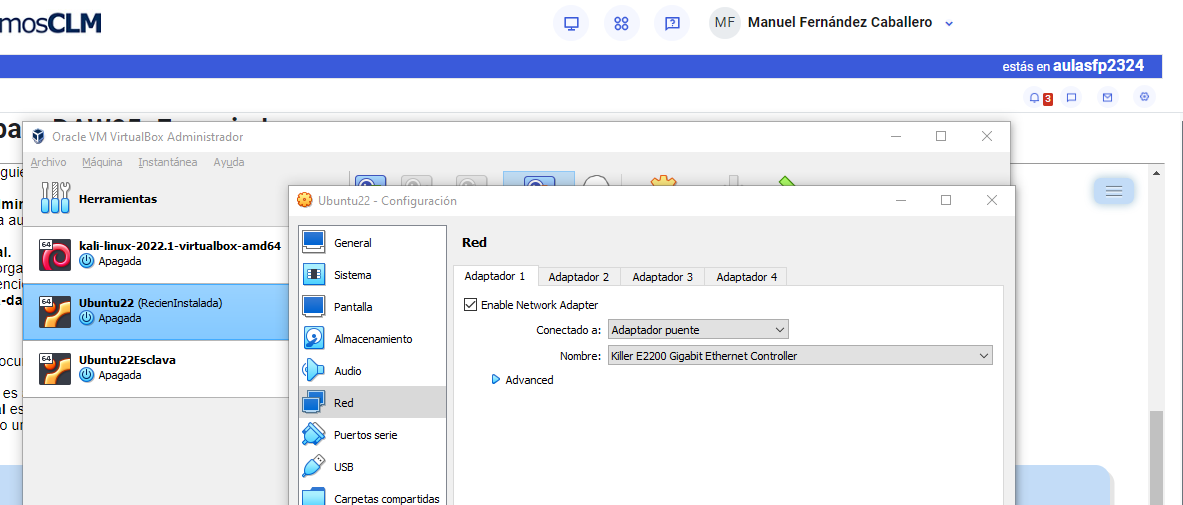
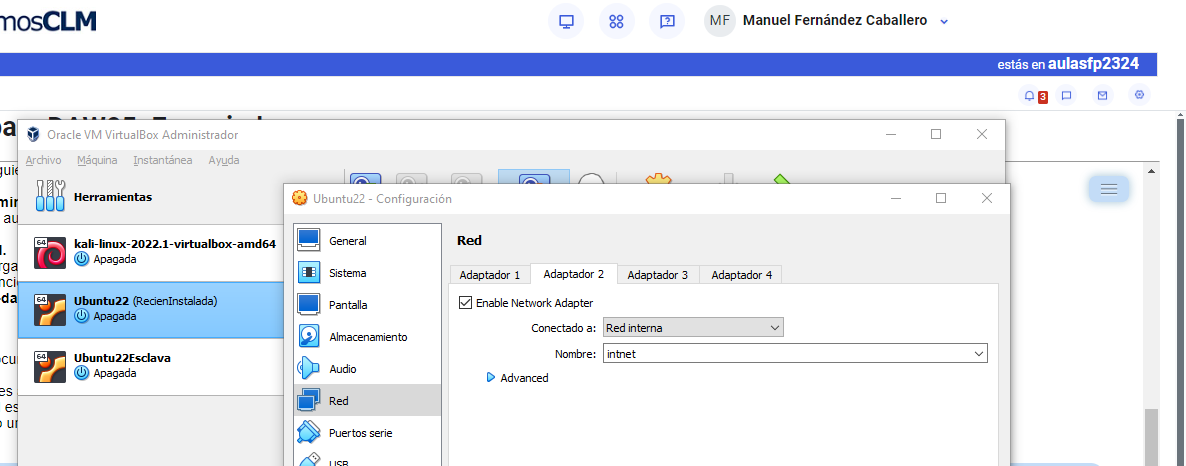
#### ¿Qué te pedimos que hagas?

