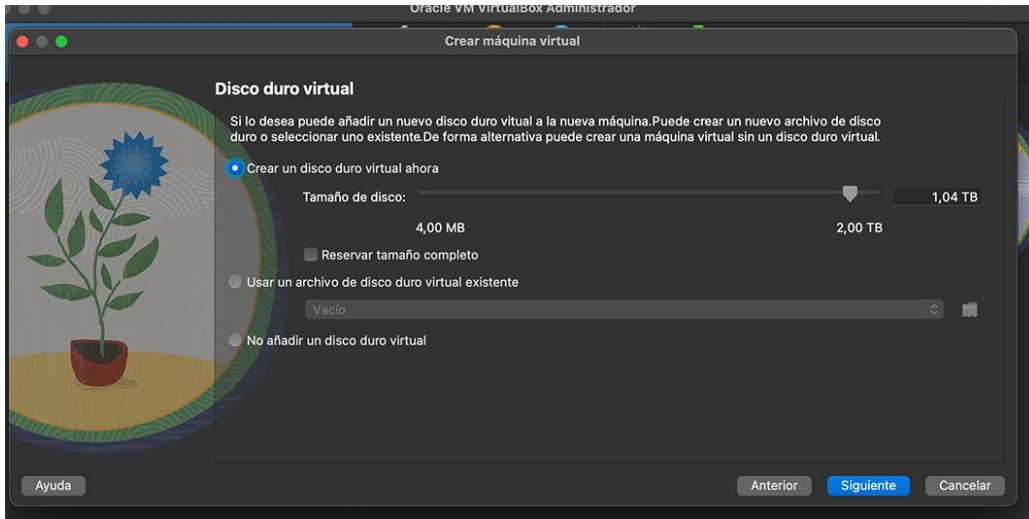


1. **Ponemos nombre a la nueva máquina virtual.**



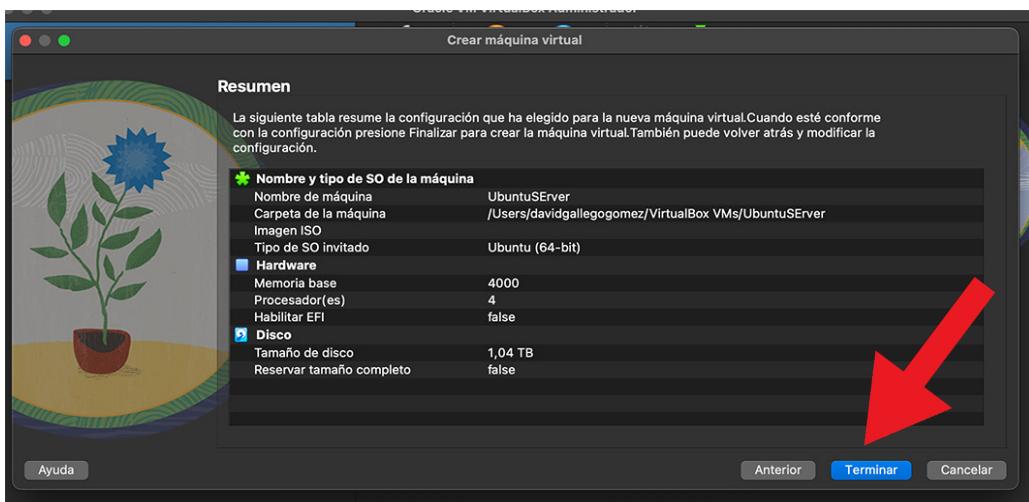
2. **Seleccionamos la cantidad de RAM el número de CPU.**

En este caso 4Gb de RAM y 4 CPU.

