

Manual de Instalación de Odoo 17 en Ubuntu server

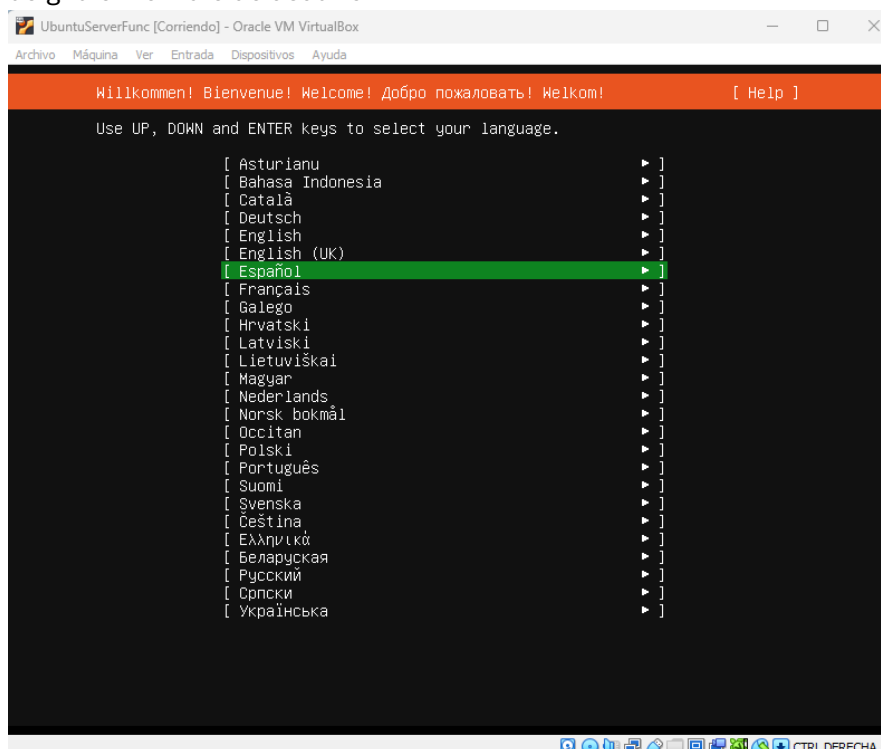
Nombre: Leonardo Ramos Barrenozo

Introducción:

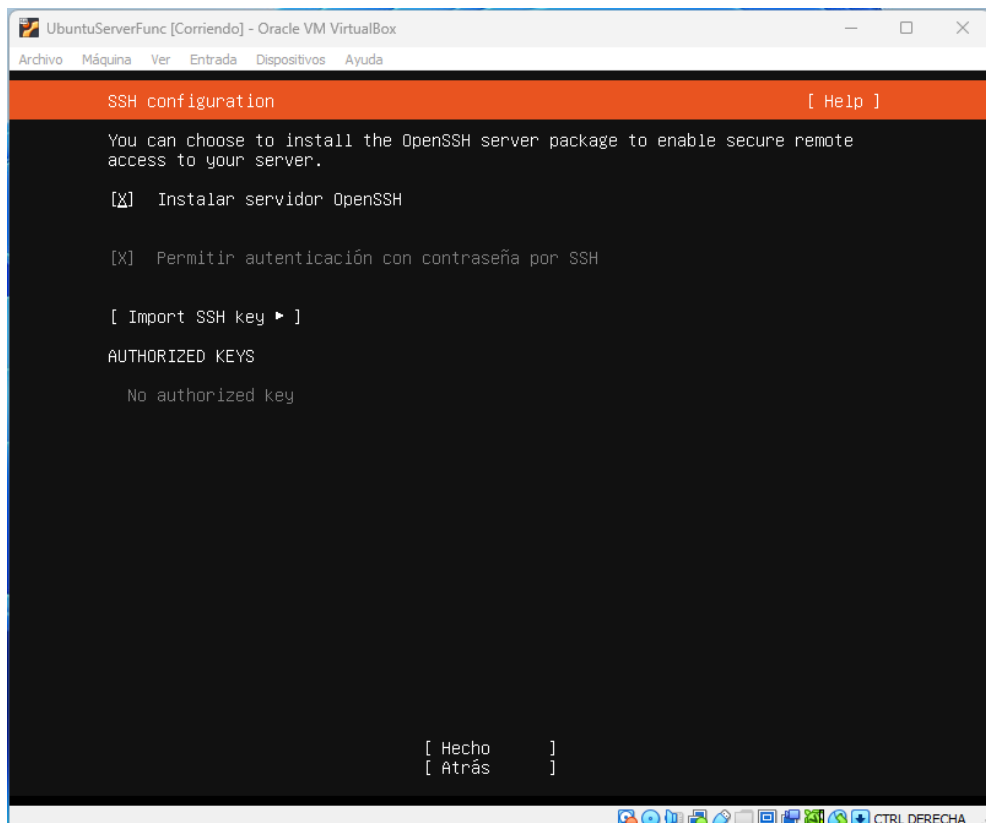
El primer paso consiste en crear la máquina virtual sin incluir la ISO en esta etapa. Durante la configuración inicial, asignamos un nombre, definimos el espacio de almacenamiento necesario y completamos los parámetros básicos. Es importante destacar que la máquina no se iniciará en este punto.

A continuación, configuramos la red de la máquina virtual, seleccionando la opción "Adaptador puente" (Bridge Adapter), ya que es esencial para que Odoo funcione correctamente. Finalmente, agregamos la ISO correspondiente y procedemos a arrancar la máquina virtual.

