

Entrega Tema 2

24/10/2023

Fernando José

2 DAM / SGE

Índice

1. Preparación

Preparación de la máquina	3
Instalación del SO	6
Activación del root	15
Conexión	16
Redireccionamiento de puertos	19

2.Instalación

Instalación	20
Arranque.....	27
Error durante el arranque	28
Web dependencys.....	29

3.Configuración

Ficheros log.....	31
Fichero de configuración	32
Arranque automático de Odoo.....	33

4. Entrada al sistema

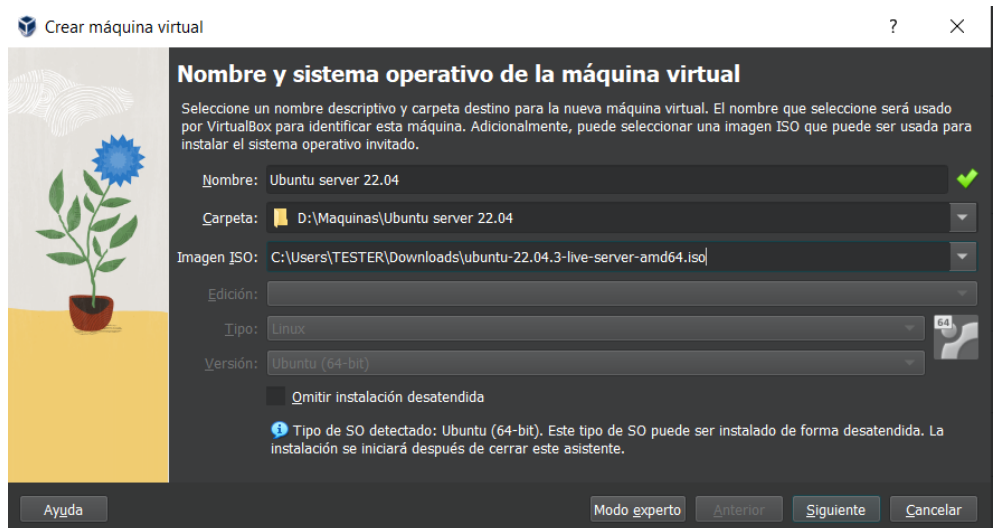
Entrada al sistema.....	35
-------------------------	----

1.1

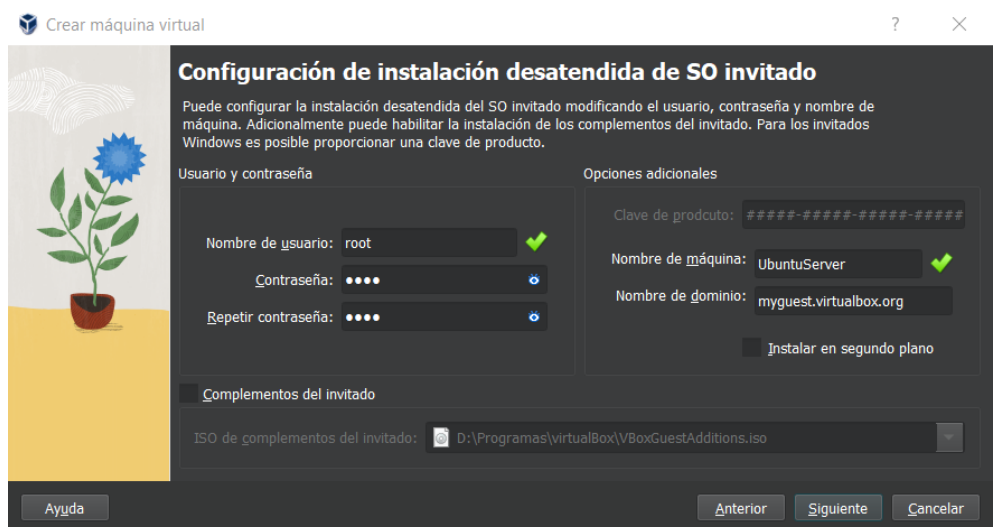
Preparación de la máquina

Antes de comenzar con la instalación del sistema operativo tendremos que crear la máquina virtual.

Para ello nos iremos a VirtualBox y le daremos a crear una nueva, allí indicaremos el nombre, la carpeta donde se almacenará y la imagen ISO.



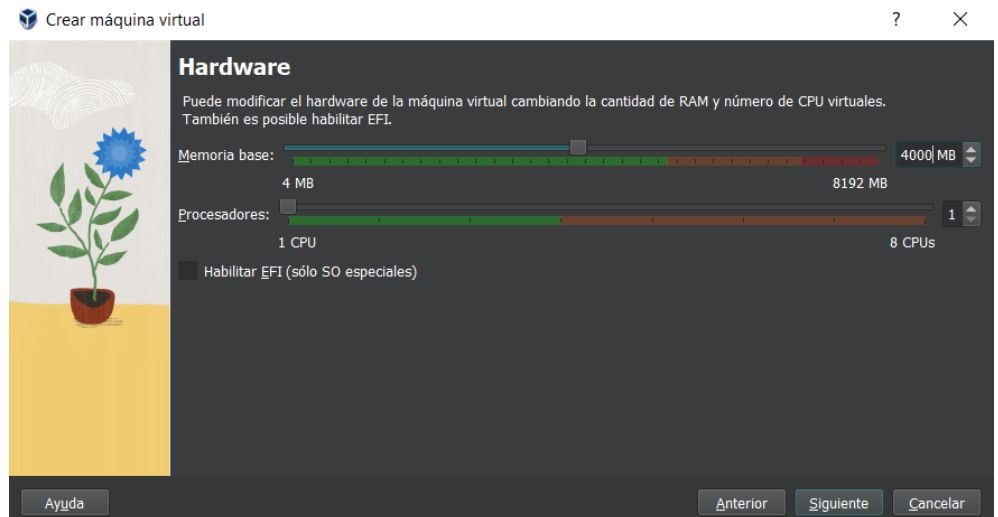
Después introduciremos el usuario, contraseña y nombre que le queremos poner a nuestra máquina.



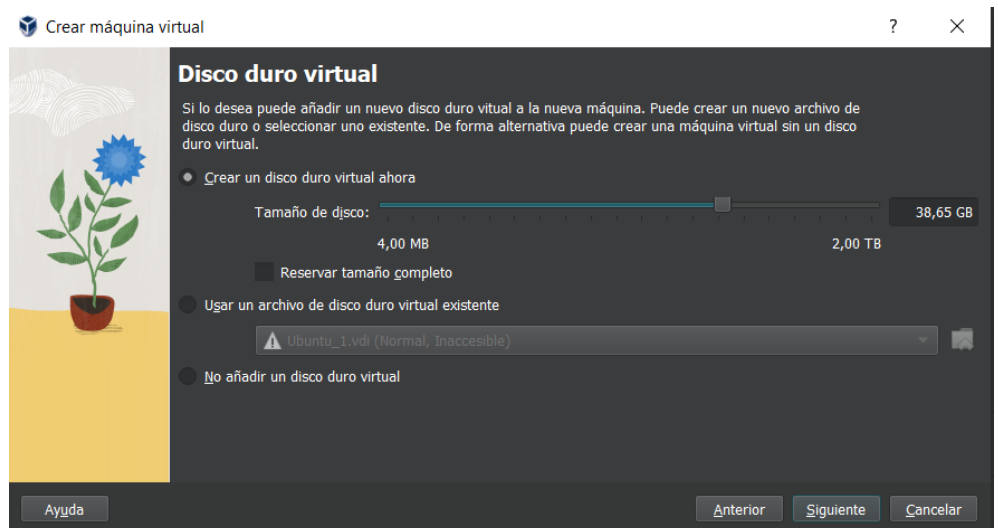
1.1

Preparación de la máquina

Después seleccionaremos la cantidad de memoria RAM y el número de procesadores para la máquina.



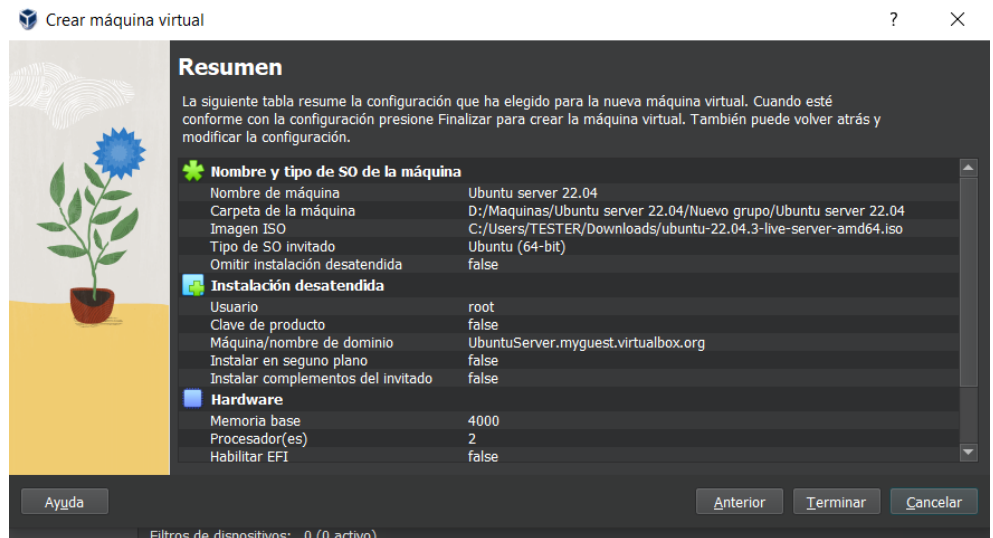
Después seleccionaremos el tamaño del disco que tendrá.