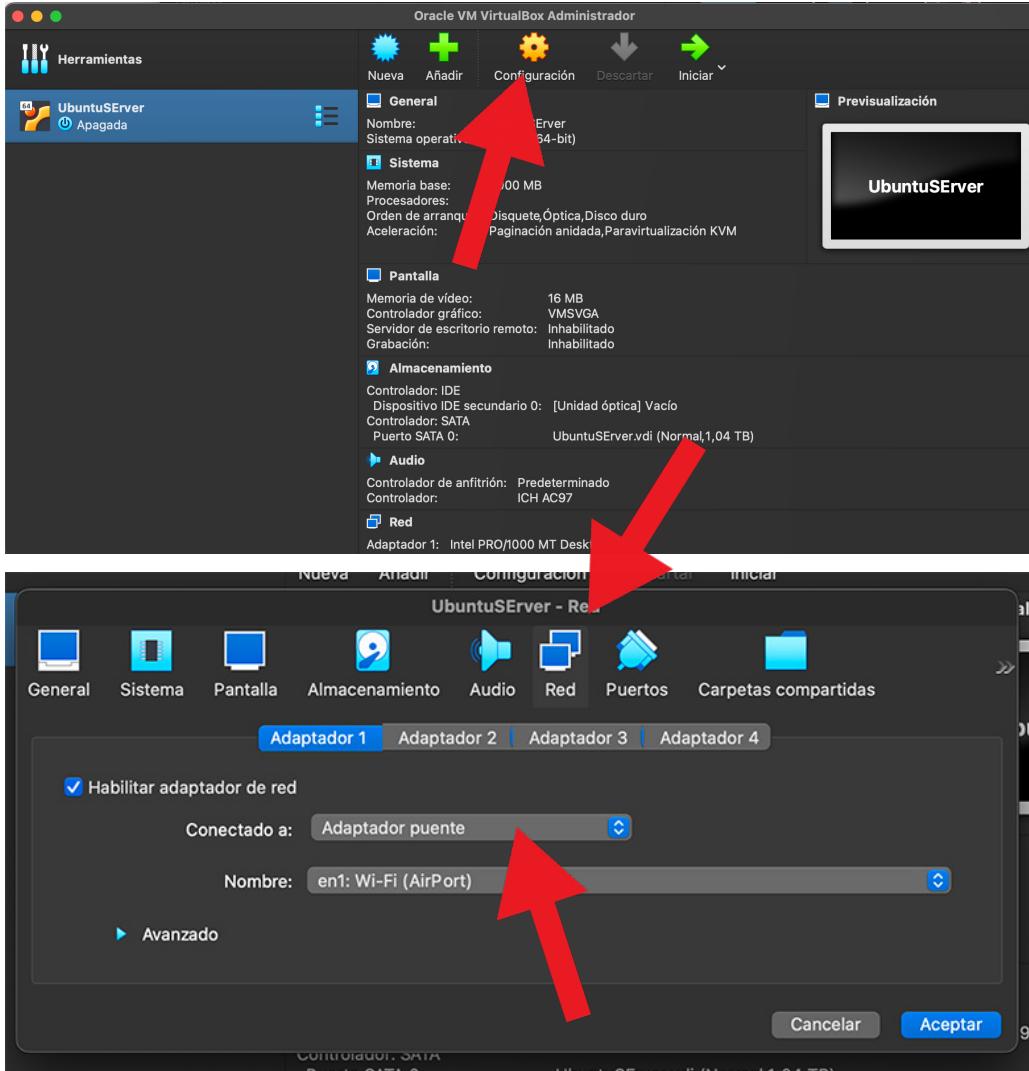


3. Creamos el disco duro y le damos la capacidad que creamos conveniente.

Para este caso 1Tb



4. Repasamos que está todo OK y aceptamos.

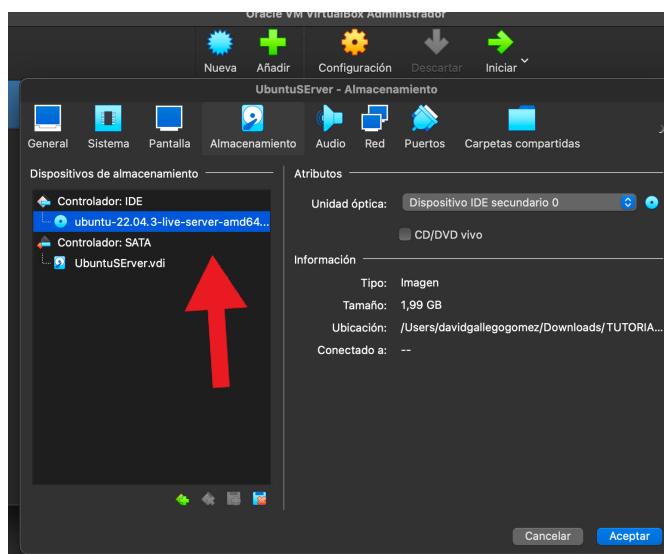
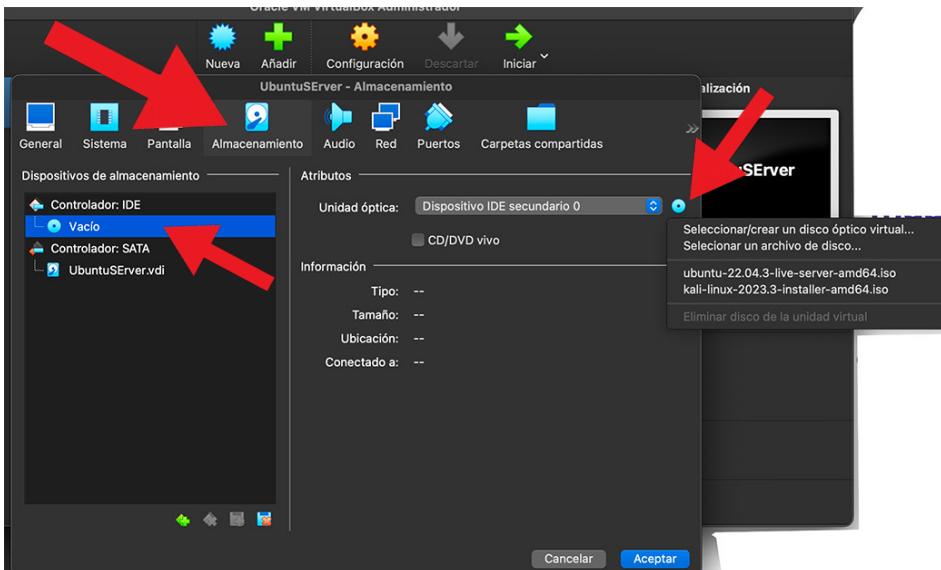


5. Ya tenemos máquina creada y ahora debemos de configurarla para instalar el Servidor.

clic en **Configuración**.

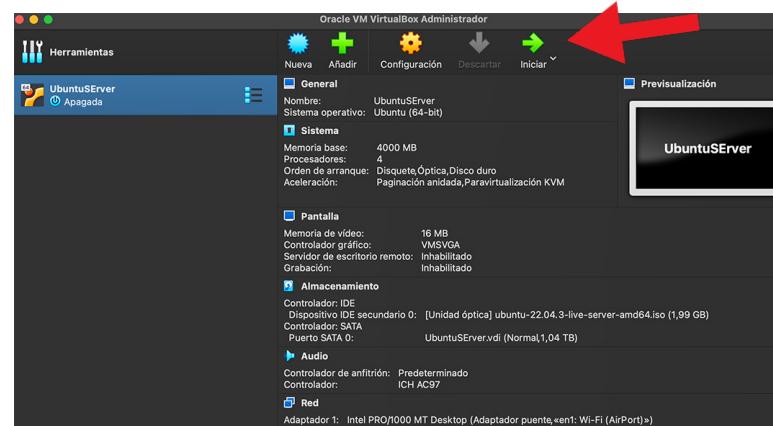
6. Para poder tener conexión a internet:

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.



Archivo Máquina Ver Entrada Dispositivos Ayuda

GNU GRUB version 2.06

*Try or Install Ubuntu Server
Ubuntu Server with the HWE kernel
Test memory



Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 22s.

Ctrl Derecho

Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom!

[Help]

Use UP, DOWN and ENTER keys to select your language.

- [Asturianu ►]
- [Bahasa Indonesia ►]
- [Català ►]
- [Deutsch ►]
- [English ►]
- [English (UK) ►]
- [**Español** ►]
- [Français ►]
- [Galego ►]
- [Hrvatski ►]
- [Latviski ►]
- [Lietuviškai ►]
- [Magyar ►]
- [Nederlands ►]

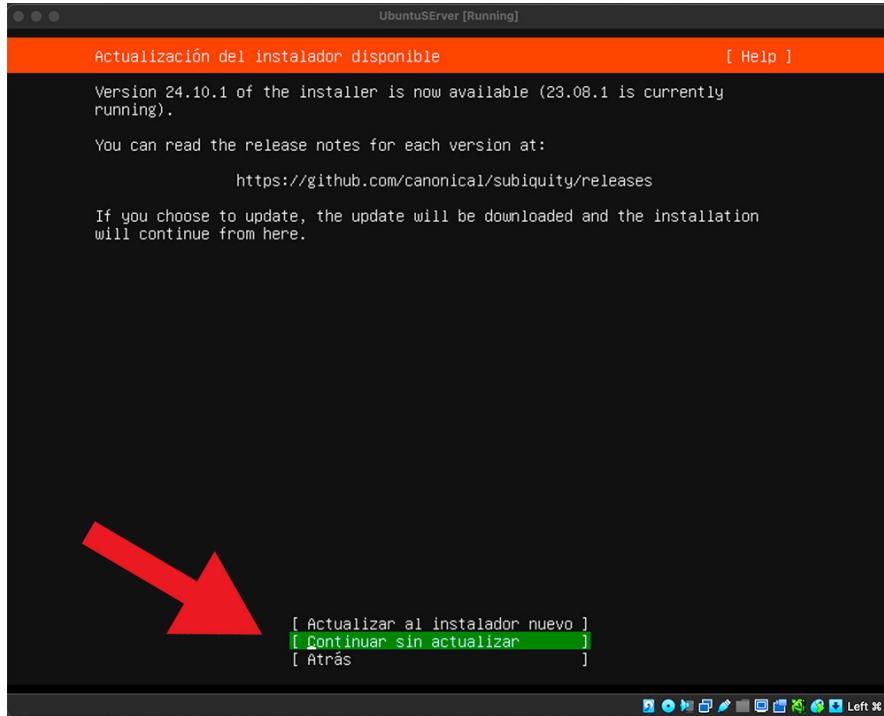
10. Install Ubuntu Server.

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

Esto rondará los 40 / 45 minutos.

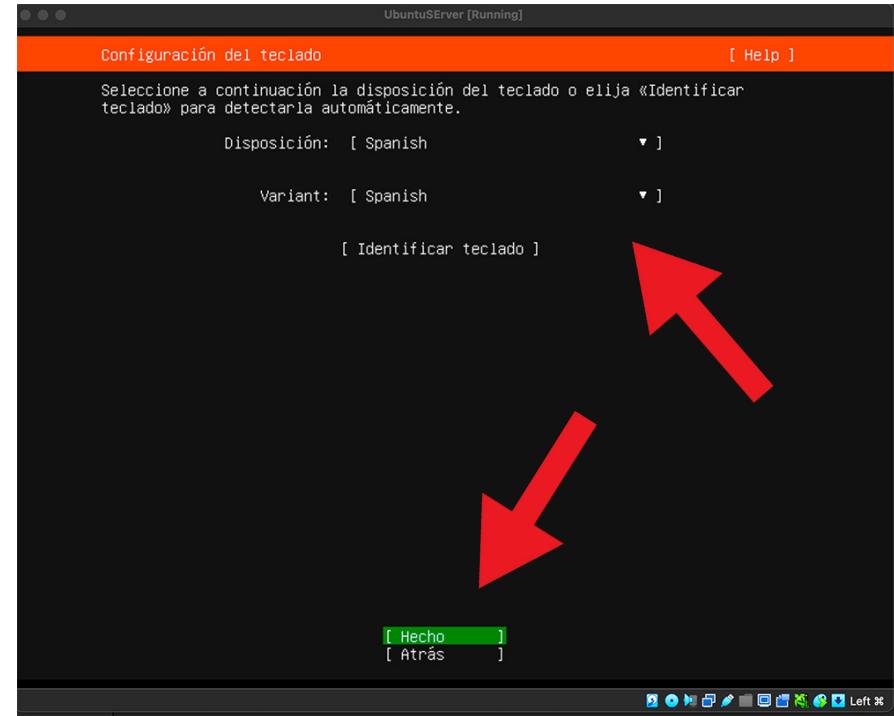
11. Seleccionamos el idioma.

12.