Se pide:

1. Configurar dos servidores BIND:
   1. Uno como servidor maestro con nombre DNS **ns1.empresa-tarea-daw05.local** en la IP **192.168.200.250.**
   2. Otro como servidor esclavo con nombre en la IP **192.168.200.249.**
   3. Con la posibilidad de transferencia de zona entre ellos.
   4. Donde la zona del servidor maestro contiene los siguientes registros de recursos:
      * **mail** que apunta a la IP **192.168.100.249.**
      * **aplicacion-web** que es un alias de **www.redmine.empresa-tarea-daw05.local.**
   5. Donde la transferencia de zona se realiza de forma automática cuando se modifica la zona en el servidor maestro.

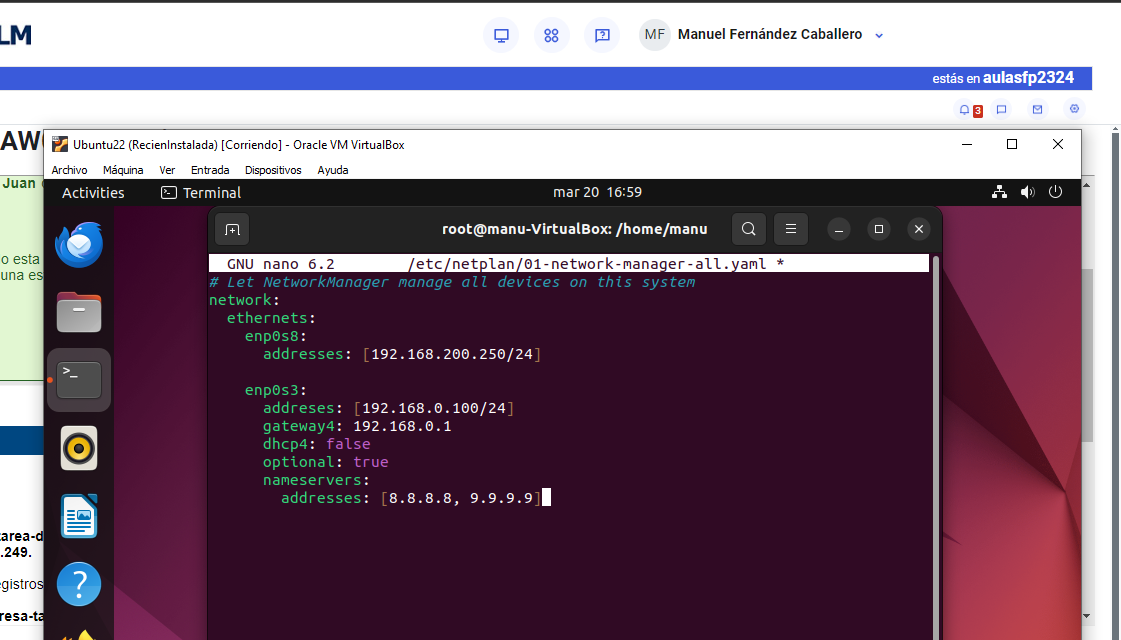
**Para el desarrollo de esta tarea necesitaremos dos máquinas virtuales, una como servidor maestro, y otra como esclavo, para ello clono mi máquina virtual de Ubuntu 22.04.4 LTS y en el servidor maestro “Ubuntu22” en mi caso. Cambio los adaptadores de red, uno en modo bridge y otro en modo red interna:**





**Dentro de la máquina virtual editamos el archivo de configuración de red con el comando:**

**Sudo nano /etc/netplan/01-network-manager-all.yaml**



**Damos permisos con:**

**sudo chmod 600 /etc/netplan/01-network-manager-all.yaml**

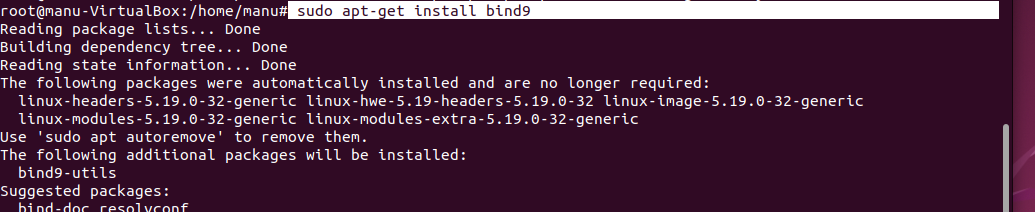
**y actualizamos los los cambios con:**

**Sudo netplan apply**



**Instalamos bind, un software de servidor de nombres de dominio (DNS) de código abierto ampliamente utilizado para la resolución de nombres de dominio en Internet con el comando:**