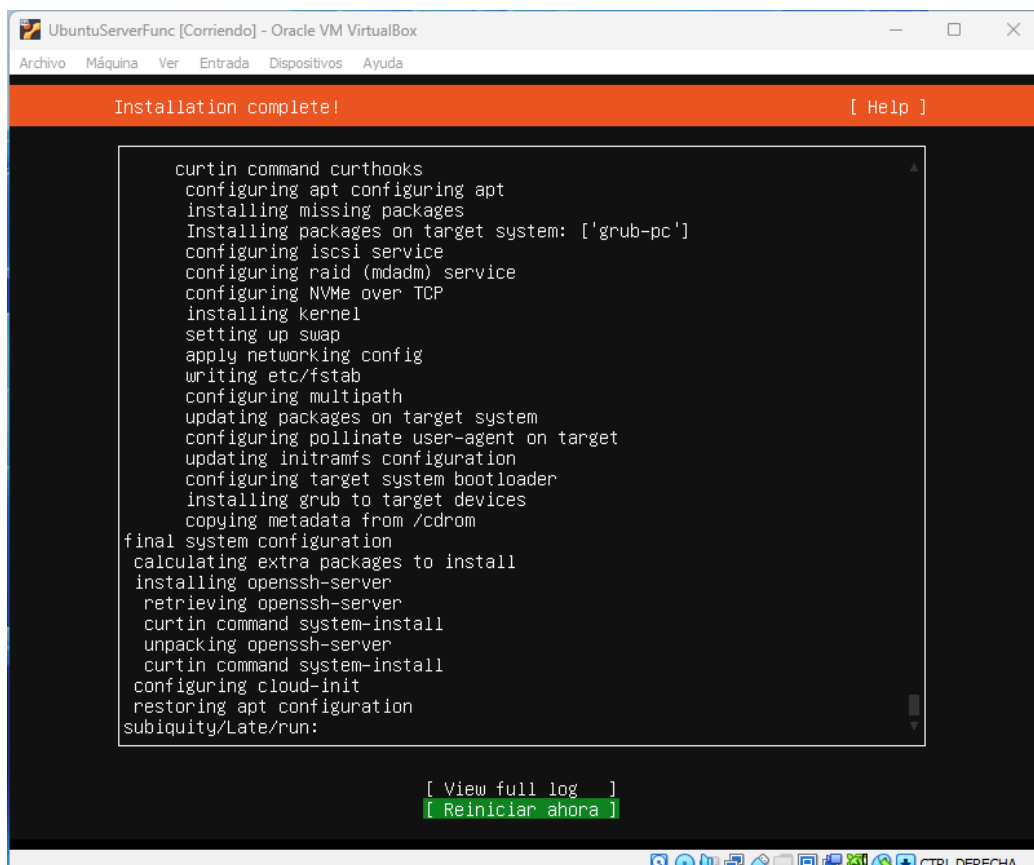
Al iniciar el proceso de instalación, lo primero que debemos seleccionar es el idioma. En nuestro caso, elegimos español. Todas las configuraciones posteriores se aceptan con los valores predeterminados, sin realizar modificaciones, hasta llegar al paso donde se asigna el nombre de usuario.



Es recomendable elegir un nombre de usuario adecuado y una contraseña fácil de recordar. Una vez configurados estos datos, continuamos con la instalación hasta llegar al apartado relacionado con la activación de OpenSSH.

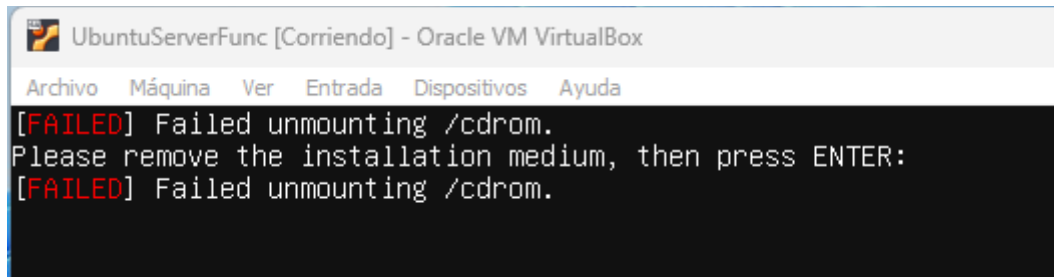


Seleccionamos la opción correspondiente presionando "Enter" y procedemos aceptando todas las configuraciones predeterminadas. Esto nos llevará al proceso de instalación, el cual puede tardar unos minutos en completarse.