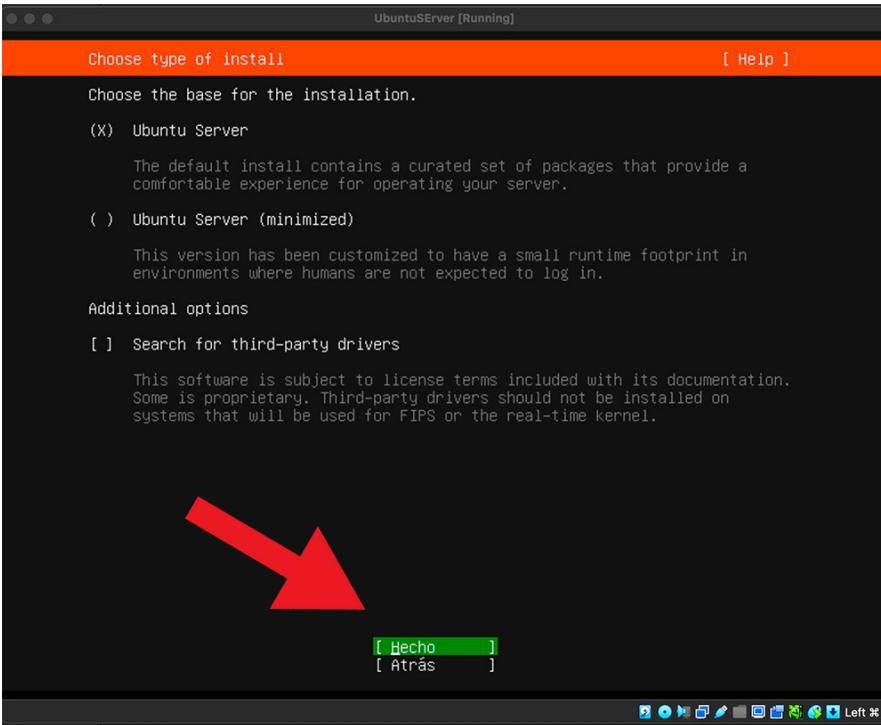
No actualizamos al nuevo.
Esta versión es muy estable

13.



Configuramos el idioma del teclado
aunque Ubuntu lo hace automaticamente.

14.



Choose type of install

[Help]

Choose the base for the installation.

(X) Ubuntu Server

The default install contains a curated set of packages that provide a comfortable experience for operating your server.

() Ubuntu Server (minimized)

This version has been customized to have a small runtime footprint in environments where humans are not expected to log in.

Additional options

[] Search for third-party drivers

This software is subject to license terms included with its documentation. Some is proprietary. Third-party drivers should not be installed on systems that will be used for FIPS or the real-time kernel.

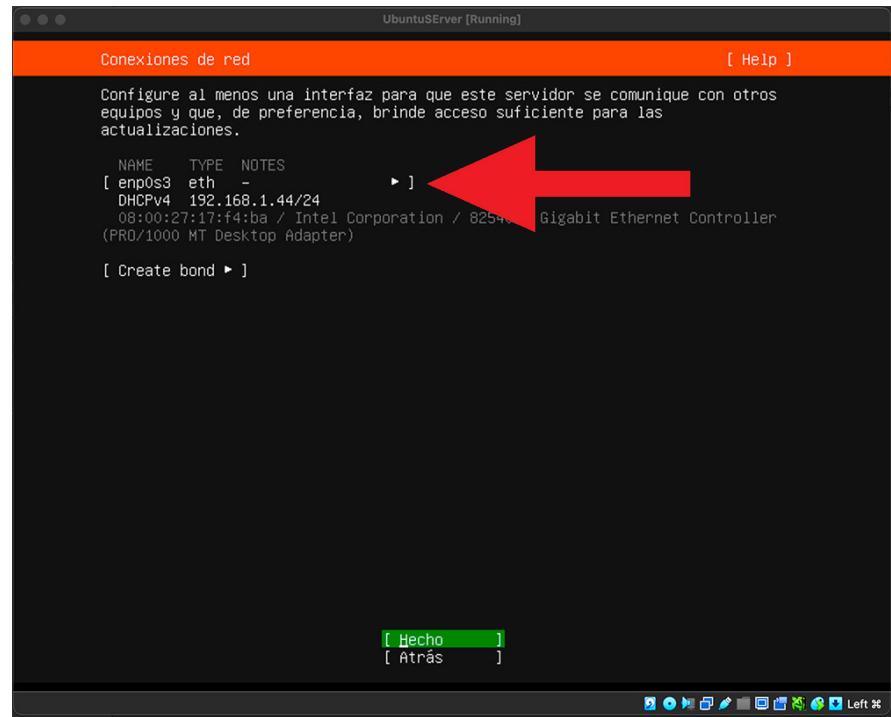
[Hecho]
[Atrás]

Left ☰

Dejamos marcado Ubuntu Server

14

15.



Conexiones de red

[Help]

Configure al menos una interfaz para que este servidor se comunique con otros equipos y que, de preferencia, brinde acceso suficiente para las actualizaciones.

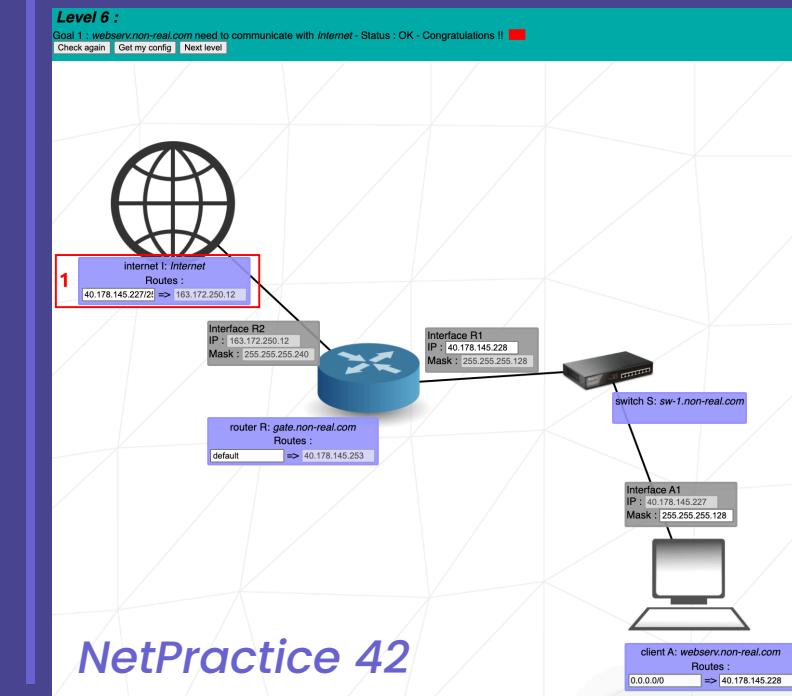
NAME	TYPE	NOTES
[enp0s3	eth	-
DHCPv4	192.168.1.44/24	
08:00:27:17:f4:ba	/ Intel Corporation / 8254	Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)

[Create bond ▶]

[Hecho]
[Atrás]

Left ☰

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



07

Conexiones de Red

¿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, ect.
- **Medios de transmisión:** Los canales a través de los cuales **vian 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**.

08 Protocolos TCP/IP

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**.

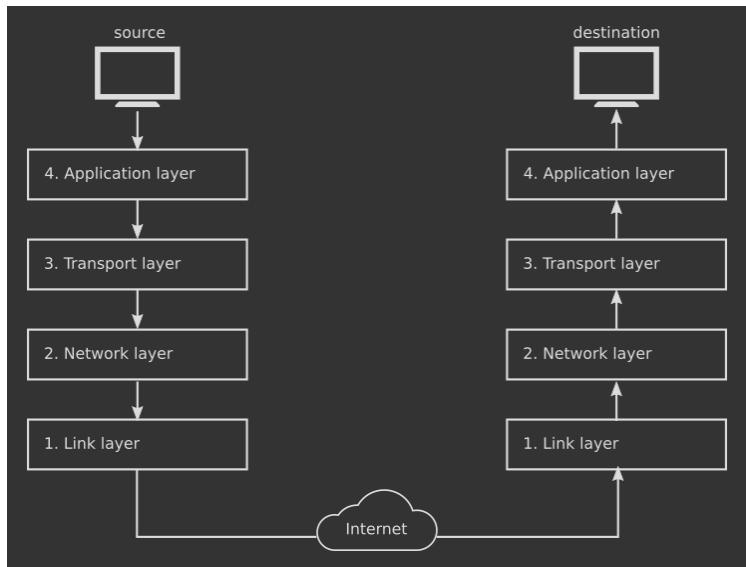
TCP **garantiza** la integridad de los datos. Antes de transmitir datos, TCP establece una **conexión** entre una **fuente** y su **destino**, y permanece activa hasta que comienza la comunicación.

¿Qué significa TCP?