1.1

Preparación de la máquina

Por último, nos aparecerá un resumen de las opciones que hemos seleccionado y cuando le demos a terminar se creará la máquina.

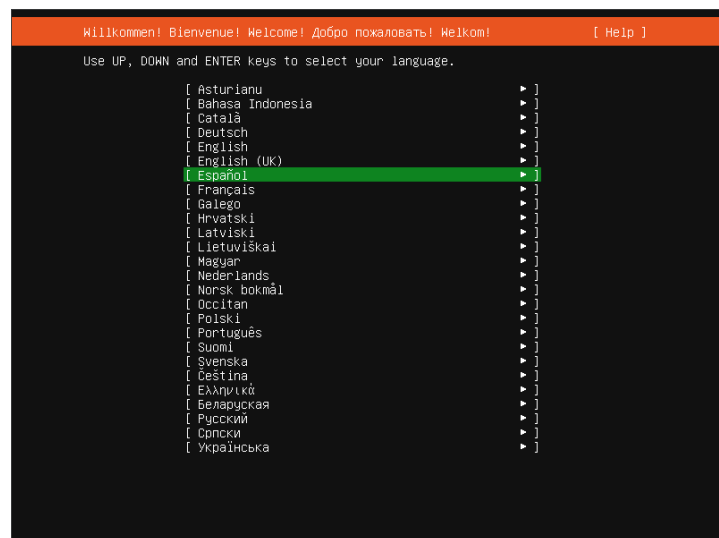


1.2

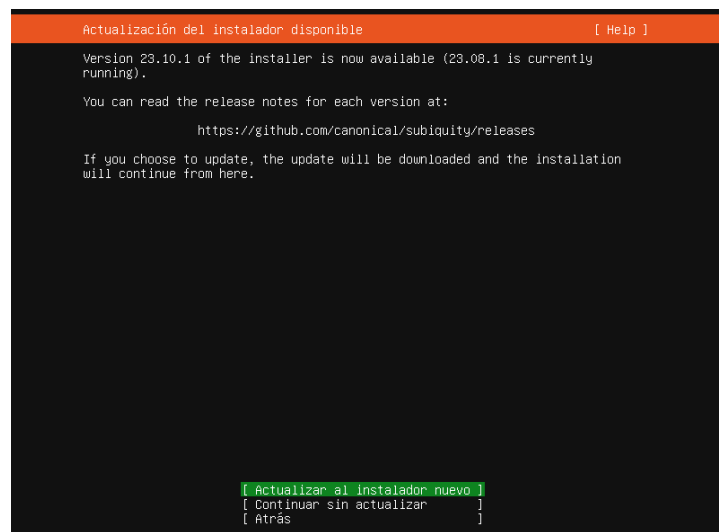
Instalación del SO

Ahora procederemos con la instalación del sistema operativo, en concreto Ubuntu Server.

Como ya hemos puesto la ISO en el apartado anterior solo tenemos que iniciar la máquina para que se nos abra el instalador. El primer paso será seleccionar el idioma.



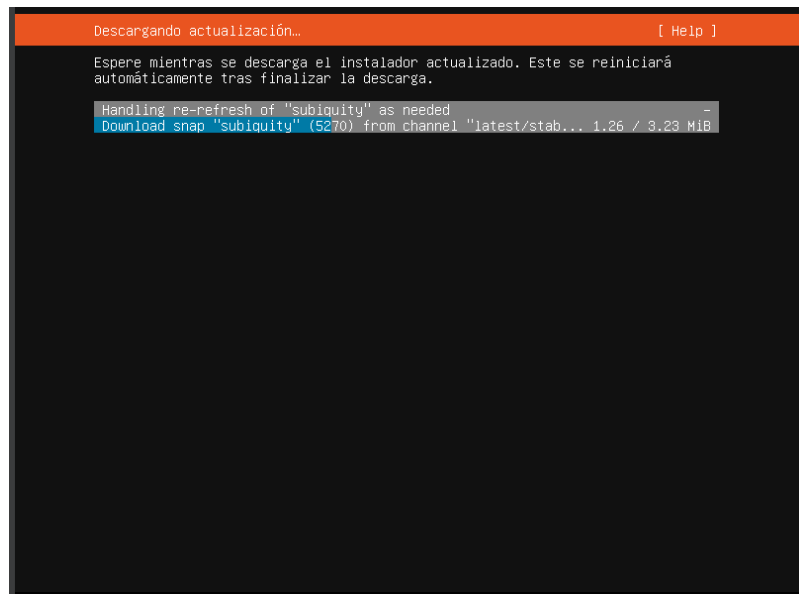
Después nos indica que hay una actualización disponible. Le podemos dar a actualizar ahora o pasar este paso, aunque es recomendable siempre actualizarlo.



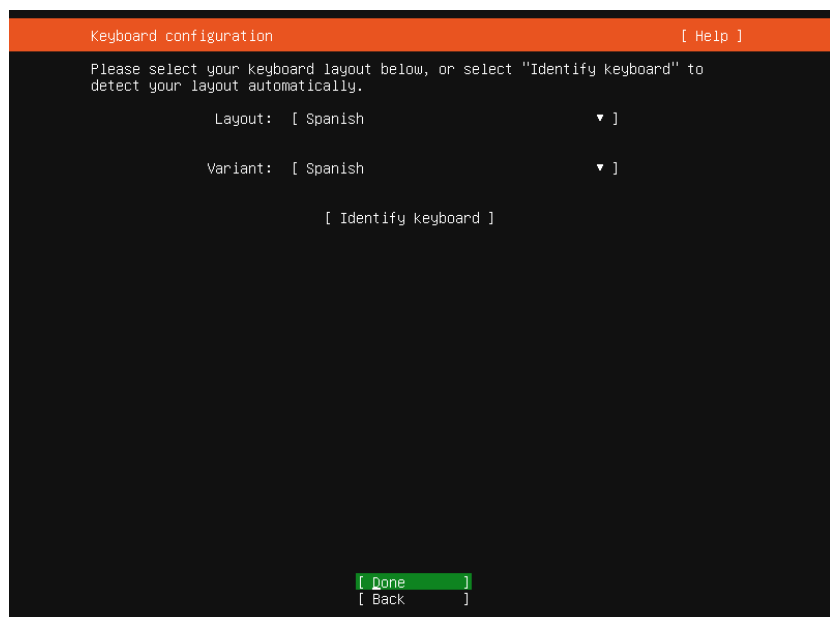
1.2

Instalación del SO

Si le hemos dado a actualizar en el paso anterior, el instalador comenzara a actualizarse.



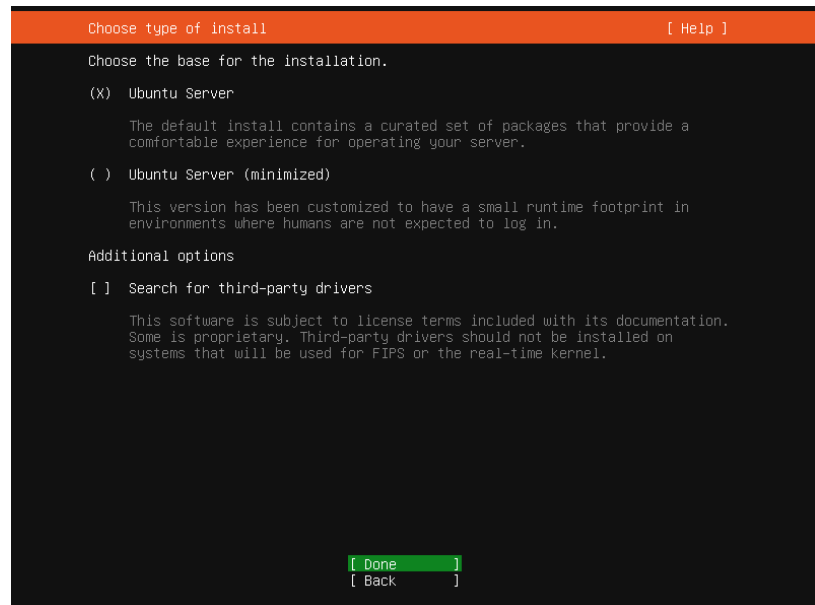
Una vez que haya finalizado la actualización procederemos a seguir con la instalación del sistema operativo. En el siguiente paso seleccionaremos la configuración del teclado.