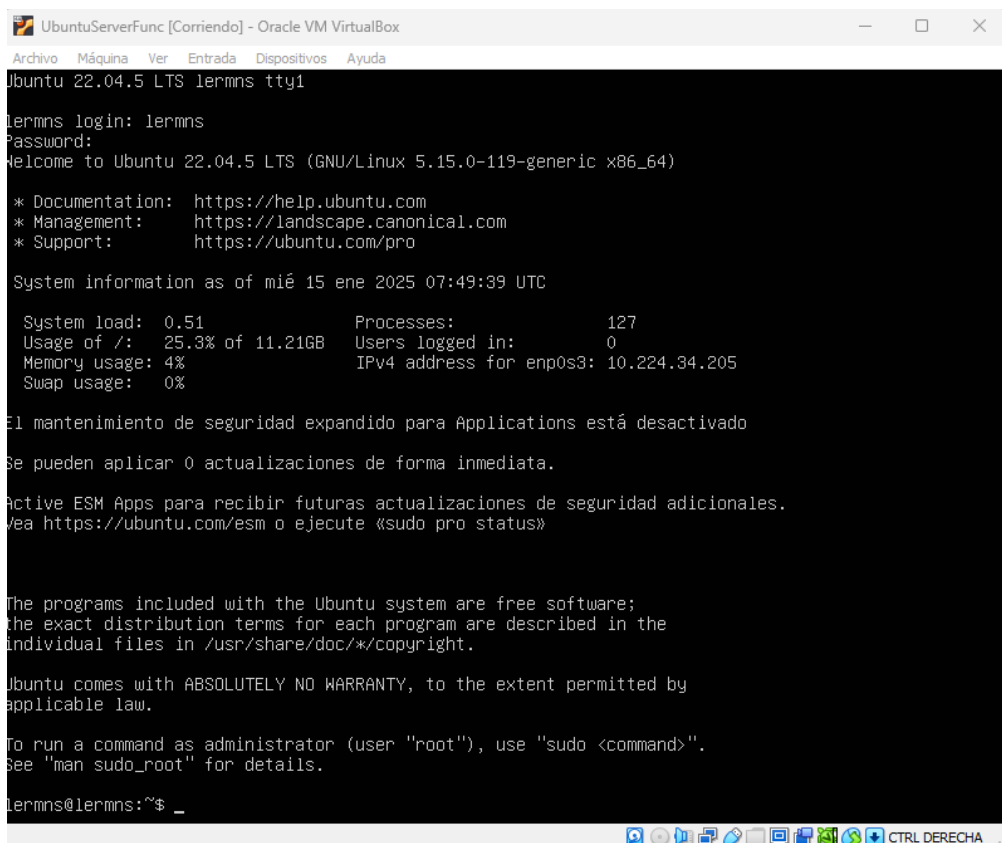


Una vez que el proceso de instalación finalice, se nos presentará la opción de reiniciar el sistema. Procedemos a seleccionarla para completar esta etapa.

Reiniciamos la máquina virtual y, al iniciarla nuevamente, aparecerá un mensaje en pantalla. Seleccionamos la opción "Start" y, a continuación, el sistema nos solicitará ingresar el login.



Iniciamos sesión utilizando el nombre de usuario y la contraseña configurados durante la instalación. Este será el usuario principal del sistema.



Una vez dentro, comenzamos actualizando los paquetes disponibles utilizando los siguientes comandos:

```
sudo apt update
```

```
sudo apt upgrade
```

Luego, procedemos a instalar los paquetes necesarios para PostgreSQL, que es requerido por Odoo 17. Utilizamos el siguiente comando:

```
sudo apt install postgresql postgresql-server-dev-14
```

A continuación, instalamos Python y los paquetes esenciales para el funcionamiento de Odoo 17. Usamos el siguiente comando, asegurándonos de escribirlo en una sola línea:

```
sudo apt install build-essential python3-pillow python3-lxml python3-dev python3-pip python3-setuptools npm nodejs git gdebi libldap2-dev libsasl2-dev libxml2-dev python3-wheel python3-venv libxslt1-dev node-less libjpeg-dev
```

Para verificar que PostgreSQL se ha instalado correctamente, utilizamos el siguiente comando:

```
psql --version
```

Al ejecutarlo, deberíamos ver un resultado similar al mostrado en la imagen, que confirma la versión instalada de PostgreSQL.

```
lermns@lermns:~$ psql --version
psql (PostgreSQL) 14.15 (Ubuntu 14.15-0ubuntu0.22.04.1)
lermns@lermns:~$
```

A continuación, iniciamos el servidor de PostgreSQL utilizando el siguiente comando:

```
sudo pg_ctlcluster 14 main start
```

Para verificar el estado del servidor de PostgreSQL, usamos el siguiente comando:

```
sudo systemctl status postgresql
```

Al ejecutarlo, deberíamos obtener un resultado similar al mostrado en la imagen, indicando que el servidor está en funcionamiento.

```
lermns@lermns:~$ sudo systemctl status postgresql
[sudo] password for lermns:
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2025-01-15 13:34:20 UTC; 1min 53s ago
   Process: 832 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 832 (code=exited, status=0/SUCCESS)
   CPU: 3ms

ene 15 13:34:20 lermns systemd[1]: Starting PostgreSQL RDBMS...
ene 15 13:34:20 lermns systemd[1]: Finished PostgreSQL RDBMS.
lermns@lermns:~$ _
```

Antes de continuar, es importante crear un usuario llamado "odoo17", el cual utilizaremos para levantar el servicio de Odoo. Para ello, ejecutamos el siguiente comando:

```
sudo useradd -m -d /opt/odoo17 -U -r -s /bin/bash odoo17
```

A continuación, asignamos una contraseña al usuario con el comando:

```
sudo passwd odoo17
```

Finalmente, agregamos el usuario "odoo17" al grupo de sudo para otorgarle privilegios administrativos, utilizando el siguiente comando:

```
sudo adduser odoo17 sudo
```

Una vez establecida la contraseña, deberíamos ver una imagen similar a la mostrada.

```
lermns@lermns:~$ sudo passwd odoo17
New password:
Retype new password:
passwd: password updated successfully
lermns@lermns:~$
```

Para verificar que el usuario se ha creado correctamente, utilizamos el siguiente comando:

```
tail -n 1 /etc/passwd
```