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

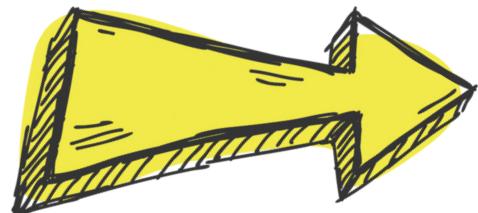
IP y máscara subred

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** (1111111.1111111.1111111.1) 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** (1111111.1111111.1111111.1) 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

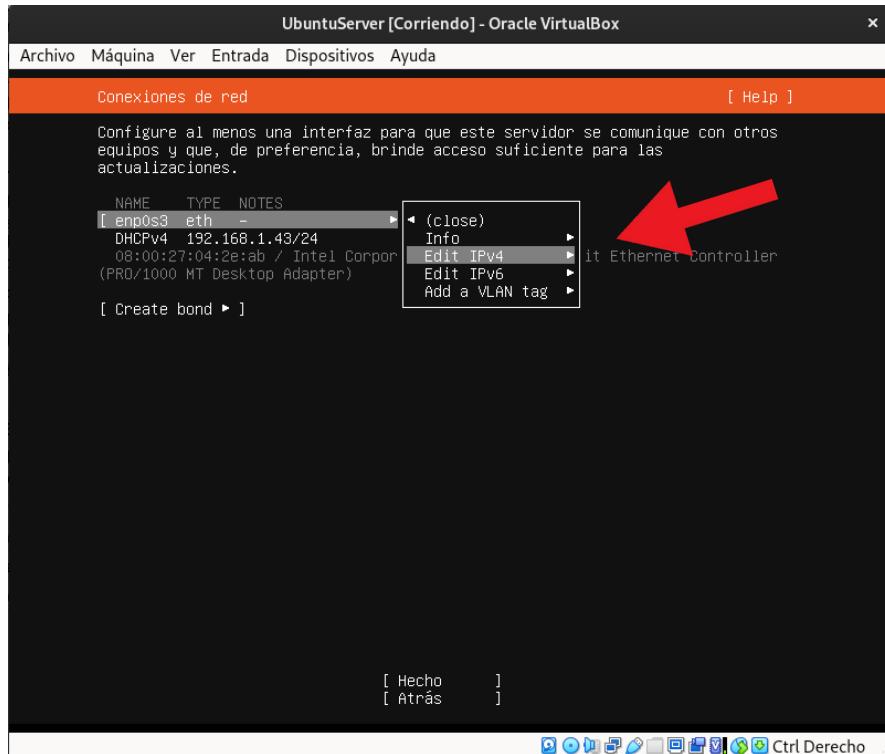
Último Host Disponible: 104.198.241.126

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).

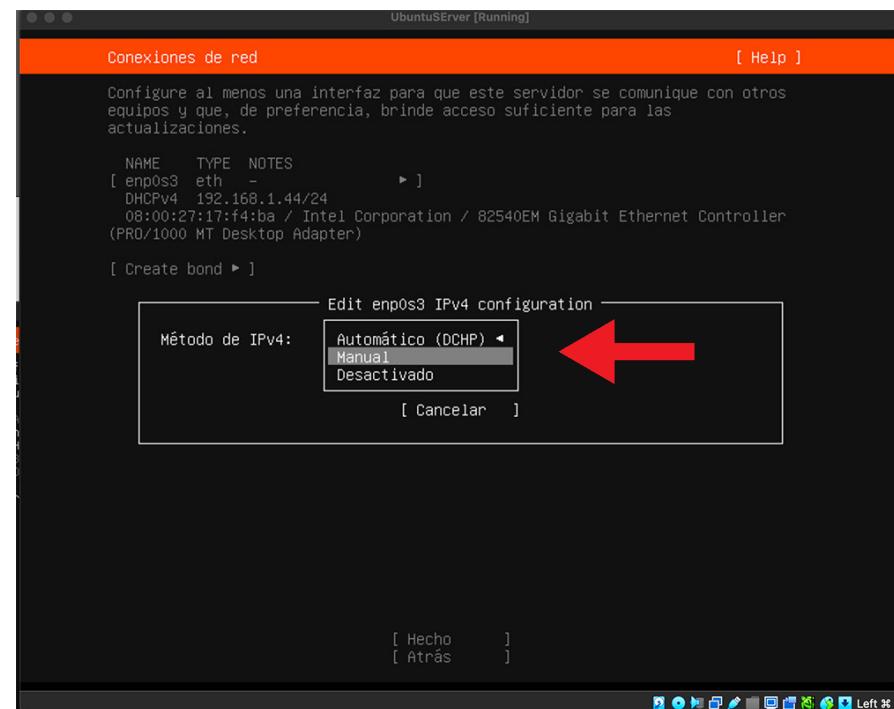
16.



Creamos una IP estática usando el protocolo de 32bits
IPv4

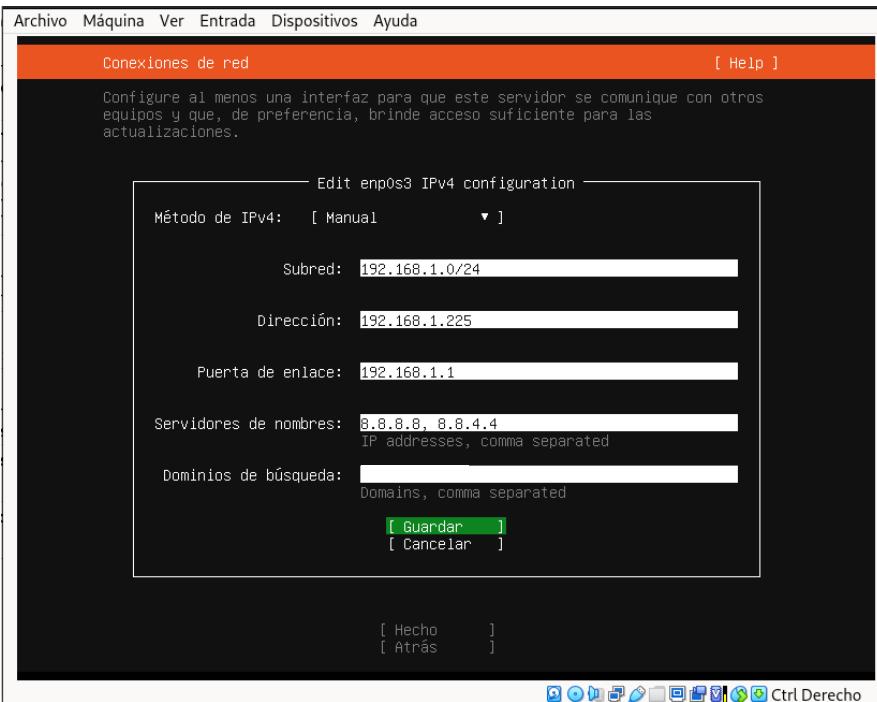
20

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*

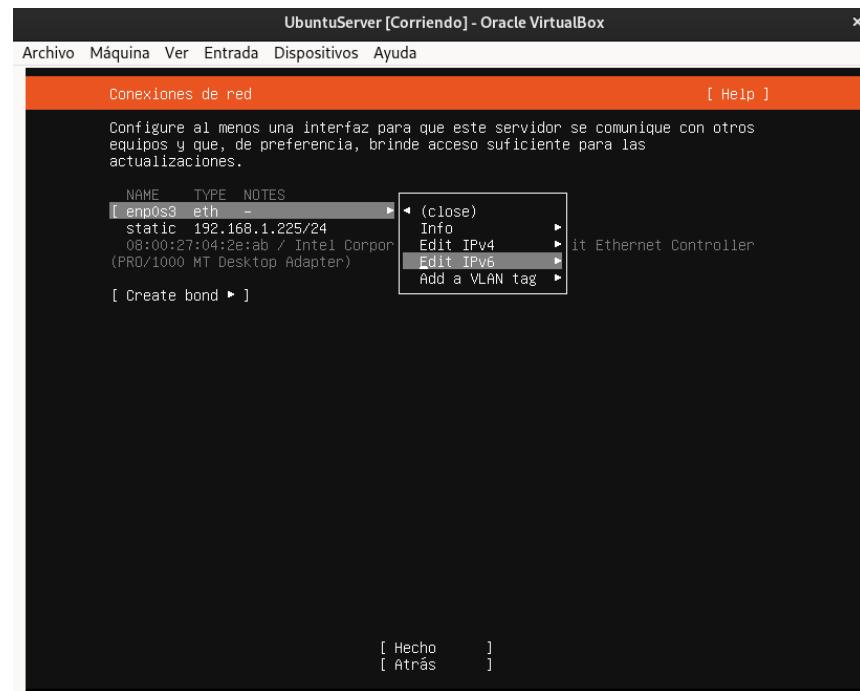
Puerta de enlace: 192.168.1.1 *La IP del Router*

Servidores: 8.8.8.8, 8.8.4.4 *Servidores de Google y Yahoo*

Dominios: *No es necesario.*

21

19.