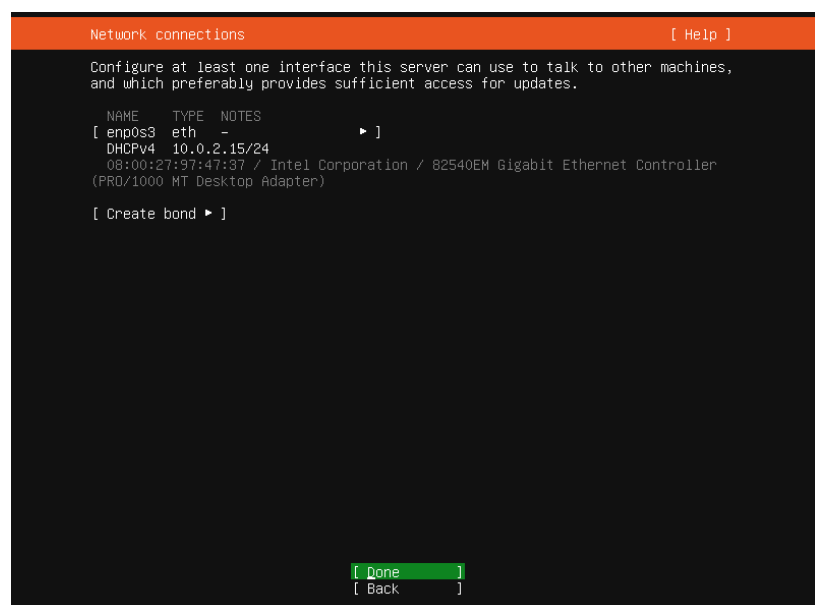
1.2

Instalación del SO

Después nos pedirá el tipo de instalación nosotros dejaremos la opción por defecto para instalar Ubuntu Server.



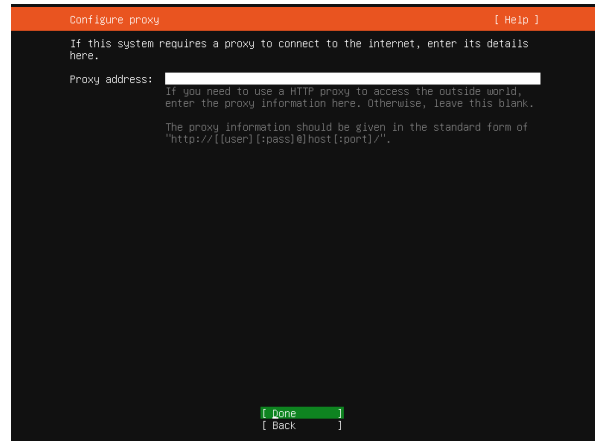
Ahora nos aparecerá para configurar las conexiones de red. Volveremos a dejar la opción por defecto



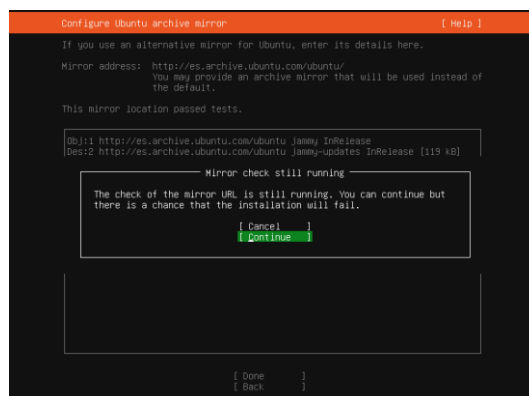
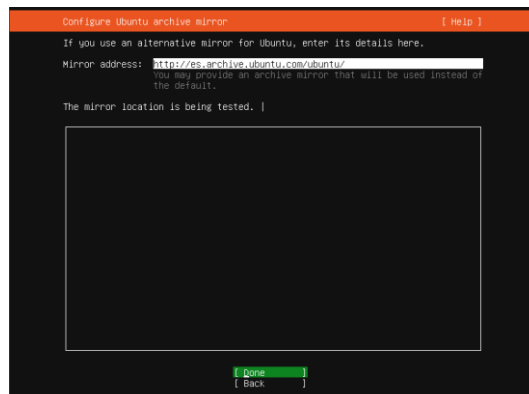
1.2

Instalación del SO

Ahora configuraremos el proxy, lo dejaremos el que está por defecto.



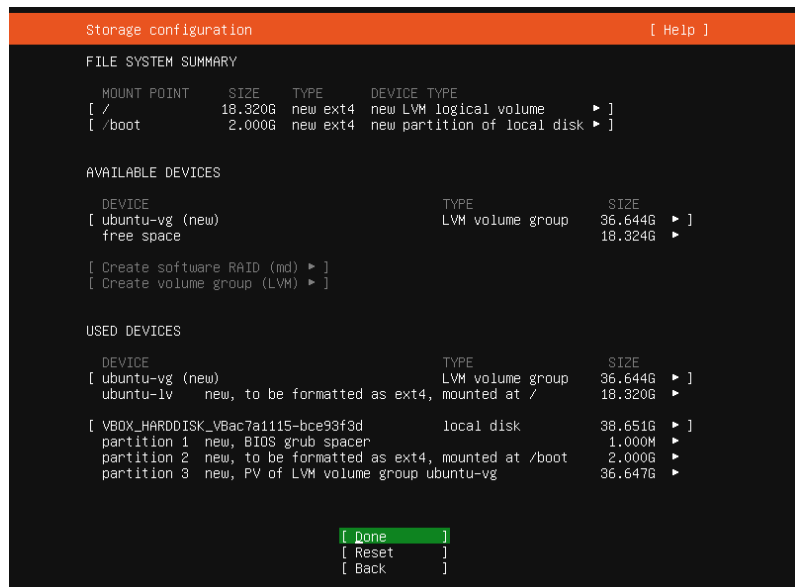
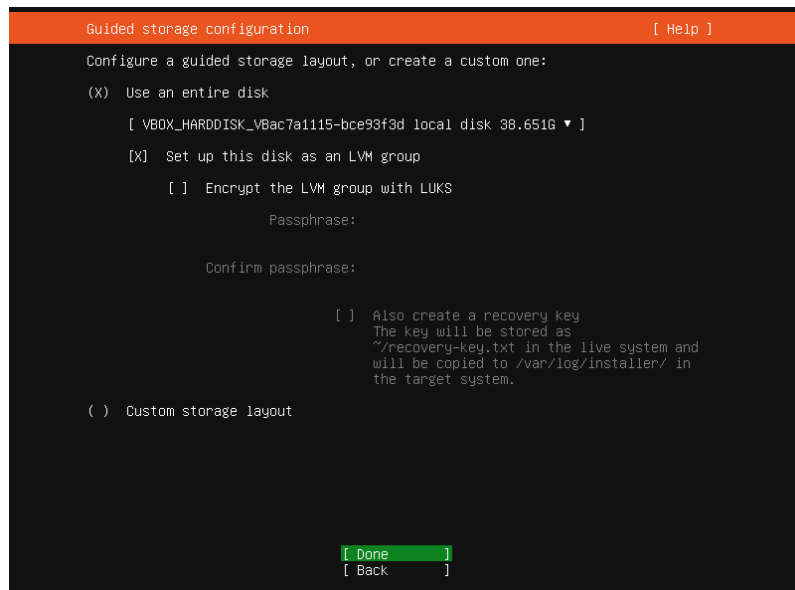
Después nos dirá si queremos seleccionar un servidor o repositorio específico de donde se descargarán los paquetes y actualizaciones del sistema operativo. Nosotros dejaremos el que hay por defecto.



1.2

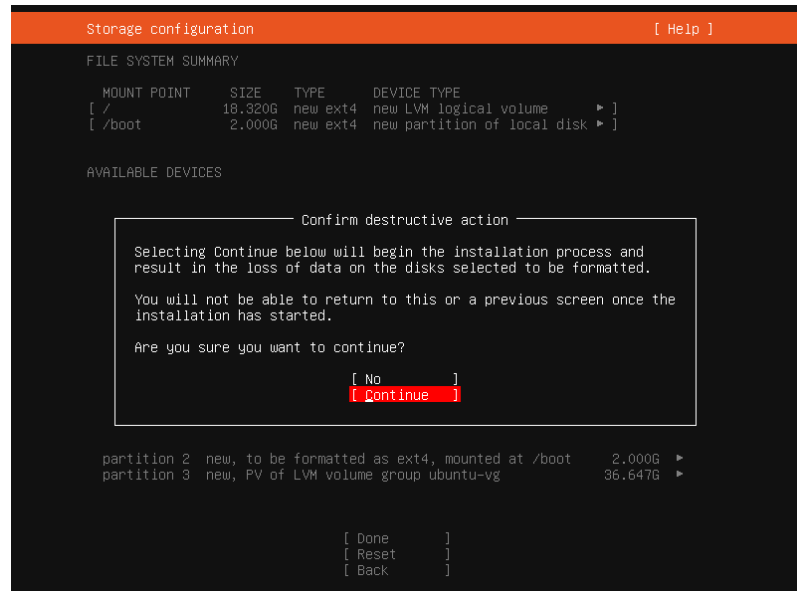
Instalación del SO

En el siguiente paso nos pedirá la configuración del almacenamiento, como en el resto de los pasos dejaremos las distintas opciones por defecto.