**Sudo apt-get install bind9**



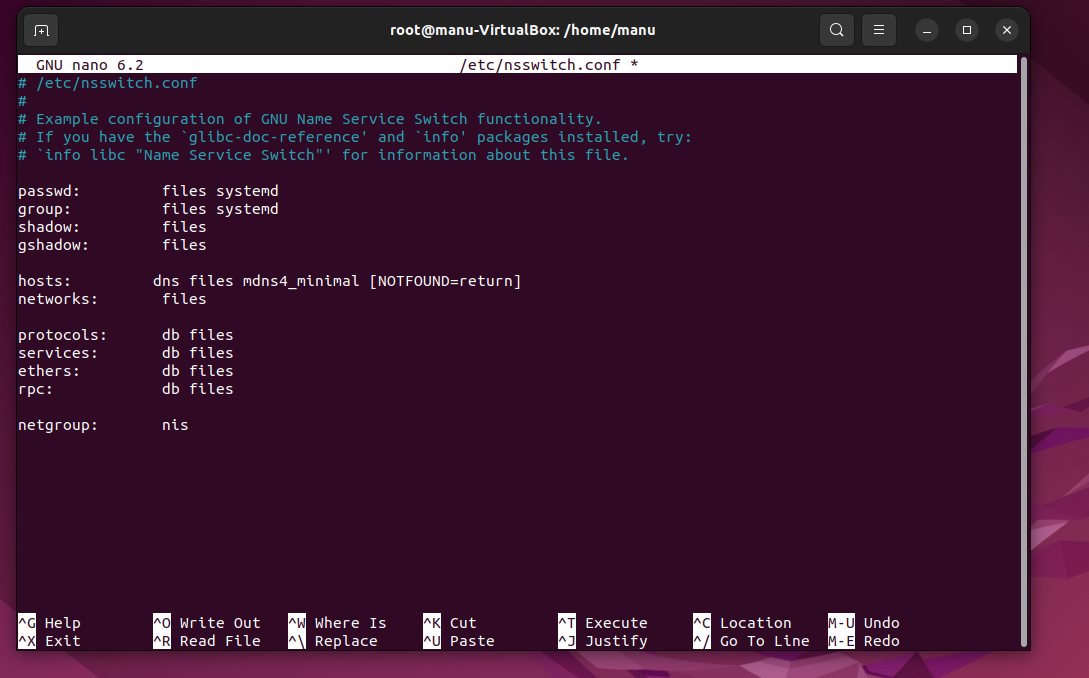
Ejecuto **sudo nano /etc/nsswitch.conf** para abrir el archivo de configuración.

Dentro de ese archivo, busco la línea que tiene la configuración para "hosts".

Cambio la posición de "dns" al principio de la línea.