Este comando nos muestra el último usuario registrado en el sistema. Al ejecutarlo, podemos comprobar que el usuario "odoo17" ha sido creado correctamente.

```
lermns@lermns:~$ tail -n 1 /etc/passwd
odoo17:x:998:998:./opt/odoo17:/bin/bash
```

Para verificar que el usuario "odoo17" ha sido agregado correctamente al grupo de sudo, utilizamos el siguiente comando:

```
groups odoo17
```

Este comando nos mostrará los grupos a los que pertenece el usuario "odoo17", incluyendo el grupo "sudo", como se muestra en la imagen.

```
lermns@lermns:~$ groups odoo17
odoo17 : odoo17 sudo
```

A continuación, es importante crear un usuario dentro de PostgreSQL para poder levantar el servidor de Odoo 17. Para ello, utilizamos el siguiente comando (que debe ejecutarse como una sola línea):

```
sudo su - postgres -c "createuser -s odoo17"
```

Este comando crea el usuario "odoo17" dentro de PostgreSQL con privilegios adecuados para gestionar Odoo 17.

Una vez completado este paso, necesitamos descargar un paquete especial que nos permitirá generar informes desde Odoo. Para ello, utilizamos el siguiente comando:

```
sudo curl -O https://github.com/wkhtmltopdf/packaging/releases/tag/0.12.6.1-3/wkhtmltox_0.12.6.1-3.jammy_amd64.deb
```

Este comando descargará el paquete necesario para generar informes en formato PDF.

Luego, podemos verificar que el archivo se ha descargado correctamente con el siguiente comando:

```
ls
```

Este comando nos mostrará el listado de archivos en el directorio actual, y deberíamos ver el archivo descargado, tal como se muestra en la imagen.

```
lermns@lermns:~$ ls
wkhtmltox_0.12.6.1-3.jammy_amd64.deb
```

Una vez confirmado que el archivo está en la ubicación correcta, procedemos a instalar el paquete utilizando el siguiente comando:

```
sudo apt install ./wkhtmltox_0.12.6.1-3.jammy_amd64.deb
```

Este comando instalará el paquete descargado, sin generar problemas, siempre y cuando estemos ubicados en el directorio donde se descargó el archivo.

Ahora debemos crear un ambiente virtual para mantener nuestro servidor aislado de otros servicios o aplicaciones de Ubuntu. Para hacerlo, cambiamos al usuario `odoo17` con el siguiente comando (todo en una sola línea):

```
sudo su - odoo17
```

Es probable que te pida la contraseña; la contraseña que debes ingresar es la misma que configuraste previamente. Al ejecutar este comando, verás que el usuario cambia a `odoo17`, tal como se muestra en la imagen.

```
lermns@lermns:~$ sudo su - odoo17
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

odoo17@lermns:~$ _
```

Este paso es importante, ya que, dependiendo de la velocidad de tu conexión a Internet, el siguiente comando podría generar errores. Por esta razón, te recomiendo hacerlo en un lugar con buena velocidad de conexión, ya que, de lo contrario, podrías encontrarte con problemas. Utilizaremos el siguiente comando para clonar el repositorio de GitHub, que contiene el código fuente de Odoo 17. En mi caso, cuando lo hice en casa, funcionó perfectamente; aunque tarda un par de minutos, no lanza errores:

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 17.0 /opt/odoo17/odoo
```

Una vez ejecutado, deberíamos ver algo similar a lo mostrado en la imagen.

```
odoo17@lermns:~$ git clone https://www.github.com/odoo/odoo --depth 1 --branch 17.0 /opt/odoo17/odoo
Cloning into '/opt/odoo17/odoo'...
warning: redirecting to https://github.com/odoo/odoo.git/
remote: Enumerating objects: 41562, done.
remote: Counting objects: 100% (41562/41562), done.
remote: Compressing objects: 100% (31464/31464), done.
remote: Total 41562 (delta 10181), reused 33966 (delta 8641), pack-reused 0 (from 0)
Receiving objects: 100% (41562/41562), 164.56 MiB | 6.34 MiB/s, done.
Resolving deltas: 100% (10181/10181), done.
Updating files: 100% (35848/35848), done.
```

Ahora necesitamos crear el ambiente virtual donde instalaremos las dependencias de Odoo. Primero, nos ubicamos en el directorio adecuado (seguimos usando el usuario `odoo17`):

```
cd /opt/odoo17
```

A continuación, creamos el entorno virtual de Python con el siguiente comando:

```
python3 -m venv odoo-venv
```

Para activar el entorno virtual, usamos el siguiente comando desde la misma ruta en la que nos encontramos:

```
source odoo-venv/bin/activate
```

Al hacerlo, veremos cómo el usuario cambia al entorno virtual, tal como se muestra en la imagen.

```
odoo17@lermns:~$ python3 -m venv odoo-venv
odoo17@lermns:~$ source odoo-venv/bin/activate
(odoo-venv) odoo17@lermns:~$
```

Ahora, una vez dentro del entorno virtual, vamos a instalar las dependencias necesarias. Para hacerlo, usamos el siguiente comando:

pip3 install wheel

Veremos cómo las dependencias se instalan correctamente, tal como se muestra en la imagen.

```
(odoo-venv) odoo17@lermns:~$ pip3 install wheel
Collecting wheel
  Downloading wheel-0.45.1-py3-none-any.whl (72 kB)
    _____ 72.5/72.5 KB 1.0 MB/s eta 0:00:00
Installing collected packages: wheel
Successfully installed wheel-0.45.1
(odoo-venv) odoo17@lermns:~$
```

Continuamos con el siguiente comando para instalar las dependencias:

pip3 install -r odoo/requirements.txt

Es posible que veas un error en la consola similar al de la imagen, donde se indica un problema al compilar "gevent". Esto se debe a que necesitaremos eliminar algunas líneas problemáticas para poder continuar con la instalación.