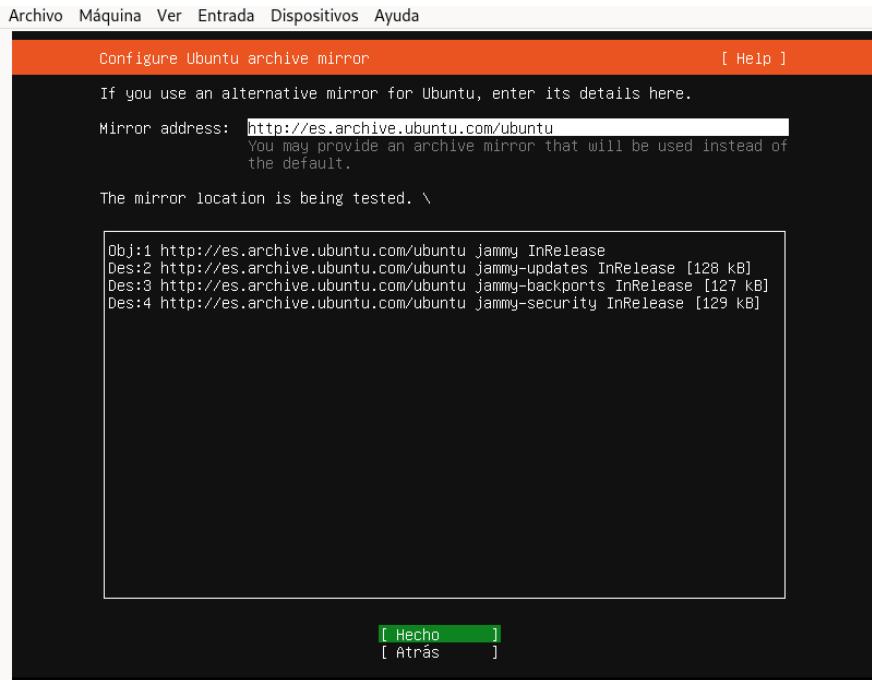
Subred: 192.168.1.0/24

IP: 192.168.1.42 *Rango del 1 al 254 usando prefijo 24*

**Protocolo IPv6 de 128bits
lo dejamos en automático.**

**Esperamos a que se apliquen los
cambios y continuamos**

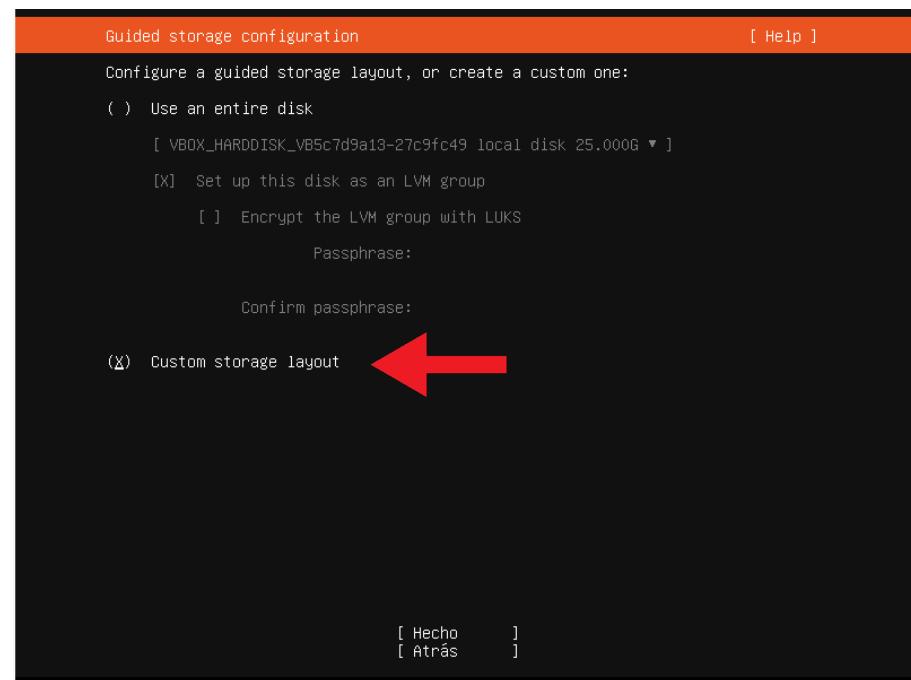
20.



Esperamos a que termine el test de localización de Ubuntu.

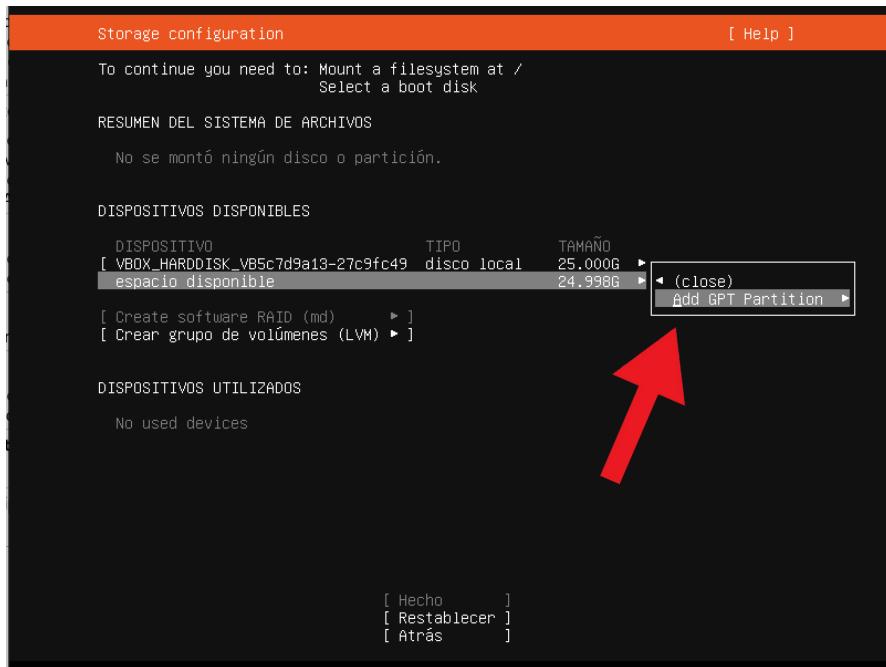
22

21.



Configuramos manualmente la partición del disco duro.

22.

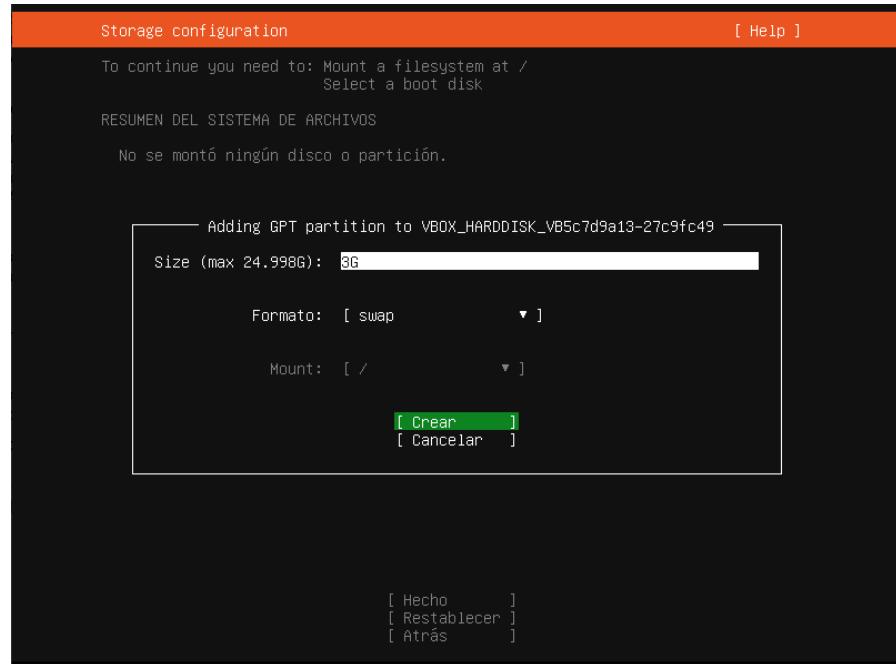


Elegimos la parte de espacio disponible que es la hemos configurado en nuestra MV.

espacio vacío => Add GPT Partition

23

23.