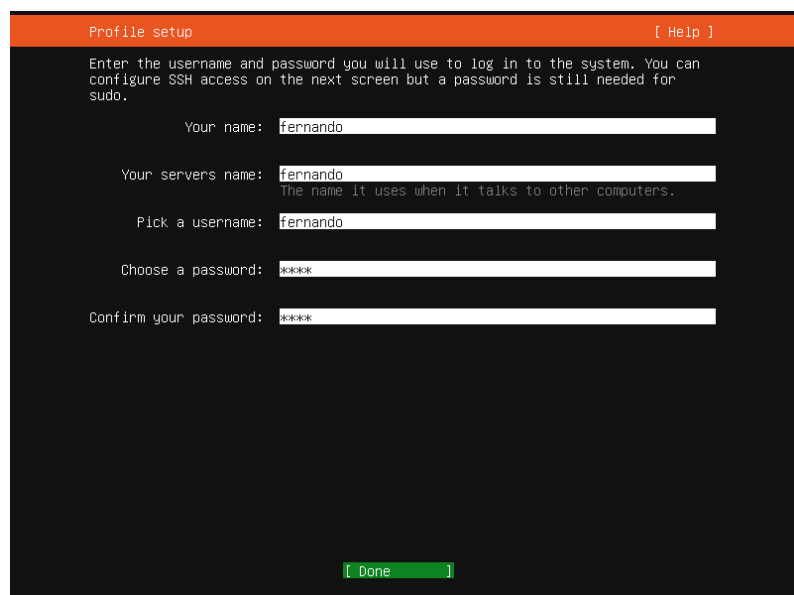


1.2

Instalación del SO



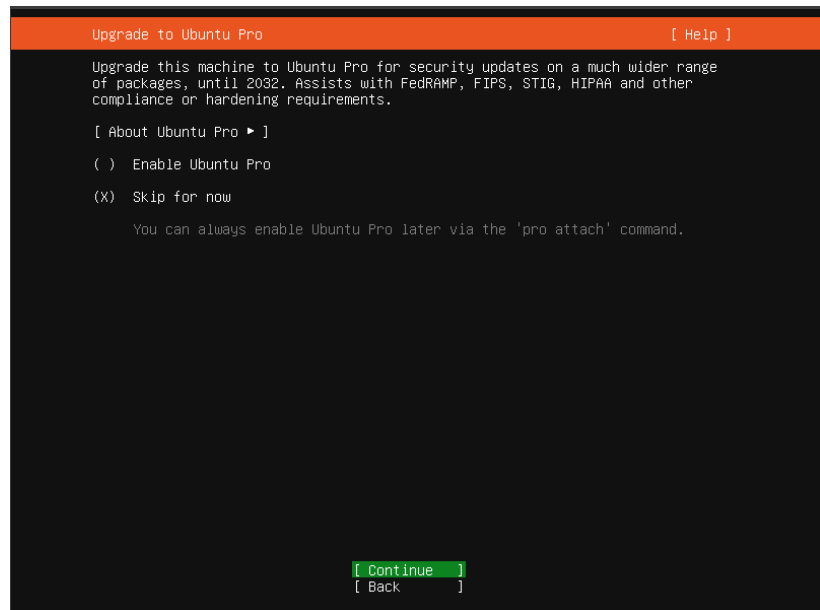
Una vez que hemos finalizado la configuración del almacenamiento. El siguiente paso sera establecer el perfil (nombre de usuario, nombre del servidor, contraseña...)



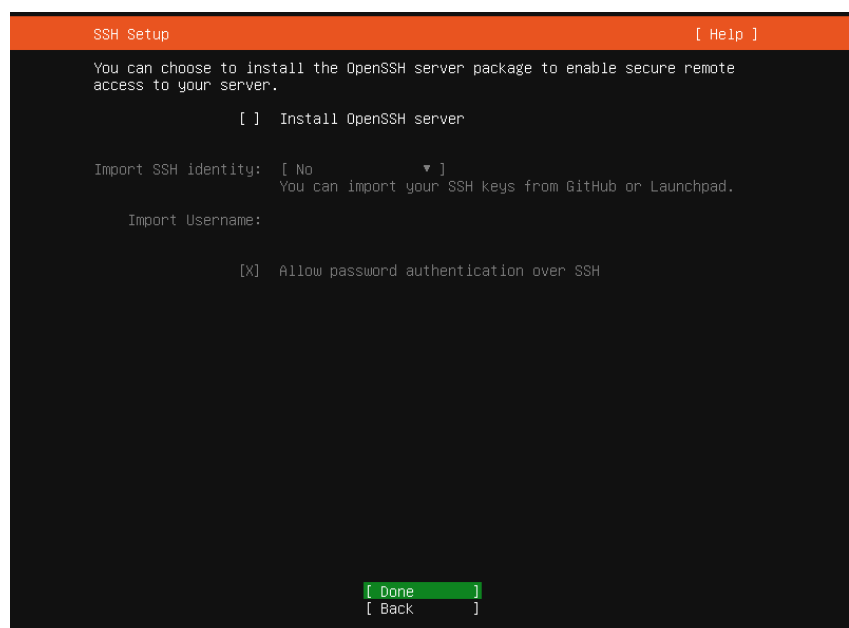
1.2

Instalación del SO

Después nos pedirá que versión queremos instalar, el Ubuntu pro o el normal. Nosotros seleccionaremos el normal para ello le daremos a skip



Ahora nos pedirá si queremos instalar ssh, nosotros lo dejaremos vacío por defecto porque de momento no lo necesitamos



1.2

Instalación del SO

A continuación, nos aparecerá una lista de servicios que se nos recomienda instalar. Para instalar alguno lo marcaremos y después le daremos a done, pero como no necesitamos ninguno los dejaremos vacíos.

```
Featured Server Snaps [ Help ]

These are popular snaps in server environments. Select or deselect with SPACE,
press ENTER to see more details of the package, publisher and versions
available.

[ ] microk8s      Kubernetes for workstations and appliances
[ ] nextcloud     Nextcloud Server - A safe home for all your data
[ ] wekan         The open-source kanban
[ ] kata-containers Build lightweight VMs that seamlessly plug into the c
[ ] docker        Docker container runtime
[ ] canonical-livepatch Canonical Livepatch Client
[ ] rocketchat-server Rocket.Chat server
[ ] mosquitto     Eclipse Mosquitto MQTT broker
[ ] etcd          Resilient key-value store by CoreOS
[ ] powershell   PowerShell for every system!
[ ] sabnzbd       SABnzbd
[ ] wormhole      get things from one computer to another, safely
[ ] aws-cli       Universal Command Line Interface for Amazon Web Servi
[ ] google-cloud-sdk Google Cloud SDK
[ ] slcli         Python based SoftLayer API Tool.
[ ] doctl         The official DigitalOcean command line interface
[ ] conjure-up    Package runtime for conjure-up spells
[ ] postgresql10 PostgreSQL is a powerful, open source object-relat
[ ] heroku        CLI client for Heroku
[ ] keepalived    High availability VRRP/BFD and load-balancing for Lin
[ ] prometheus    The Prometheus monitoring system and time series data
[ ] juju          Juju - a model-driven operator lifecycle manager for

[ Done ]
[ Back ]
```

Una vez que hemos acabado de configurar todas las opciones, por fin empezara la instalación. Después de que haya finalizado tendremos que reiniciar la máquina.

```
Install complete! [ Help ]

executing curtin install curthooks step
curtin command install
  configuring installed system
    running 'curtin curthooks'
      curtin command curthooks
        configuring apt
        installing missing packages
        Installing packages on target system: ['grub-pc']
        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
  calculating extra packages to install
  configuring cloud-init
  downloading and installing security updates
  curtin command in-target
  restoring apt configuration
  curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

1.2

Instalación del SO

Una vez que se haya reiniciado ya podremos iniciar sesión con el usuario que hemos puesto durante la configuración y usar el sistema operativo.