```
File "/tmp/pip-build-env-c7opzk01/overlay/lib/python3.10/site-packages/Cython/Build/Dependencies.py", line 1256, in cythonize_one
    raise CompileError(None, pyx_file)
Cython.Compiler.Errors.CompileError: src/gevent/libev/corecext.pyx
[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.
error: subprocess-exited-with-error

× Getting requirements to build wheel did not run successfully.
  exit code: 1
  → See above for output.

note: This error originates from a subprocess, and is likely not a problem with pip.
```

El problema radica en las dependencias del archivo requirements.txt, por lo que necesitamos editar algunas líneas. Para ello, utilizamos el siguiente comando:

sudo nano odoo/requirements.txt

Una vez dentro del editor, debes eliminar las siguientes líneas completas (si alguna de estas líneas tiene otras dependencias, también deben ser eliminadas):

gevent==21.8.0 ; python_version == '3.10' # (Jammy)

greenlet==1.1.2 ; python_version == '3.10' # (Jammy)

pywin32 ; sys_platform == 'win32'

Después de borrar estas líneas, guarda los cambios y sal del editor.

```
freezegun==1.1.0 ; python_version < '3.11' # (Jammy)
freezegun==1.2.1 ; python_version >= '3.11'
geopandas==2.9.0
gevent==21.8.0 ; sys_platform != 'win32' and python_version == '3.10' # (Jammy)
gevent==22.10.2 ; sys_platform != 'win32' and python_version > '3.10' and python_version < '3.12'
gevent==24.2.1 ; sys_platform != 'win32' and python_version >= '3.12' # (Noble)
greenlet==1.1.2 ; sys_platform != 'win32' and python_version == '3.10' # (Jammy)
greenlet==2.0.2 ; sys_platform != 'win32' and python_version > '3.10' and python_version < '3.12'
greenlet==3.0.3 ; sys_platform != 'win32' and python_version >= '3.12' # (Noble)
```

```
PyPDF2==1.26.0 ; python_version <= '3.10'
PyPDF2==2.12.1 ; python_version > '3.10'
pypiwin32 ; sys_platform == 'win32'
pyserial==3.5
python-dateutil==2.8.1 ; python_version < '3
python-dateutil==2.8.2 ; python_version >= '
python-ldap==3.4.0 ; sys_platform != 'win32'
```

Ahora, nos ubicamos en el directorio adecuado con el siguiente comando:

```
cd /opt/odoo17
```

Una vez allí, volvemos a ejecutar el comando para instalar las dependencias desde el archivo requirements.txt:

```
pip3 install -r odoo/requirements.txt
```

Con esto, ya no deberían aparecer problemas y deberías ver un mensaje similar al que aparece en la imagen.

```
Successfully built ebaysdk ofxparse psycogp2 PyPDF2 python-ldap qrcode rjsmin vobject docopt
Installing collected packages: xlwt, rjsmin, pytz, python-stdnum, pyserial, PyPDF2, polib, passlib,
docopt, XlsxWriter, xlrd, Werkzeug, urllib3, soupsieve, six, qrcode, pyusb, pyparsing, pycparser, py
asn1, psycogp2, psutil, platformdirs, Pillow, num2words, maxminddb, MarkupSafe, lxml, isodate, idna,
docutils, decorator, chardet, certifi, cached-property, Babel, attrs, requests, reportlab, python-d
ateutil, pydot, pyasn1_modules, libsass, Jinja2, cffi, beautifulsoup4, vobject, requests-toolbelt, r
equests-file, python-ldap, ofxparse, geoip2, freezegun, ebaysdk, cryptography, zeep, pyopenssl
Successfully installed Babel-2.9.1 Jinja2-3.0.3 MarkupSafe-2.0.1 Pillow-9.0.1 PyPDF2-1.26.0 Werkzeug
-2.0.2 XlsxWriter-3.0.2 attrs-24.3.0 beautifulsoup4-4.12.3 cached-property-2.0.1 certifi-2024.12.14
cffi-1.17.1 chardet-4.0.0 cryptography-3.4.8 decorator-4.4.2 docopt-0.6.2 docutils-0.17 ebaysdk-2.1.
5 freezegun-1.1.0 geoip2-2.9.0 idna-2.10 isodate-0.7.2 libsass-0.20.1 lxml-4.8.0 maxminddb-2.6.3 num
2words-0.5.10 ofxparse-0.21 passlib-1.7.4 platformdirs-4.3.6 polib-1.1.1 psutil-5.9.0 psycogp2-2.9.2
pyasn1-0.6.1 pyasn1_modules-0.4.1 pycparser-2.22 pydot-1.4.2 pyopenssl-21.0.0 pyparsing-3.2.1 pyser
ial-3.5 python-dateutil-2.8.1 python-ldap-3.4.0 python-stdnum-1.17 pytz-2024.2 pyusb-1.2.1 qrcode-7.
3.1 reportlab-3.6.8 requests-2.25.1 requests-file-2.1.0 requests-toolbelt-1.0.0 rjsmin-1.1.0 six-1.1
7.0 soupsieve-2.6 urllib3-1.26.5 vobject-0.9.6.1 xlrd-1.2.0 xlwt-1.3.0 zeep-4.1.0
(odoo-venv) odoo17@1ermns:~$ _
```

Para probar el servidor en la máquina anfitrión, primero debemos conocer la IP de nuestro servidor Ubuntu. Para ello, usamos el siguiente comando:

```
ip addr show
```

La IP que utilizaremos para acceder a nuestro servicio Odoo desde la máquina anfitriona es la que aparece en la imagen. Ten en cuenta que esta IP puede variar dependiendo del equipo.