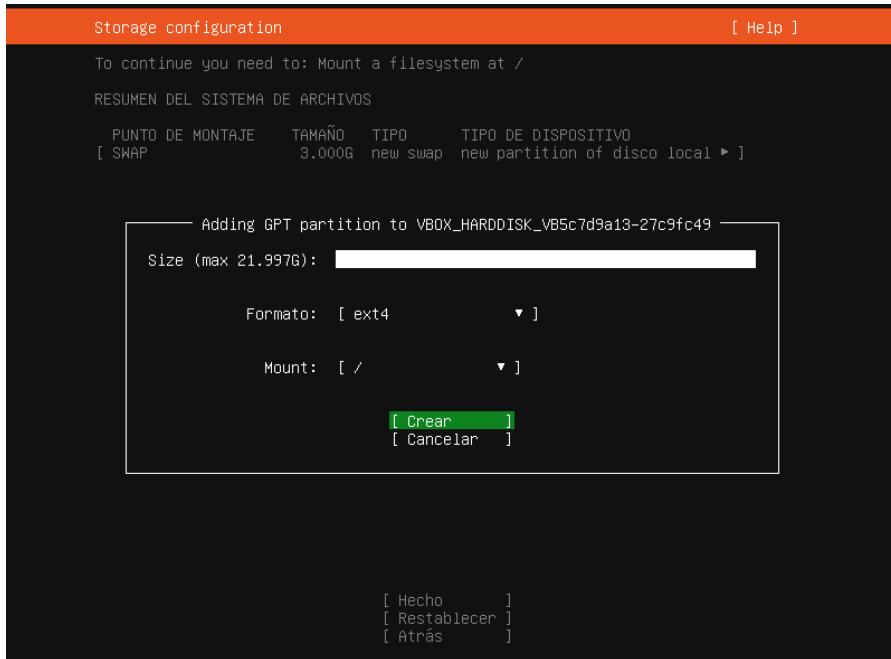


Creamos la partición SWAP con un tamaño algo menor o igual a la RAM que hemos configurado en la MV.

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.

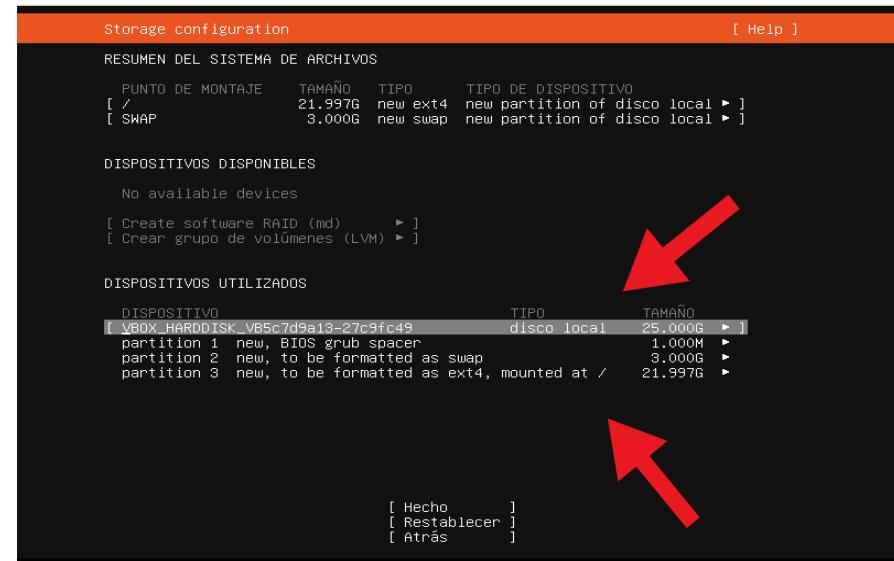


Creamos otra partición del resto del disco y se Mount de deja en '/' que es la raiz.

Size en blanco para max.

24

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: David Gallego

El nombre del servidor: [REDACTED]
The name it uses when it talks to other computers.

Elija un nombre de usuario: davgalle

Elija una contraseña: *****

Confirme la contraseña: *****

[Hecho]

29.

Upgrade to Ubuntu Pro [Help]

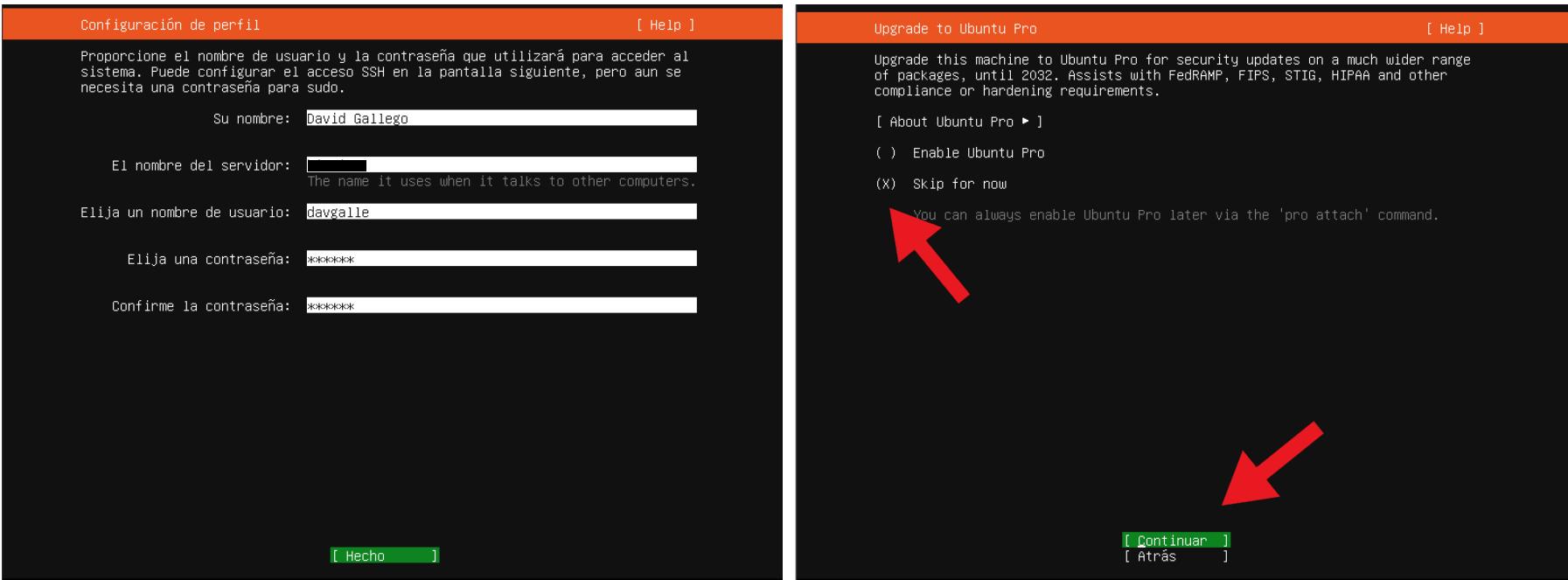
Upgrade this machine to Ubuntu Pro for security updates on a much wider range of packages, until 2032. Assists with FedRAMP, FIPS, STIG, HIPAA and other compliance or hardening requirements.

[About Ubuntu Pro ▶]

() Enable Ubuntu Pro
(X) Skip for now

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

[Continuar]
[Atrás]

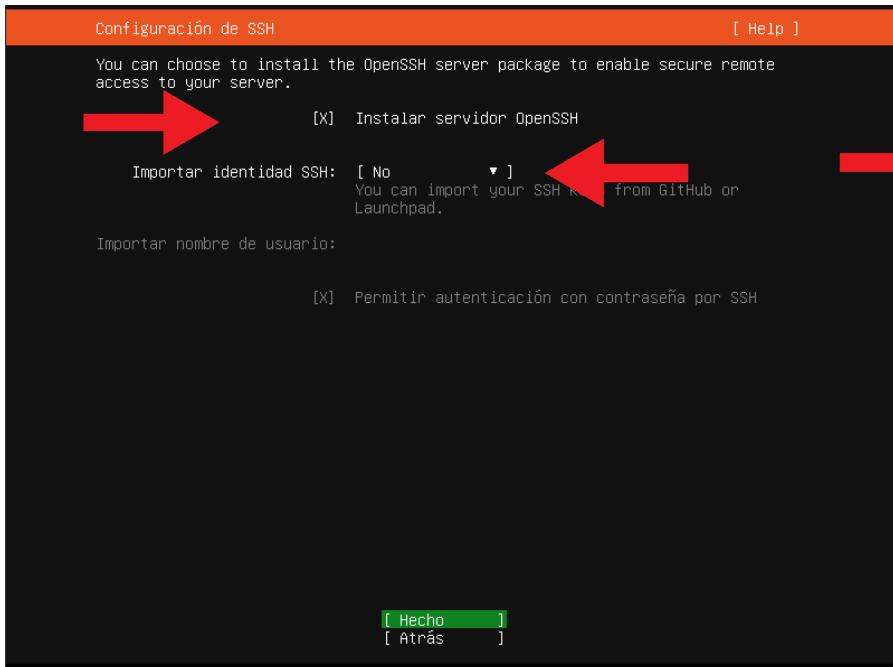


Configuración de Usuario Admin

Nombre: xxxx xxx
Nombre del servidor: xxxxxxx
Nombre Usuario: xxxxxx
Contraseña: *****
Rep. Contraseña: *****

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

30.