```
Starting Execute cloud user/final scripts...
Starting Record Runlevel Change in UTMF...
[ OK ] Finished Record Runlevel Change in UTMF.
[ 43.114319] cloud-init[1350]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 running 'modules:final' at Sa
t, 21 Oct 2023 14:35:23 +0000. Up 43.03 seconds.
ci-info: no authorized SSH keys fingerprints found for user fernando.
<14>Oct 21 14:35:24 cloud-init: #####
<14>Oct 21 14:35:24 cloud-init: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Oct 21 14:35:24 cloud-init: 1024 SHA256:ucr91TjrxIKrkGUF/8a1QRhJBQhLXaWVnq53CAcVWJI root@fernand
o (DSA)
<14>Oct 21 14:35:24 cloud-init: 256 SHA256:cItWBygIe8Dn5P7QKrYjg2atacyBAPRbj4hbel4562k root@fernando
(ECDSA)
<14>Oct 21 14:35:24 cloud-init: 256 SHA256:EXTfXRjJpCsujqJKI+XBT3F8LiQ4cYr8V8pVdr/q5w root@fernando
(ED25519)
<14>Oct 21 14:35:24 cloud-init: 3072 SHA256:TTdVb2dxcidzYt9W5+f7XN10p70v17RfuChfRk/0BqM root@fernand
o (RSA)
<14>Oct 21 14:35:24 cloud-init: -----END SSH HOST KEY FINGERPRINTS-----
<14>Oct 21 14:35:24 cloud-init: #####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBLfZ1v5gGWV4X9PdvSGgREHAo0F0
s2vMmtK/z1qGy2mRA7C5P1kaJ0GnWAYySvGeF98WVh/mUda06bY5954GcTU= root@fernando
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI0mfkZoSbokWfcC4Po99xjToDc3qSMg7d37Na+z0+w5p root@fernando
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDbalA00fb20yBJ6xo75Dwr+rHiHzGqSR4w4ChDZ5838ucZhu5B3owI0zLWJB2q
B1UEuCeGBJTSr8CB151JHGdFT/aXTDj1hYVQ+Szci1FJquN3/eIILX4k8earMu3o2e7zQmz25XP0auhF2AVSNJn/wBaYoVnH30qvu
Hyy312M0K6/R8GzGspJ0pyzuf4BX206zF4EypnkAbbdrrkKh0rXsXs29Dg6/xD7e9mvr+8yg443CzRXFI4E9Jr4w/AC7i0FKd0B1
31Tw+8bWAD/kyyblomzdXN51KepxDPOV/KK7JJZPT+h3MFgtGIUn5ar0nQmOUJ7FCLx2QaETN8CEakIGBkxp5TR16G8M+p+q2Cum
o15bLcbgeqNc0ayxn5W53RPy.jxdj7E/rzzE8rCKi0cg5VEDicxxNf+h71xAow/oe7Jxu3MzF4U2JQ6xIB//UFKU0D3Fpttf0zT/
HSw0ZPfFy02/hwyi622WkXS6t7mMSnvFkME2FLanCvxL/K7s3Ps= root@fernando
-----END SSH HOST KEY KEYS-----
[ 43.346955] cloud-init[1350]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 finished at Sat, 21 Oct 2023
14:35:24 +0000. Datasource DataSourceNone. Up 43.33 seconds
[ 43.348474] cloud-init[1350]: 2023-10-21 14:35:24,235 - cc_final_message.py[WARNING]: Used fallback
ck datasource
[ OK ] Finished Execute cloud user/final scripts.
[ OK ] Reached target Cloud-init target.

fernando login:
```

1.3

Activación del root

Para poder iniciar sesión con el usuario root tendremos que establecerle una contraseña para ello usaremos el comando: `sudo passwd root`

Para entrar en el usaremos el comando `su root` y para salir `exit`

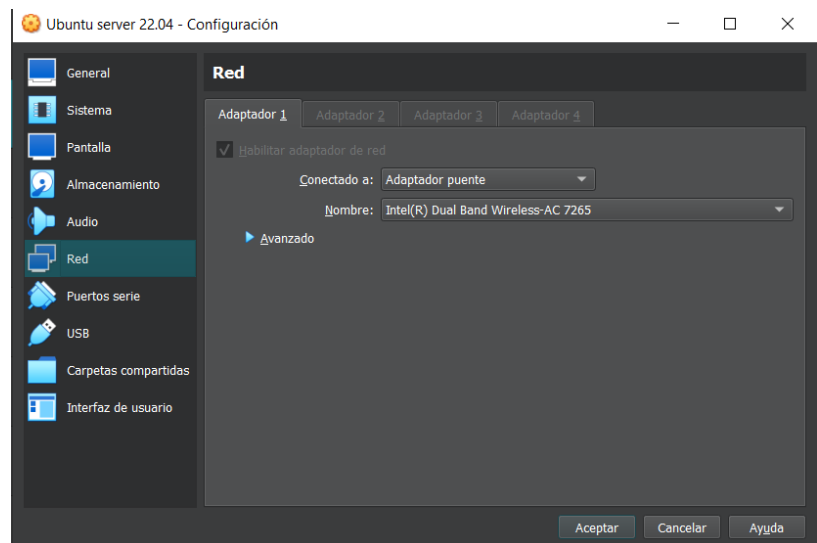
```
fernando@fernando:~$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
fernando@fernando:~$ su root
Password:
root@fernando:/home/fernando# exit
exit
fernando@fernando:~$ _
```

1.4

Conexión

Ahora estableceremos la conexión entre nuestra computadora y la máquina virtual.

El primer paso será establecer un adaptador puente para que tengan conexión. Esto lo podemos hacer en las opciones de la máquina virtual.



Después tendremos que modificar el fichero netplan de la siguiente forma. Tendremos que poner la IP de la red y la IP que queremos. Modificaremos el fichero con el comando `sudo nano /etc/netplan/00-installer-config.yaml`

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernet:
    enp0s3:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.1.228/24]
      gateway4: 192.168.1.0
      nameservers:
        addresses: [192.168.1.1]
```


Una vez que hemos modificado el fichero netplan tenemos que aplicar los cambios con el comando: `sudo netplan apply`

```
fernando@fernando:~$ sudo netplan apply
** (generate:2239): WARNING **: 19:59:19.442: `gateway4` has been deprecated, use default routes instead.
See the 'Default routes' section of the documentation for more details.
** (process:2237): WARNING **: 19:59:20.160: `gateway4` has been deprecated, use default routes instead.
See the 'Default routes' section of the documentation for more details.
** (process:2237): WARNING **: 19:59:20.176: `gateway4` has been deprecated, use default routes instead.
See the 'Default routes' section of the documentation for more details.
** (process:2237): WARNING **: 19:59:20.342: `gateway4` has been deprecated, use default routes instead.
See the 'Default routes' section of the documentation for more details.
** (process:2237): WARNING **: 19:59:20.342: `gateway4` has been deprecated, use default routes instead.
See the 'Default routes' section of the documentation for more details.
fernando@fernando:~$
```

Después podemos usar el `comando ifconfig -a` para saber si la configuración de la red se ha aplicado.

```
fernando@fernando:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.228 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe97:4737 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:97:47:37 txqueuelen 1000 (Ethernet)
    RX packets 872 bytes 575173 (575.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 367 bytes 43190 (43.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

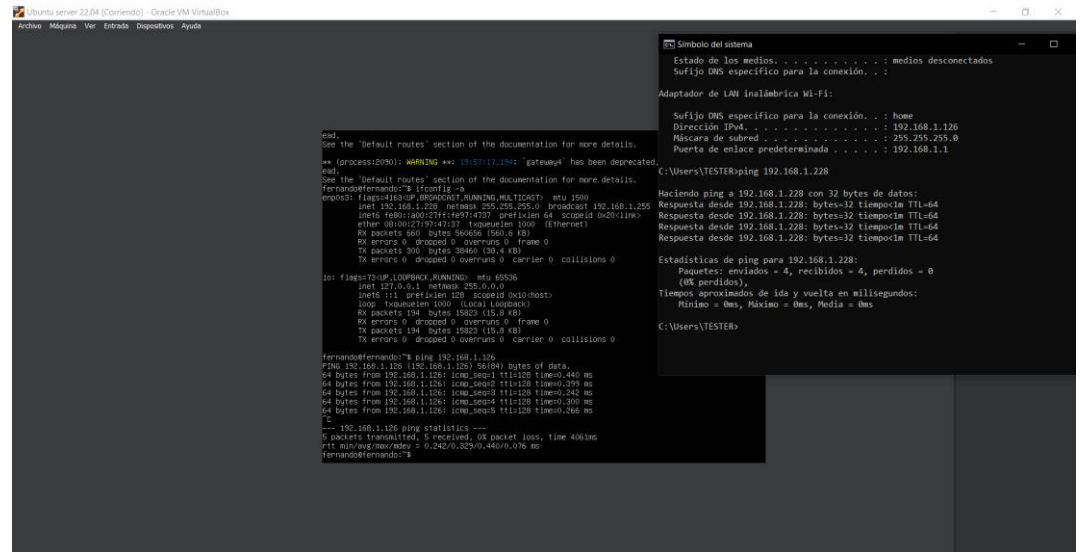
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 207 bytes 17063 (17.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 207 bytes 17063 (17.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

fernando@fernando:~$ _
```