```
(odoo-venv) odoo17@1ermns:~$ ip addr show
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:da:f9:07 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.29/24 metric 100 brd 192.168.0.255 scope global dynamic enp0s3
        valid_lft 84686sec preferred_lft 84686sec
    inet6 fd19:f656:5780:ca4a:a00:27ff:feda:f907/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 1724sec preferred_lft 1724sec
    inet6 fe80::a00:27ff:feda:f907/64 scope link
        valid_lft forever preferred_lft forever
```


Ahora que ya conocemos la IP, nos situamos en el directorio del código fuente de Odoo con el siguiente comando:

```
cd /opt/odoo17/odoo
```

Luego, ejecutamos el siguiente comando:

```
./odoo-bin
```

No deberíamos encontrar problemas y, al ejecutar este comando, deberíamos ver algo similar a la imagen.

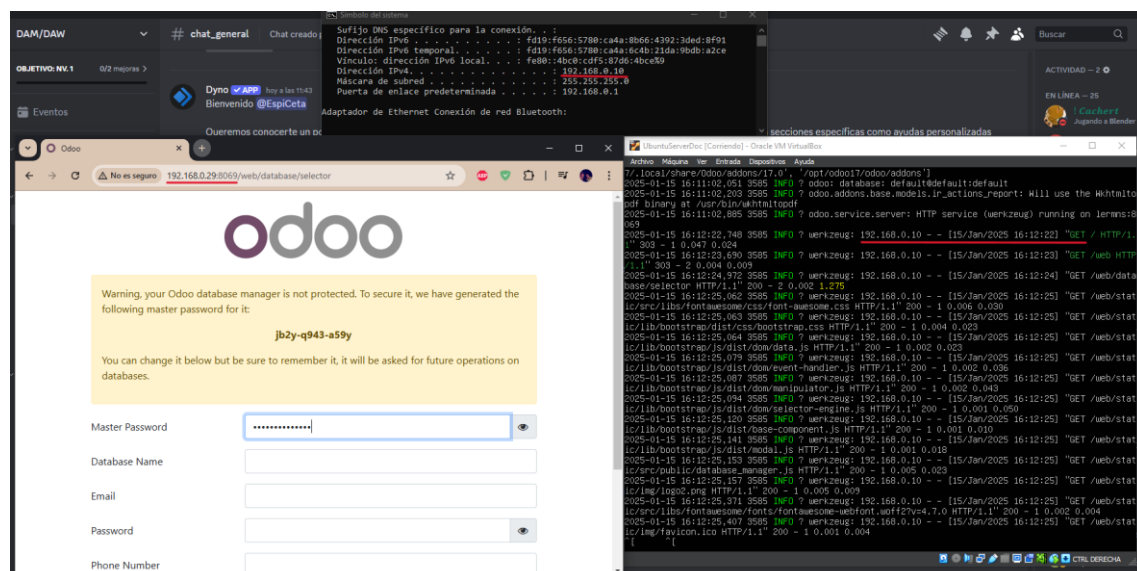
```
(odoo-venv) odoo17@lermns:~$ cd /opt/odoo17/odoo
(odoo-venv) odoo17@lermns:~/odoo$ ./odoo-bin
2025-01-15 16:11:02,051 3585 INFO ? odoo: Odoo version 17.0
2025-01-15 16:11:02,051 3585 INFO ? odoo: addons paths: ['/opt/odoo17/odoo/addons', '/opt/odoo17.0.local/share/Odoo/addons/17.0', '/opt/odoo17/odoo/addons']
2025-01-15 16:11:02,051 3585 INFO ? odoo: database: default:default
2025-01-15 16:11:02,203 3585 INFO ? odoo.addons.base.models.ir_actions_report: Will use the Wkhtmlto
pdf binary at /usr/bin/wkhtmltopdf
2025-01-15 16:11:02,885 3585 INFO ? odoo.service.server: HTTP service (werkzeug) running on lermns:8069
_
```

Ahora probamos Odoo en la máquina anfitriona utilizando la IP de Ubuntu Server como URL, y el puerto 8069.

La URL será algo como:

```
http://<IP_ubuntu_server>:8069
```

Cuando nos conectemos, veremos en Ubuntu Server un aviso de que ha recibido una solicitud "GET" desde una IP, que en este caso es la nuestra. Esto podemos corroborarlo utilizando el comando `ipconfig` en la máquina anfitriona, donde veremos que las IPs coinciden.



Para salir y desactivar el ejecutable del servidor, simplemente presionamos Ctrl+C.

Ahora vamos a ejecutar módulos adicionales para Odoo. Primero, debemos salir del entorno virtual con el siguiente comando:

deactivate

Al ejecutar este comando, regresamos al usuario odoo17, como se muestra en la imagen.

```
(odoo-venv) odoo17@lermns:~/odoo$ deactivate
odoo17@lermns:~/odoo$ _
```

Con el usuario odoo17, creamos el siguiente directorio:

mkdir /opt/odoo17/odoo-custom-addons

Si lo deseas, puedes verificar si se ha creado correctamente como se muestra en la imagen. Este directorio será referenciado más adelante.