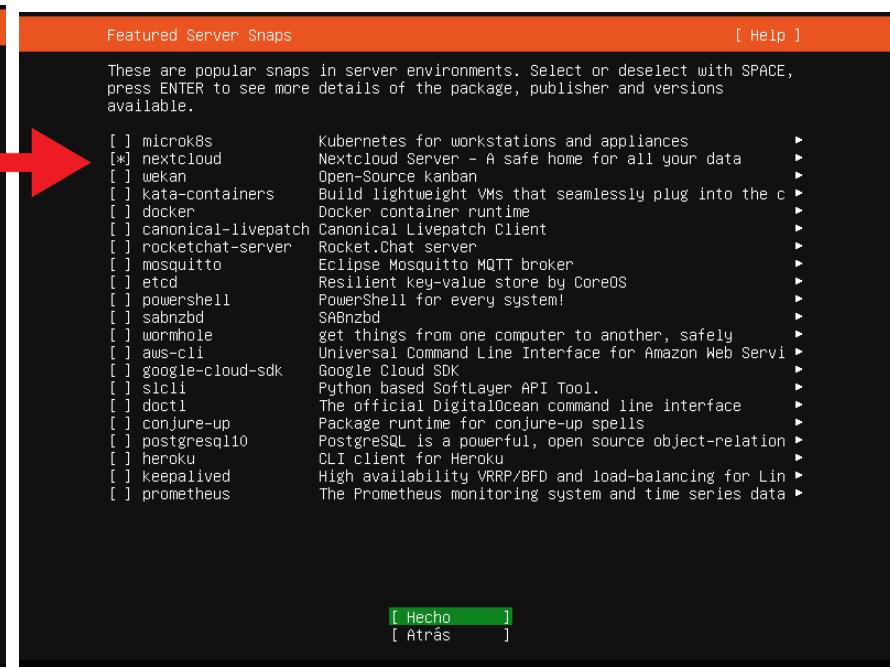


Instalamos SSH Server sin importar una identidad SSH

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.

26

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  
    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]



33.

Ubuntu 22.04.3 LTS bioaire tty1

```
bioaire login: [ 166.330069] 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 05:59:24,981 - cc_final_message.py[WARNING]: Used fallback datasource  
Hint: Num Lock on  
bioaire login: davgalle  
Password: _
```



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

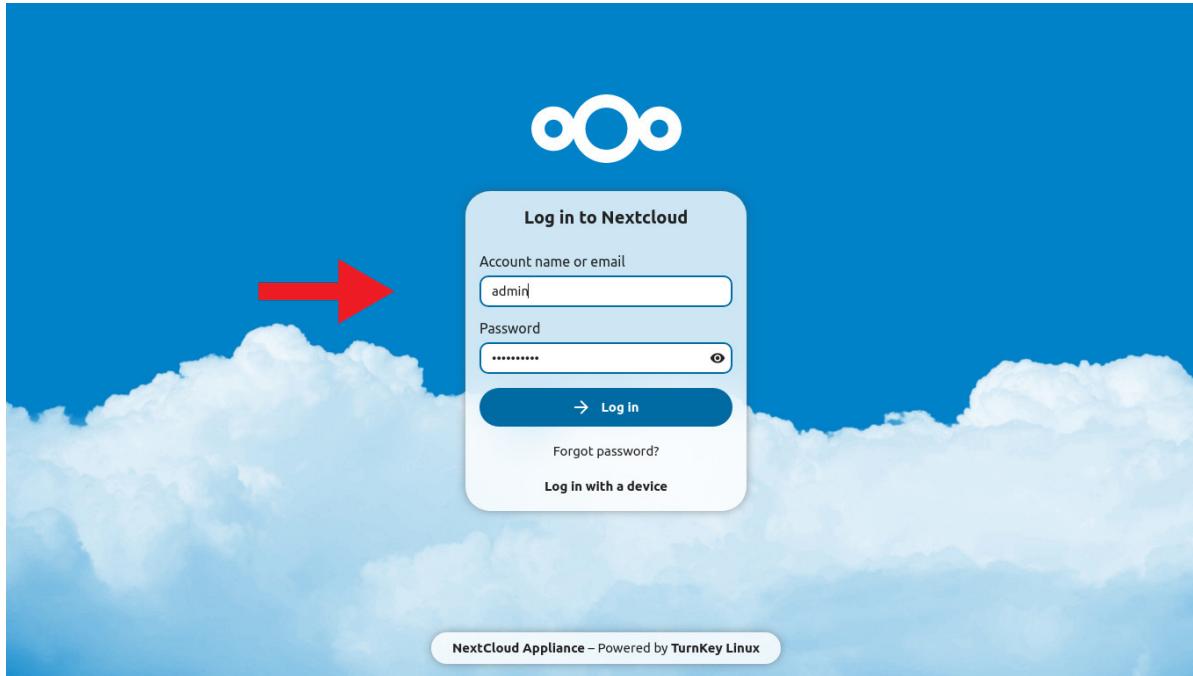
- * Antes de reboot retirar la imagen del disco:
- * Dispositivos => Unidades Ópticas => Remove from virtual drive.

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

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



29

¿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**.

The slide has a dark blue background with a faint watermark of a server tower. In the center, there is a black rounded rectangle containing white text. To the left of this rectangle is the Ubuntu logo (a red circle with a white dash) and the word 'ubuntu' in white. To the right is the NW logo (a white 'NW' inside a square). The main text reads 'Proteger el puerto SSH con UFW'.

35.

```
dvgalle@ubuntuserver:~$ 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
dvgalle@ubuntuserver:~$
```



36.

```
dvgalle@ubuntuserver:~$ ping google.es
PING google.es (142.250.201.67) 56(84) bytes of data.
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=1 ttl=116 time=4.80 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=2 ttl=116 time=4.72 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=3 ttl=116 time=4.29 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=4 ttl=116 time=4.78 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=5 ttl=116 time=4.90 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=6 ttl=116 time=4.49 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=7 ttl=116 time=4.55 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=8 ttl=116 time=5.07 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=9 ttl=116 time=4.35 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=10 ttl=116 time=4.05 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=11 ttl=116 time=5.12 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=12 ttl=116 time=5.12 ms
dvgalle@ubuntuserver:~$ ping -4 google.es
PING google.es (142.250.201.67) 56(84) bytes of data.
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=1 ttl=116 time=5.59 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=2 ttl=116 time=4.47 ms
<C>
--- google.es ping statistics ---
15 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 3.693/4.580/5.586/0.593 ms
dvgalle@ubuntuserver:~$
```

Verificamos conexiones a internet:

Verificamos IPv6:

```
dvgalle@ubuntuserver: ~$ ping google.es
```

Verificamos IPv4:

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

37.

```
davgalle@ubuntuserver:~$ sudo -i
[sudo] password for davgalle:
root@ubuntuserver:~# service ssh status
● ssh.service - OpenBSD 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:sshd(8)
          man:sshd_config(5)
    Main PID: 878 (sshd)
      Tasks: 1 (limit: 5636)
     Memory: 7.0M
        CPU: 2.026s
       Group: /system.slice/ssh.service
          ↳ 878 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct 24 17:55:21 ubuntuserver sshd[358863]: Accepted password for davgalle from 192.168.1.1 port 58896 ssh2
oct 24 17:55:21 ubuntuserver sshd[358863]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
oct 24 20:35:45 ubuntuserver sshd[374849]: error: key_exchange_identification: banner line contains invalid characters
oct 24 20:35:45 ubuntuserver sshd[374849]: banner exchange: Connection from 167.94.145.109 port 55620: invalid format
oct 24 20:36:10 ubuntuserver sshd[374863]: Connection closed by 167.94.145.109 port 41976 [preauth]
oct 25 10:32:38 ubuntuserver sshd[460552]: Accepted password for davgalle from 195.55.210.136 port 42303 ssh2
oct 25 10:32:38 ubuntuserver sshd[460552]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
oct 25 12:38:18 ubuntuserver sshd[474180]: Connection closed by 45.84.89.3 port 36316 [preauth]
oct 25 15:48:40 ubuntuserver sshd[492366]: Accepted password for davgalle from 192.168.1.1 port 49905 ssh2
oct 25 15:48:40 ubuntuserver sshd[492366]: pam_unix(sshd:session): session opened for user davgalle(uid=1000) by (uid=0)
root@ubuntuserver:~# grep "Port" /etc/ssh/sshd_config
Port 4242
root@ubuntuserver:~#
```

**Deberíamos de tener instalado sudo.
Accedemos como usuario root.**

root:

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

Mostramos el estado de ssh:

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

Mostramos el puerto sshd:

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

38.

```
# This ssh was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/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 4242
#AddressFamily Inherit
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_dsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# 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
```

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

Modificamos el archivo sshd_config:

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

```
#Port 22 -> Port 4242
#PermitRootLogin prohibit-password -> PermitRootLogin no
```

39.

```
root@ubuntuserver: ~
# 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
# BatchMode 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-md5-96-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1G 1h
# UserKnownHostsFile ~/.ssh/known_hosts.d/%k
SendEnv LANG LC_
HashKnownHosts yes
GSSAPIAuthentication yes
"/etc/ssh/ssh_config" 53L, 1648B
```

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@ubuntuserver: sudo service ssh restart
```

32

40.