1.4

Conexión

Ahora probaremos la conexión entre la maquina y nuestro ordenador usando el comando **ping**



The screenshot shows a terminal window with the following content:

```
ed,
See the 'default routes' section of the documentation for more details.
++ (Process:2090): WARNING ++: 192.168.1.194: 'gateway4' has been deprecated.
++
See the 'default routes' section of the documentation for more details.
fernando@fernando:~$ ifconfig -a
emp03: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.228 netmask 255.255.255.0 broadcast 192.168.1.255
    inet fe80::800:27ff:fe97:4797 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:97:47:97 txqueuelen 1000 (Ethernet)
    RX packets 660 bytes 560636 (560.6 kB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 360 bytes 58660 (58.6 kB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet ::1 prefixlen 128 scopeid 0x10<host>
    loop 0 txqueuelen 1000 (Local Loopback)
    RX packets 194 bytes 15823 (15.8 kB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 194 bytes 15823 (15.8 kB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

fernando@fernando:~$ ping 192.168.1.126
PING 192.168.1.126 (192.168.1.126) 56(84) bytes of data:
64 bytes from 192.168.1.126: icmp_seq=1 ttl=64 time=0.440 ms
64 bytes from 192.168.1.126: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 192.168.1.126: icmp_seq=3 ttl=64 time=0.242 ms
64 bytes from 192.168.1.126: icmp_seq=4 ttl=64 time=0.260 ms
64 bytes from 192.168.1.126: icmp_seq=5 ttl=64 time=0.266 ms
^C
--- 192.168.1.126 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 406ms
rtt min/avg/max/mdev = 0.242/0.329/0.440/0.076 ms
fernando@fernando:~$
```

On the right side of the terminal, there is a window titled 'Símbolo del sistema' showing network status:

```
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . : home
Dirección IPv4. . . . . : 192.168.1.126
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . : 192.168.1.1
```

Below this, it shows the command being executed in the terminal:

```
C:\Users\TESTER>ping 192.168.1.228
```

And the output of the ping command:

```
Se inició ping a 192.168.1.228 con 32 bytes de datos:
Respuesta desde 192.168.1.228: bytes=32 tiempo=64 ms TTL=64
Respuesta desde 192.168.1.228: bytes=32 tiempo=64 ms TTL=64
Respuesta desde 192.168.1.228: bytes=32 tiempo=64 ms TTL=64
Respuesta desde 192.168.1.228: bytes=32 tiempo=64 ms TTL=64

Estadísticas de ping para 192.168.1.228:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 6ms, Máximo = 6ms, Medio = 6ms

C:\Users\TESTER>
```

1.5

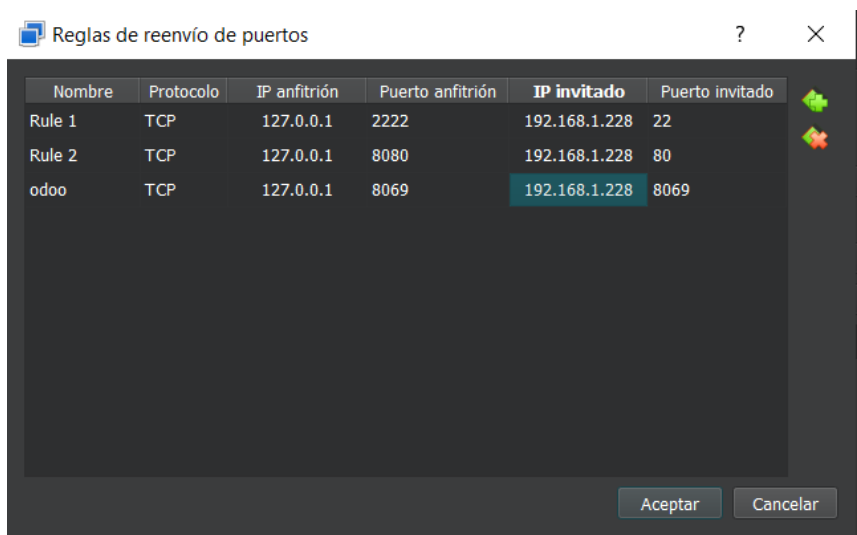
Redireccionamiento de puertos

El redireccionamiento de puertos se usa para permitir el acceso a servicios dentro de una máquina virtual como si fueran servicios locales en el sistema anfitrión. Esto facilita el acceso a servicios como Odoo, PostgreSQL o SSH sin necesidad de utilizar la dirección IP de la máquina virtual.

Para ello tendremos que cambiar el adaptador a NAT dentro de las opciones de red de nuestra máquina.

Después para establecer las reglas de reenvío de puertos tendremos que entrar al apartado de opciones avanzadas de la red de la maquina y después entrar en reenvío de puertos.

- **La primera regla** redirige el puerto 8080 del sistema anfitrión al puerto 80 de la máquina virtual, lo que permite el acceso a un servicio web
- **La segunda regla** redirige el puerto 2222 del sistema anfitrión al puerto 22 de la máquina virtual, lo que facilita el acceso a través de SSH
- **La tercera regla** redirige el puerto 8069 del sistema anfitrión al puerto 8069 de la maquina virtual, lo que permite conectarnos a Odoo.



Antes de empezar la instalación tendremos que ejecutar estos dos comandos:

- **sudo apt-get update**: Este comando se usa para actualizar la lista de paquetes disponibles en los repositorios de software. Esto asegura que estemos instalando las versiones más actualizadas.
- **sudo apt-get dist-upgrade**: Este comando se utiliza para realizar una actualización completa del sistema, incluyendo la instalación de nuevas versiones de paquetes. A diferencia de apt-get upgrade, que solo actualiza los paquetes instalados, dist-upgrade es más agresivo y puede instalar o eliminar paquetes si es necesario.

```
fernando@fernando:/etc/netplan$ sudo apt-get update
[sudo] password for fernando:
Hit:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [806 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [170 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [889 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [143 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [532 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [785 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [144 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.7 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 3,100 kB in 34s (91.6 kB/s)
Reading package lists... Done
fernando@fernando:/etc/netplan$ _
```

```
fernando@fernando:/etc/netplan$ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
fernando@fernando:/etc/netplan$ _
```

Ahora vamos a crear el usuario básico para que funcione Odoo. Tendrá como home el directorio /opt/odoo. Como intérprete de comandos usaremos /bin/bash y le pondremos contraseña.

Este es el comando que usaremos: `sudo adduser --system --quiet --shell=/bin/bash --home=/opt/odoo --group odoo`

```
fernando@fernando:/etc/netplan$ sudo adduser --system --quiet --shell=/bin/bash --home=/opt/odoo --group odoo
fernando@fernando:/etc/netplan$ sudo passwd odoo
New password:
Retype new password:
passwd: password updated successfully
fernando@fernando:/etc/netplan$ _
```

Explicación del comando:

1. **adduser:** Este es el comando para agregar un nuevo usuario al sistema.
2. **--system:** Esta opción indica que se está creando un usuario de sistema en lugar de un usuario regular.
3. **--quiet:** Esta opción suprime la mayoría de las salidas del comando, lo que hace que la creación del usuario sea más silenciosa.
4. **--shell=/bin/bash:** Esta opción especifica el shell (intérprete de comandos) que se asignará al usuario.
5. **--home=/opt/odoo:** Esta opción define el directorio de inicio del usuario, que es el directorio en el que el usuario comenzará cuando inicie sesión.
6. **--group odoo:** Esta opción especifica el grupo al que pertenecerá el usuario.