```
odoo17@lermns:~/odoo$ mkdir /opt/odoo17/odoo-custom-addons
odoo17@lermns:~/odoo$ cd /opt/odoo17/
odoo17@lermns:~$ ls
odoo  odoo-custom-addons  odoo-venv
```

Ahora debemos volver al usuario principal, es decir, salir de odoo17 usando el siguiente comando:

exit

Veremos cómo regresamos al usuario principal, que en mi caso es lermns.

```
odoo17@lermns:~$ exit
logout
lermns@lermns:~$ _
```

Ahora debemos crear un archivo y agregarle el código necesario. Para ello, usamos el siguiente comando:

sudo nano /etc/odoo17.conf

Esto abrirá el editor, donde debemos copiar el siguiente código (como se muestra en la imagen).

```
[options]
admin_passwd = admin_passwd
db_host = False
db_port = False
db_user = odoo17
db_password = False
addons_path = /opt/odoo17/odoo/addons/opt/odoo17/odoo-custom-addons
xmlrpc_port = 8069
longpolling_port = 8072
workers = 2
; dbfilter = [ur database name]
; los ';' son usados para comentar las lineas.
```

Ahora hacemos lo mismo, pero con otro archivo. Usamos el siguiente comando:

```
sudo nano /etc/systemd/system/odoo17.service
```

Y volvemos a copiar el siguiente código en el archivo.

Una vez hecho esto, usamos el siguiente comando para avisar a Ubuntu sobre los cambios establecidos:

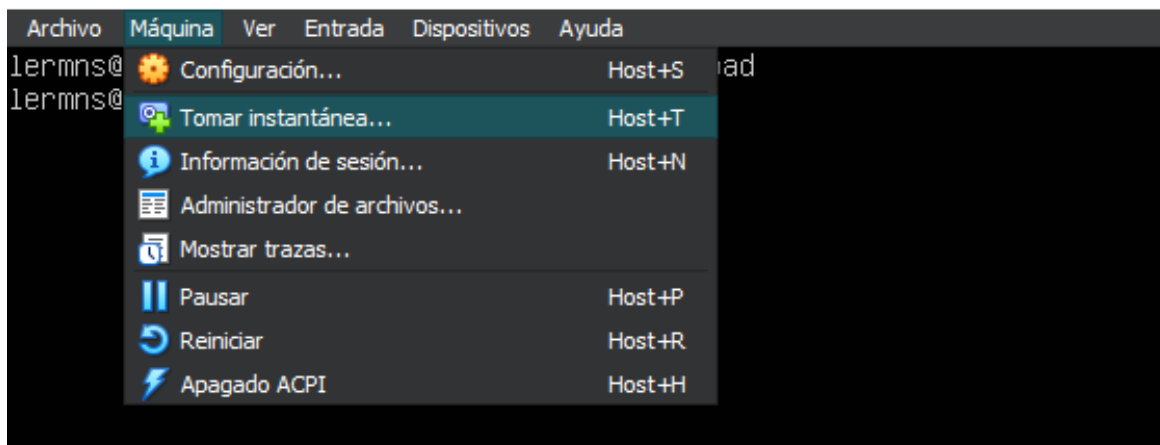
```
sudo systemctl daemon-reload
```

```
[Unit]
Description=Odoo 17
Requires=postgresql.service
After=network.target postgresql.service

[Service]
Type=simple
SyslogIdentifier=odoo17
PermissionsStartOnly=true
User=odoo17
ExecStart=/opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
StandardOutput=journal+console

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

Este punto es importante porque es propenso a errores de dependencias que pueden entrar en un bucle y dañar la máquina virtual. Por eso, recomiendo crear una instantánea por seguridad antes de proceder.

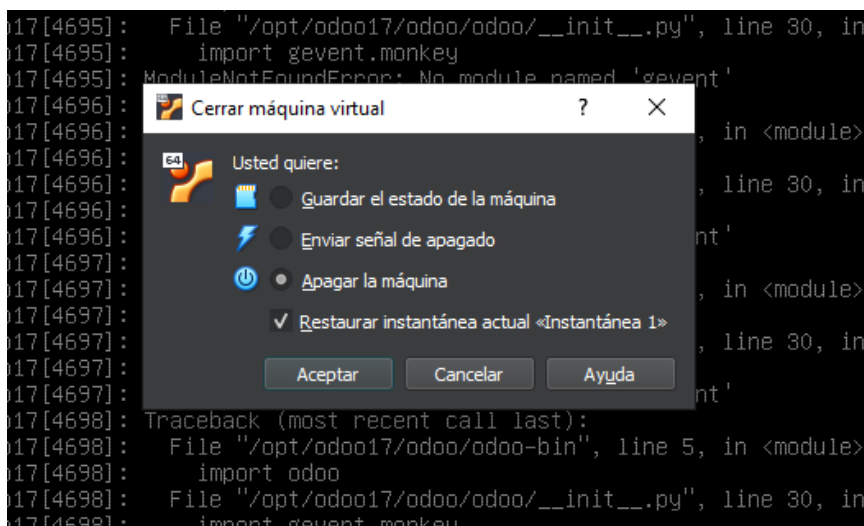


Usamos el comando:

```
sudo systemctl enable --now odoo17
```

para levantar el servicio. En mi caso, vemos cómo se presenta un error de un módulo no encontrado, que es "gevent". Recordemos que eliminamos este módulo del fichero requirements.txt debido a los problemas que causaba. Ahora, vamos a instalarlo manualmente.