```
root@ubuntuserver: ~
root@ubuntuserver: ~# ufw status
Status: active
To           Action      From
--          ----       --
root@ubuntuserver: ~#
```



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

Verificamos, instalamos y habilitamos:

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

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

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

41.

```
root@ubuntuserver:~# 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@ubuntuserver:~#
```



Tenemos que habilitar los puertos en el firewall.

Habilitamos el puerto ssh:

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

Habilitamos 80 y 443 de http y https:

```
davgalle@ubuntuserver: sudo ufw allow 80, 443/tcp
```

42.

```
davgalle@ubuntuserver:~# ssh davegallegogomez@192.168.1.42 -p 4242
password:
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/pro

System information as of vie 25 oct 2024 16:46:20 UTC

System load: 0.87 Processes: 141
Usage of /: 3.0% of 1.8TB Users logged in: 1
Memory usage: 15% IPv4 address for empp0s3: 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 Applications está desactivado
Se pueden aplicar 20 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm 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 15:48:41 2024 from 192.168.1.1
davgalle@ubuntuserver:~$
```

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

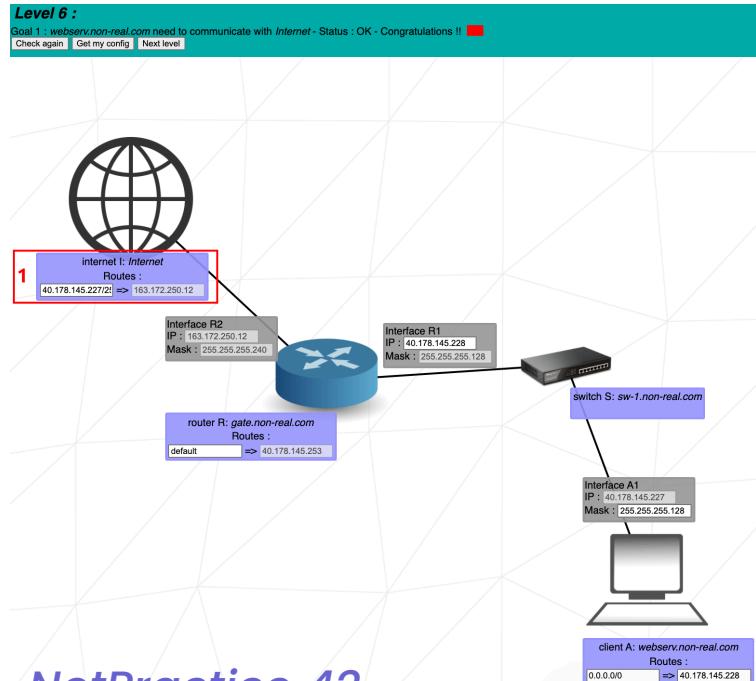
```
ssh nombreusuario@192.168.1.42 -p 4242
```

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

```
davgalle@ubuntuserver: ~
```

```
),
'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' => 'oc1ptz4csrwq',
'passwordsalt' => '',
'secret' => '1a083f9e0393b8c940972d4a2a2a2a2a',
'trusted_domains' =>
array (
  0 => '████████',
  1 => '████████',
  2 => '████████',
),
'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' => '',
'dbtableprefix' => 'oc_',
'mysql.utf8mb4' => true,
'dbuser' => 'nextcloud',
'dbpassword' => '████████',
'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)
```

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
X	cloud	TCP	80	80	192.168.1.42	<input checked="" type="checkbox"/>
X	https	TCP	443	443	192.168.1.42	<input checked="" type="checkbox"/>
X	Exter	TCP	4242	4242	192.168.1.42	<input checked="" type="checkbox"/>
X	NUEVA	TCP	90	90	192.168.1.13	<input checked="" type="checkbox"/>

Mapeamos los puertos para http, https y ssh

Nombre:	http	https
Dirección IP:	192.168.1.43	192.168.1.43
Protocolo:	TCP	TCP
WLAN:	80	443



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:

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

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 configuaremos las DNS para que apunten a la dirección IP de nuestro Router.

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

<input type="checkbox"/>	A	letter	Apunta a: [REDACTED]	900	Editar registro	
<input type="checkbox"/>	A	smtp	Apunta a: [REDACTED]	900	Editar registro	
<input type="checkbox"/>	A	cloud	Apunta a: [REDACTED]	900	Editar registro	
<input type="checkbox"/>	A	imap	Apunta a: [REDACTED]	900	Editar registro	
<input type="checkbox"/>	A	gestiona	Apunta a: [REDACTED]	900	Editar registro	

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
```

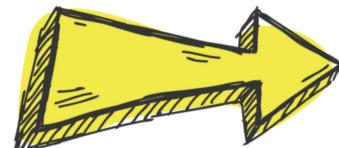
```
(Y)es / (N)o => N
```

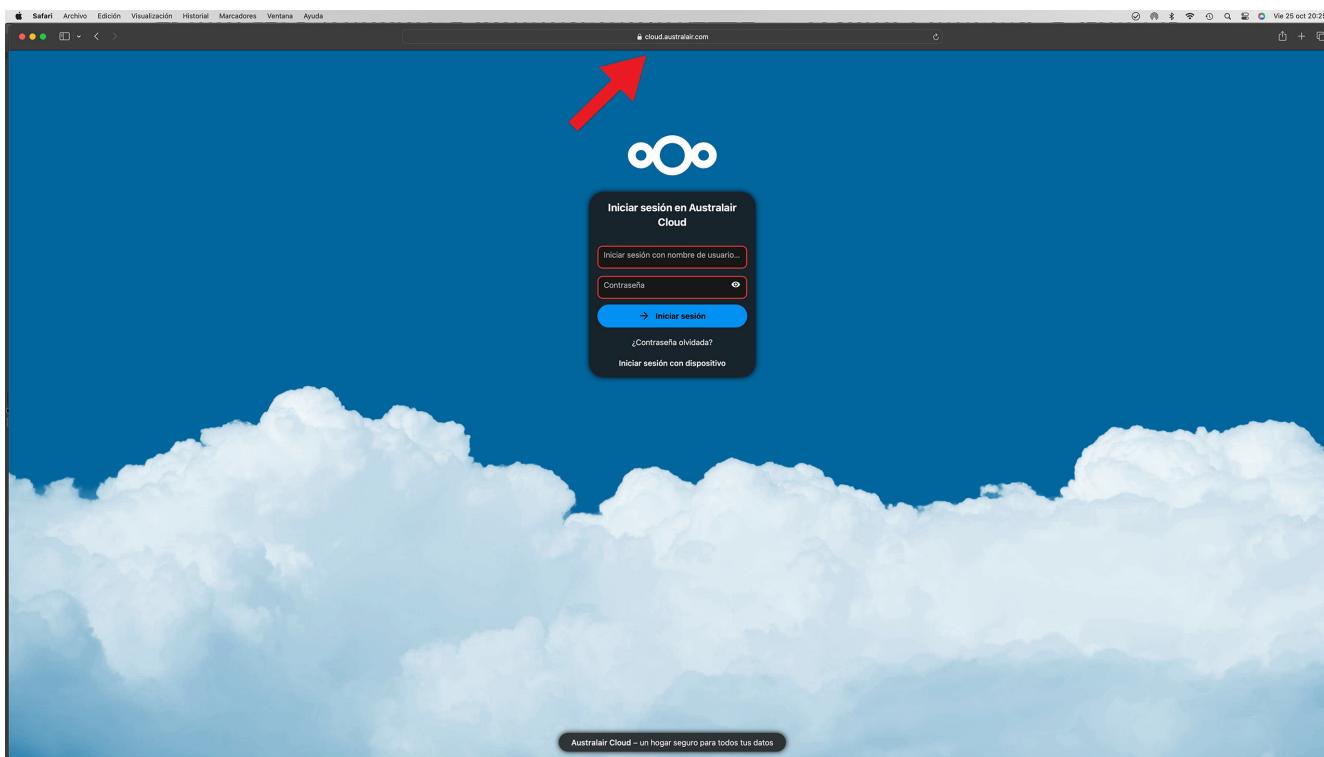
```
Select the appropriate number[ 1 - 2 ] => 2
```

Reiniciaremos apache.

```
sudo systemctl restart apache2
```

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





Cloud Server Project

Implementa un servicio cloud en opensource

¡Muchas gracias!



David G. & Paolo S.