Ahora instalaremos postgresql para ello usaremos el comando: `sudo apt-get install postgresql`

```
fernando@fernando:/etc/netplan$ sudo apt-get install postgresql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
postgresql is already the newest version (14+238).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
fernando@fernando:/etc/netplan$ _
```

Ahora crearemos el usuario odoo. Este nos servirá para administrar y acceder a la base de datos. Para ello usaremos el siguiente comando: `sudo su -postgres -c "createuser -s odoo"`

```
fernando@fernando:/etc/netplan$ sudo su - postgres -c "createuser -s odoo"
fernando@fernando:/etc/netplan$
```

Explicación del comando:

1. **su -postgres**: Este comando cambia al usuario "postgres".
2. **-c**: Esta opción se utiliza con el comando "su" para especificar un comando que se ejecutará en la sesión del usuario "postgres"
3. **"createuser -s odoo"**: Este es el comando que se ejecutará en la sesión del usuario "postgres"

2.1

Instalación

Después instalaremos Odoo de su repositorio y lo almacenamos en la carpeta /opt/odoo del usuario odoo. Para entrar al usuario usamos `su odoo` luego `cd /opt/odoo` para movernos a la carpeta y por último `git clone https://www.github.com/odoo/odoo --depth 1 --branch 16.0 --single-branch` para instalar odoo.

```
fernando@fernando:~$ su odoo
Password:
odoo@fernando:/home/fernando$ cd /opt/odoo
odoo@fernando:/opt/odoo$ sudo git clone https://www.github.com/odoo/odoo --depth 1 --branch 16.0 --single-branch
[sudo] password for odoo:
Cloning into 'odoo'...
warning: redirecting to https://github.com/odoo/odoo.git/
remote: Enumerating objects: 41380, done.
remote: Counting objects: 100% (41380/41380), done.
remote: Compressing objects: 39% (11298/28969)
```

Ahora tendremos que instalar las librerías necesarias para la posterior instalación de Odoo. Para ello usaremos el comando:

```
sudo apt-get install build-essential python3-pillow python3-lxml python3-dev python3-pip python3-setuptools libpq-dev npm nodejs git gdebi libldap2-dev libsasl2-dev libxml2-dev libxslt1-dev libjpeg-dev apache2 -y
```

```
fernando@fernando:/$ sudo apt-get install build-essential python3-pillow python3-lxml python3-dev python3-pip python3-setuptools libpq-dev npm nodejs git gdebi libldap2-dev libsasl2-dev libxml2-dev libxslt1-dev libjpeg-dev apache2 -y
```

Las librerías que se instalarán son:

- **Python3-pip:** Instalador de librerías Python
- **Gdebi-core:** Hace lo mismo que apt solo que es para paquetes remotos (http, ftp)
- **Libxml2-dev:** Archivos de desarrollo para la biblioteca XML de GNOME
- **Libjpeg-dev:** Es una biblioteca C ampliamente usada para leer y escribir archivos de imagen JPEG

- **Libxslt-dev:** XSLT es un lenguaje XML para definir transformaciones de archivos XML.
- **Libldap2-dev:** Librerías de openldap
- **Libsasl2-dev:** Cyrus SASL: archivos de desarrollo para la biblioteca de abstracción de autenticación
- **Build-essential:** Se trata de un paquete que contiene una lista informativa de los paquetes que se consideran esenciales.
- **Python3-pillow:** Es una biblioteca de procesamiento de imágenes potente y conveniente de python.
- **Python3-lxml:** Es una biblioteca de python que permite un fácil manejo de archivos XML y HTML.
- **Python3-dev:** Contiene los archivos de encabezado que necesita para crear extensiones de python.
- **Python3-setuptools:** Le permite instalar un paquete sin copiar ningún archivo a su directorio de intérprete.
- **Libpq-dev:** Contiene un conjunto mínimo de binarios y encabezados 'PostgreSQL' requeridos para crear aplicaciones de terceros para PostgreSQL.
- **Npm:** Administrador de paquetes de nodos.
- **Nodejs:** Es un entorno en tiempo de ejecución multiplataforma.
- **Apache2:** servidor web.

2.1

Instalación

A continuación, instalaremos las dependencias necesarias con pip

Primero actualizaremos pip con el comando: `pip3 install --upgrade pip`.

```
fernando@fernando:/$ pip3 install wheel setuptools pip --upgrade
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: wheel in /usr/lib/python3/dist-packages (0.37.1)
Collecting wheel
  Downloading wheel-0.41.2-py3-none-any.whl (64 kB)
    64.8/64.8 KB 1.1 MB/s eta 0:00:00
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (59.6.0)
Collecting setuptools
  Downloading setuptools-68.2.2-py3-none-any.whl (807 kB)
    807.9/807.9 KB 2.0 MB/s eta 0:00:00
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (22.0.2)
Collecting pip
  Downloading pip-23.2.1-py3-none-any.whl (2.1 MB)
    2.1/2.1 MB 1.9 MB/s eta 0:00:00
Installing collected packages: wheel, setuptools, pip
Successfully installed pip-23.2.1 setuptools-68.2.2 wheel-0.41.2
```

Luego instalaremos los paquetes y dependencias necesarios para ejecutar Odoo con `sudo pip3 install -r /opt/odoo/odoo/requirements.txt`.

```
fernando@fernando:~$ pip freeze_
python-debian==0.1.43+ubuntu1.1
python-ldap==3.4.0
python-magic==0.4.24
python-stodum==1.16
pytz==2022.1
pyusb==1.0.2
PyYAML==5.4.1
qrcode==6.1
reportlab==3.5.59
requests==2.25.1
requests-file==1.5.1
requests-toolbelt==1.0.0
SecretStorage==3.3.1
service-identity==18.1.0
six==1.16.0
sos==4.5.6
soupsieve==2.5
ssh-import-id==5.11
systemd-python==234
Twisted==22.1.0
ubuntu-advantage-tools==8001
ubuntu-drivers-common==0.0.0
ufw==0.36.1
unattended-upgrades==0.1
urllib3==1.26.5
vobject==0.9.6.1
wadllib==1.3.6
Werkzeug==2.0.2
xkit==0.0.0
xlrd==1.2.0
XlsxWriter==1.1.2
xlwt==1.3.0
zeep==4.0.0
zipp==1.0.0
zope.event==5.0
zope.interface==5.4.0
fernando@fernando:~$
```

2.1

Instalación

Para ver las librerías que se nos han instalado usaremos `pip freeze`

```
fernando@fernando:~$ pip freeze_
python-debian==0.1.43+ubuntu1.1
python-ldap==3.4.0
python-magic==0.4.24
python-stdnum==1.16
pytz==2022.1
pyusb==1.0.2
PyYAML==5.4.1
qrcode==6.1
reportlab==3.5.59
requests==2.25.1
requests-file==1.5.1
requests-toolbelt==1.0.0
SecretStorage==3.3.1
service-identity==18.1.0
six==1.16.0
sos==4.5.6
soupsieve==2.5
ssh-import-id==5.11
systemd-python==234
Twisted==22.1.0
ubuntu-advantage-tools==8001
ubuntu-drivers-common==0.0.0
ufw==0.36.1
unattended-upgrades==0.1
urllib3==1.26.5
vobject==0.9.6.1
wadllib==1.3.6
Werkzeug==2.0.2
xkit==0.0.0
xlrd==1.2.0
XlsxWriter==1.1.2
xlwt==1.3.0
zeep==4.0.0
zipp==1.0.0
zope.event==5.0
zope.interface==5.4.0
fernando@fernando:~$
```

Ahora probaremos a ejecutar Odoo para comprobar que la instalación ha sido exitosa.

Para ello iniciamos sesión con el usuario odoo que hemos creado durante la instalación.

Después nos dirigimos hacia `/opt/odoo/odoo/` y una vez dentro ejecutamos el fichero odoo-bin Para ello podemos usar el comando `./odoo-bin`

```
odoo@fernando:~/odoo$ ./odoo-bin
2023-10-06 07:05:17,161 1275 INFO ? odoo: Odoo version 16.0
2023-10-06 07:05:17,162 1275 INFO ? odoo: addons paths: ['/opt/odoo/odoo/odoo/addons', '/opt/odoo/.local/share/odoo/addons/16.0', '/opt/odoo/odoo/addons']
2023-10-06 07:05:17,163 1275 INFO ? odoo: database: default@default:default
2023-10-06 07:05:17,355 1275 INFO ? odoo.addons.base.models.ir_actions_report: You need Wkhtmltopdf to print a pdf version of the reports.
2023-10-06 07:05:17,526 1275 INFO ? odoo.service.server: HTTP service (werkzeug) running on fernando:8069
```

2.3

Error durante el arranque

Durante el arranque de Odoo nos puede aparecer algún error si nos hemos equivocado durante su instalación o ha habido algún fallo al instalar alguna librería.