Apagamos la máquina virtual y nos da la opción de restaurar la instantánea. Procedemos a restaurarla y luego volvemos a arrancar la máquina, la cual nos llevará directamente al mismo punto donde teníamos la instantánea.



Si recuerdan, la parte donde instalamos los *requirements* era cuando estábamos dentro del entorno virtual, por lo cual volvemos a entrar en él con los siguientes comandos (estos también los tienes arriba para copiarlos).

```

lermns@lermns:~$ sudo su - odoo17
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

odoo17@lermns:~$ cd /opt/odoo17
odoo17@lermns:~$ ls
odoo  odoo-custom-addons  odoo-venv
odoo17@lermns:~$ source odoo-venv/bin/activate
(odoo-venv) odoo17@lermns:~$ _

```

Una vez dentro del entorno virtual, instalamos "gevent" con el siguiente comando:

pip3 install gevent

Esto arreglaría el problema y deberíamos tener algo similar a la imagen.

```

(odoo-venv) odoo17@lermns:~$ pip3 install gevent
Collecting gevent
  Downloading gevent-24.11.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
    6.6/6.6 MB 5.9 MB/s eta 0:00:00
Collecting zope.interface
  Downloading zope.interface-7.2-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (254 kB)
    254.5/254.5 KB 6.1 MB/s eta 0:00:00
Collecting greenlet==3.1.1
  Downloading greenlet-3.1.1-cp310-cp310-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl (599 kB)
    599.5/599.5 KB 6.9 MB/s eta 0:00:00
Collecting zope.event
  Downloading zope.event-5.0-py3-none-any.whl (6.8 kB)
Requirement already satisfied: setuptools in ./odoo-venv/lib/python3.10/site-packages (from zope.event->gevent) (59.6.0)
Installing collected packages: zope.interface, zope.event, greenlet, gevent
Successfully installed gevent-24.11.1 greenlet-3.1.1 zope.event-5.0 zope.interface-7.2

```

Para verificar que "gevent" se ha instalado correctamente, usamos el siguiente comando en la imagen este código en Python se ejecuta e importa "gevent" e imprime la versión. Si el import no logra encontrar la librería, lanzará un error. Pero si la instalación fue exitosa, debería devolver la versión de "gevent" instalada, como se muestra en la imagen.

```

(odoo-venv) odoo17@lermns:~$ python3 -c "import gevent; print(gevent.__version__)"
24.11.1

```

Ahora, con el problema resuelto, salimos del entorno virtual y del usuario "odoo17", como lo hicimos anteriormente. Luego, volvemos a usar el usuario principal (en mi caso, "lermns").

```

(odoo-venv) odoo17@lermns:~$ deactivate
odoo17@lermns:~$ exit
logout
lermns@lermns:~$

```

Volvemos a usar el comando:

sudo systemctl daemon-reload

para avisar a Ubuntu de los cambios. En este punto, vuelvo a crear una instantánea por si surge otro error.

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
lermns@lermns:~$ sudo systemctl daemon-reload
lermns@lermns:~$ sudo su - odoo17
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

odoo17@lermns:~$ cd /opt/odoo17
odoo17@lermns:~$ ls
odoo odoo-custom-ad
odoo17@lermns:~$ sou
(odoo-venv) odoo17@l
Collecting gevent
Downloading gevent
Collecting zope.inte
Downloading zope.i
86_64.manylinux2014_
Collecting greenlet>
Downloading greenl
Collecting zope.even
Downloading zope.e
Requirement already
nt->gevent) (59.6.0)
Installing collected
Successfully install
(odoo-venv) odoo17@l
24.11.1
(odoo-venv) odoo17@lermns: ~$ deactivate
odoo17@lermns:~$ exit
logout
lermns@lermns:~$ sudo systemctl daemon-reload
lermns@lermns:~$ _
```

Tomar instantánea de la máquina virtual

Nombre de instantánea

Instantánea 2

Descripción de instantánea

Aceptar

Cancelar

Ayuda

Usamos nuevamente el comando:

sudo systemctl enable --now odoo17

Y en mi caso, el error fue resuelto y no nos muestra errores.

Luego usamos el comando:

sudo systemctl status odoo17

para ver el estado del servidor, y vemos cómo este está activo y corriendo.

```
lermns@lermns:~$ sudo systemctl status odoo17
• odoo17.service - Odoo 17
  Loaded: loaded (/etc/systemd/system/odoo17.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2025-01-15 16:37:09 UTC; 2min 34s ago
  Main PID: 3879 (python3)
  Tasks: 10 (limit: 2226)
  Memory: 151.8M
  CPU: 2.345s
  CGroup: /system.slice/odoo17.service
          └─3879 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
          └─3881 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
          └─3882 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
          └─3883 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin gevent -c /etc/odoo
          └─3886 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
          └─3887 /opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odoo17.conf
```

Por último, para parar el servidor usamos el siguiente comando:

sudo systemctl stop odoo17

Este comando detiene el servidor y, al verificar su estado nuevamente, veremos cómo este está inactivo.

Y con esto, ya tendrías Odoo 17 funcionando en Ubuntu Server. :)

```
lermns@lermns:~$ sudo systemctl status odoo17
* odoo17.service - Odoo 17
   Loaded: loaded (/etc/systemd/system/odoo17.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2025-01-15 16:40:41 UTC; 4s ago
     Process: 3879 ExecStart=/opt/odoo17/odoo-venv/bin/python3 /opt/odoo17/odoo/odoo-bin -c /etc/odo
   Main PID: 3879 (code=exited, status=0/SUCCESS)
      CPU: 2.507s
```