Manual de Instalación de Odoo 17 en Ubuntu server

Nombre: Leonardo Ramos Barrenozo

Curso: 2do DAM

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. Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

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.



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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.



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.



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.



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

En caso de no funcionar simplemente use este:

sudo apt install wkhtmltox\_0.12.6.1-3.jammy\_amd64.deb

En algunos casos puede mostrarte solo parte del nombre como:

wkhtmltox\_0.12.6.1-3

Esto es normal y debería de funcionar igualmente.

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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 "geven". Esto se debe a que necesitaremos eliminar algunas líneas problemáticas para poder continuar con la instalación.

Texto

Descripción generada automáticamente

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)

pypiwin32 ; sys\_platform == 'win32'

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

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Una captura de pantalla de una computadora

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente con confianza media

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 incluida la coma).

Texto

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

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.

Captura de pantalla de un celular

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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).

Texto

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

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.



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").

Texto

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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.

Texto

Descripción generada automáticamente

Todo este proceso de agregar nuevos módulos a Odoo 17 permite que el servidor funcione de manera independiente una vez levantado. Esto significa que Ubuntu Server mantendrá el servidor en funcionamiento mientras realiza otras operaciones, a diferencia de un entorno virtual, donde es necesario permanecer activo en la sesión, ya que, al salir, el servidor se detiene automáticamente.

Ahora volvemos a probar usar Odoo en el anfitrión y no debería de dar problemas, pero con ahora podrías mantener el servidor levantado mientras realizamos otras tareas dentro de Ubuntu server.

Interfaz de usuario gráfica

Descripción generada automáticamente

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. :)

Texto

Descripción generada automáticamente