En nuestro caso nos apareció el siguiente error:

```
bash: ./odoo.bin: No such file or directory
root@fernando:/opt/odoo/odoo# ./odoo-bin
Traceback (most recent call last):
  File "/opt/odoo/odoo/./odoo-bin", line 5, in <module>
    import odoo
  File "/opt/odoo/odoo/odoo/__init__.py", line 75, in <module>
    import PyPDF2
ModuleNotFoundError: No module named 'PyPDF2'
root@fernando:/opt/odoo/odoo# _
```

Esto es debido a que falta alguna librería debido a un fallo durante la instalación. Podemos intentar instalarla manualmente con [pip3 install PyPDF2](#)

```
root@fernando:/opt/odoo/odoo# pip3 install PyPDF2
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
232.6/232.6 KB 1.7 MB/s eta 0:00:00
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
```

Pero en nuestro caso había múltiples librerías que no se habían instalado por lo que usamos el comando [sudo pip3 install wheel setuptools pip --upgrade](#) para unos programas relacionados con la administración de los paquetes y volveremos a intentar instalar las librerías con los comandos de los pasos anteriores.

```
root@fernando:/opt/odoo/odoo# sudo pip3 install wheel setuptools pip --upgrade
Requirement already satisfied: wheel in /usr/lib/python3/dist-packages (0.37.1)
Collecting wheel
  Obtaining dependency information for wheel from https://files.pythonhosted.org/packages/b8/8b/31273bf66016be6ad22bb7345c37ff350276cfd46e389a0c2ac5da9d9073/wheel-0.41.2-py3-none-any.whl.metadata
  Using cached wheel-0.41.2-py3-none-any.whl.metadata (2.2 kB)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (59.6.0)
Collecting setuptools
  Obtaining dependency information for setuptools from https://files.pythonhosted.org/packages/bb/26/7945080113158354380a12ce26873dd6c1ebd88d47f5bc24e2c5bb38c16a/setuptools-68.2.2-py3-none-any.whl.metadata
  Using cached setuptools-68.2.2-py3-none-any.whl.metadata (6.3 kB)
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (23.2.1)
Using cached wheel-0.41.2-py3-none-any.whl (64 kB)
Using cached setuptools-68.2.2-py3-none-any.whl (807 kB)
Installing collected packages: wheel, setuptools
  Attempting uninstall: wheel
    Found existing installation: wheel 0.37.1
    Uninstalling wheel-0.37.1:
      Successfully uninstalled wheel-0.37.1
  Attempting uninstall: setuptools
    Found existing installation: setuptools 59.6.0
    Uninstalling setuptools-59.6.0:
      Successfully uninstalled setuptools-59.6.0
Successfully installed setuptools-68.2.2 wheel-0.41.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@fernando:/opt/odoo/odoo#
```

2.4

Web dependencys

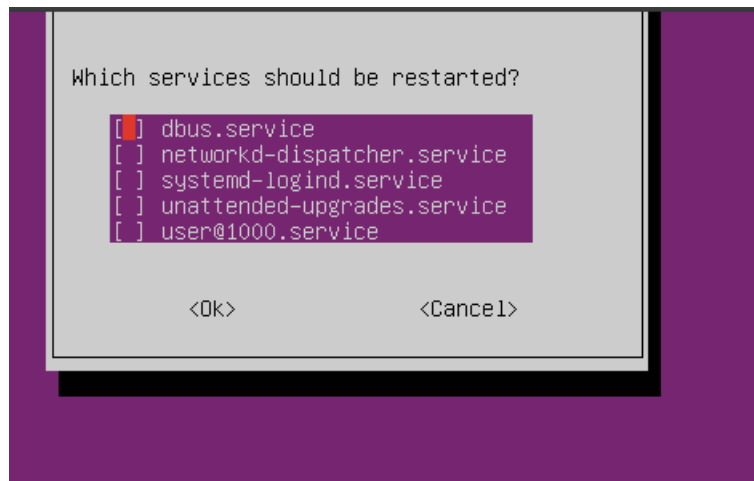
Las siguientes ejecuciones no son necesarias

Se utilizan para instalar las dependencias web requeridas por Odoo, como compiladores de estilos y herramientas de procesamiento web.

Primero instalaremos un administrador de paquetes Node.js para debian. Para ello usaremos el siguiente comando: `sudo apt install -y npm` (La opción -y se utiliza para confirmar automáticamente durante la instalación)

```
fernando@fernando:/opt$ sudo apt install -y npm
```

Después se nos pedirá reiniciar:



2.4

Web dependencys

Ahora necesitamos crear un enlace simbólico (symbolic link) llamado "nodo" en el directorio /usr/bin que apunta al ejecutable de Node.js Para ello usaremos el comando: `sudo ln -s /usr/bin/node.js /usr/bin/nodo`

```
fernando@fernando:/opt$ sudo ln -s /usr/bin/nodejs /usr/bin/nodo
fernando@fernando:/opt$
```

Ahora instalaremos los paquetes "less," "less-plugin-clean-css" y "rtlcss" a través de npm, lo que permite trabajar con estilos CSS y realizar tareas de optimización de código CSS. Para ello usaremos el siguiente comando: `sudo npm install -g less less-plugin-clean-css rtlcss`

```
fernando@fernando:/opt$ sudo npm install -g less less-plugin-clean-css rtlcss
(#####) ♦ reify:copy-anything: sill audit bulk request {
```

A continuacion instalaremos el paquete nodo-less que nos servira para compilar archivos Less (un preprocesador de CSS). Para ello usaremos el comando: `sudo apt install nodo-less`

```
fernando@fernando:/opt$ sudo apt install nodo-less
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

3.1

Ficheros log

Cuando se arranca, para o se produce algún error, la información que se genera se puede guardar en ficheros de log para su consulta.

Para ello tendremos que crear el fichero de configuración. Antes de crear el archivo crearemos la carpeta que llamaremos odoo dentro de la ruta /var/log/odoo Para crear la carpeta usaremos el comando: `mkdir /var/log/odoo/`

Después de haber creado la carpeta, la modificaremos para que el propietario sea el usuario odoo, pero el grupo propietario continúe siendo el del usuario original. Así nos aseguramos que se puedan guardar los archivos generados desde odoo. Para cambiar los permisos usaremos el comando: `chown odoo:root /var/log/odoo`

```
fernando@fernando:~$ sudo mkdir /var/log/odoo/  
fernando@fernando:~$ sudo chown odoo:root /var/log/odoo  
fernando@fernando:~$
```

3.2

Fichero de configuración

El fichero de configuración es un fichero en el que se guarda toda la configuración necesaria para que al arrancar odoo este pueda leer los datos y que funcione.

Primero copiaremos el fichero de configuración /opt/odoo/odoo/debian/odoo.conf donde se encuentra los ficheros .conf en la carpeta /etc/ Para ello usaremos el comando: `sudo cp /opt/odoo/odoo/debian/odoo.conf /etc/odoo.conf`

Luego modificaremos el propietario de la carpeta de manera que el usuario propietario sea odoo, pero el grupo propietario sea el mismo. Usaremos el comando: `sudo chown odoo: /etc/odoo.conf`

Luego modificaremos los permisos del fichero de manera que el propietario tenga todos los permisos y el grupo solo lectura. Para esto emplearemos el comando: `sudo chmod 640 /etc/odoo.conf`

Por último editamos el fichero de configuración y modificamos los datos para que el usuario de la base de datos sea odoo y asi podemos trabajar con la base de datos sin problemas. Para ello editaremos el fichero odoo.conf con `nano` de la siguiente forma.

```
fernando@fernando:~$ sudo cp /opt/odoo/odoo/debian/odoo.conf /etc/odoo.conf
fernando@fernando:~$ sudo chown odoo: /etc/odoo.conf
fernando@fernando:~$ sudo chmod 640 /etc/odoo.conf
fernando@fernando:~$ _
```

```
GNU nano 6.2 /etc/odoo.conf *
[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = False
db_port = False
db_user = odoo
db_password = False
; addons_path = /opt/odoo/odoo/addons
logfile = /var/log/odoo/odoo-server.log
default_productivity_apps = True
```


3.3

Arranque automático de Odoo

Para no tener que arrancar manualmente odoo y poder manejarlo como un servicio, necesitamos crear dicho servicio y configurarlo.

Para ello primero tenemos que copiar el fichero odoo.service de la ruta de instalación a la carpeta de servicios del sistema con el comando: `sudo cp /opt/odoo/odoo/debian/odoo.service /etc/systemd/system/odoo.service`

```
fernando@fernando:~$ sudo cp /opt/odoo/odoo/debian/odoo.service /etc/systemd/system/odoo.service
fernando@fernando:~$ sudo _
```

Después modificaremos el fichero de manera que el usuario y el grupo sea odoo. También le indicaremos la ruta de arranque de odoo y la ruta del fichero de configuración, para editar el fichero usaremos `nano`

```
GNU nano 6.2 /etc/systemd/system/odoo.service *
[Unit]
Description=Odoo Open Source ERP and CRM
After=network.target

[Service]
Type=simple
User=odoo
Group=odoo
ExecStart=/opt/odoo/odoo/odoo-bin --config /etc/odoo.conf_
KillMode=mixed

[Install]
WantedBy=multi-user.target
```

3.3

Arranque automático de Odoo

Ahora podemos activar el servicio usando el comando: `sudo systemctl enable odoo.service`

```
fernando@fernando:~$ sudo systemctl enable odoo.service
Created symlink /etc/systemd/system/multi-user.target.wants/odoo.service → /etc/systemd/system/odoo.service.
fernando@fernando:~$
```

Y ya tendríamos activado el servicio. Ahora podemos usar los siguientes comandos para gestionarlo:

- Arrancar el servicio: `sudo systemctl start odoo`
- Parar el servicio: `sudo systemctl stop odoo`
- Ver estado del servicio: `sudo systemctl status odoo`

4.1

Entrada al sistema

Ahora probaremos su funcionamiento accediendo mediante nuestro navegador a través del puerto 8069