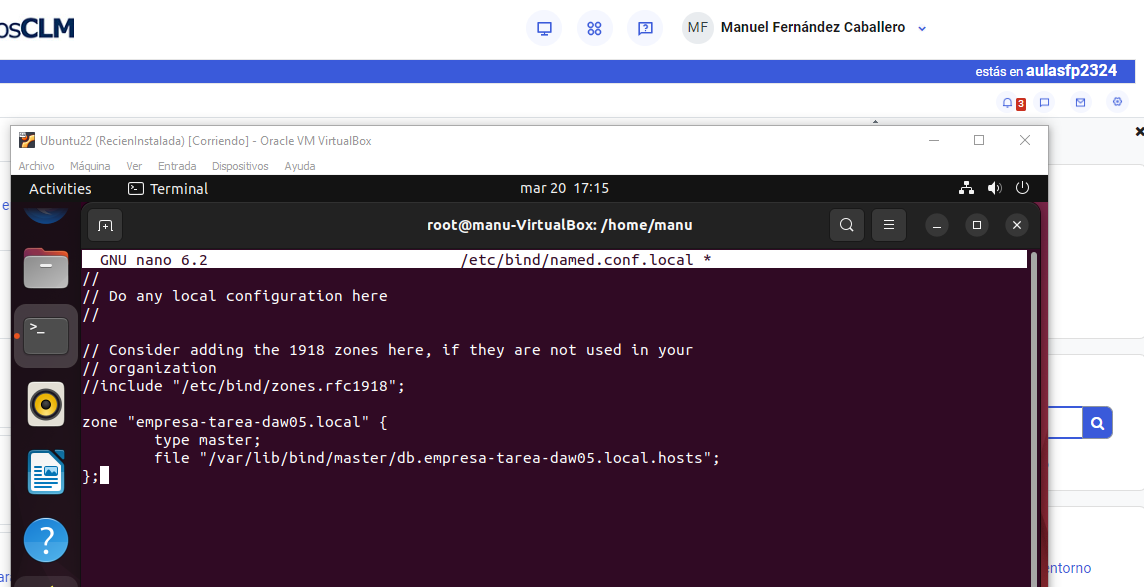
Guardo los cambios y cierro el editor.

Con esto, ahora la búsqueda de hosts se realizará primero en nuestro servidor local antes de buscar en el DNS externo.



Abrimos el archivo de configuración de BIND ejecutando:

**sudo nano /etc/bind/named.conf.local**

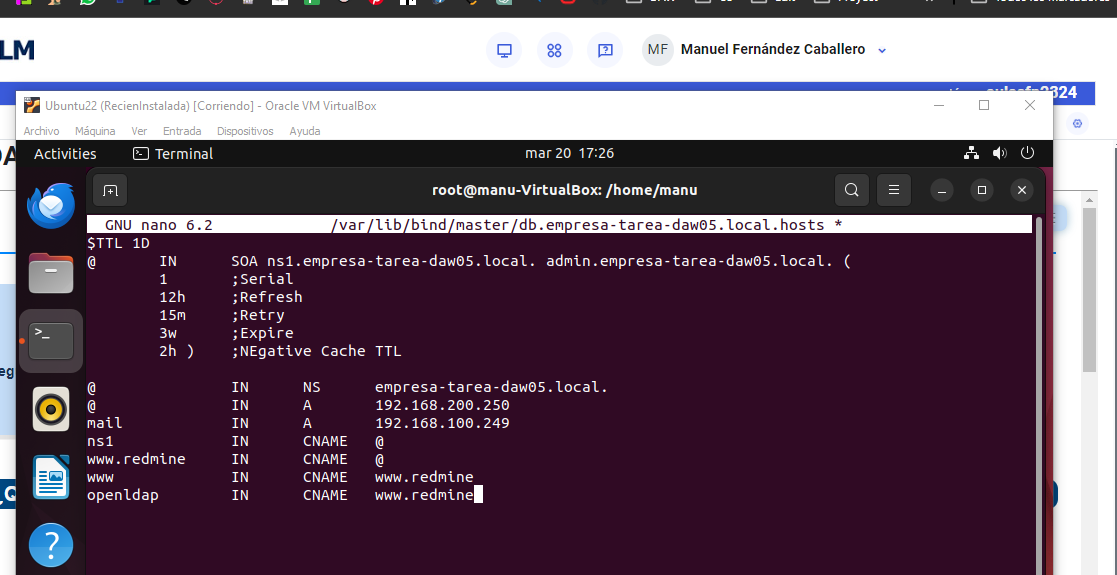


Luego, creamos el directorio donde se guardará el archivo de registro de la zona ejecutando:

**sudo mkdir /var/lib/bind/master/**

A continuación, creamos el archivo de registro de la zona ejecutando:

**sudo nano /var/lib/bind/master/db.empresa-tarea-daw05.local.hosts**

Dentro de ese archivo, introducimos el siguiente código:

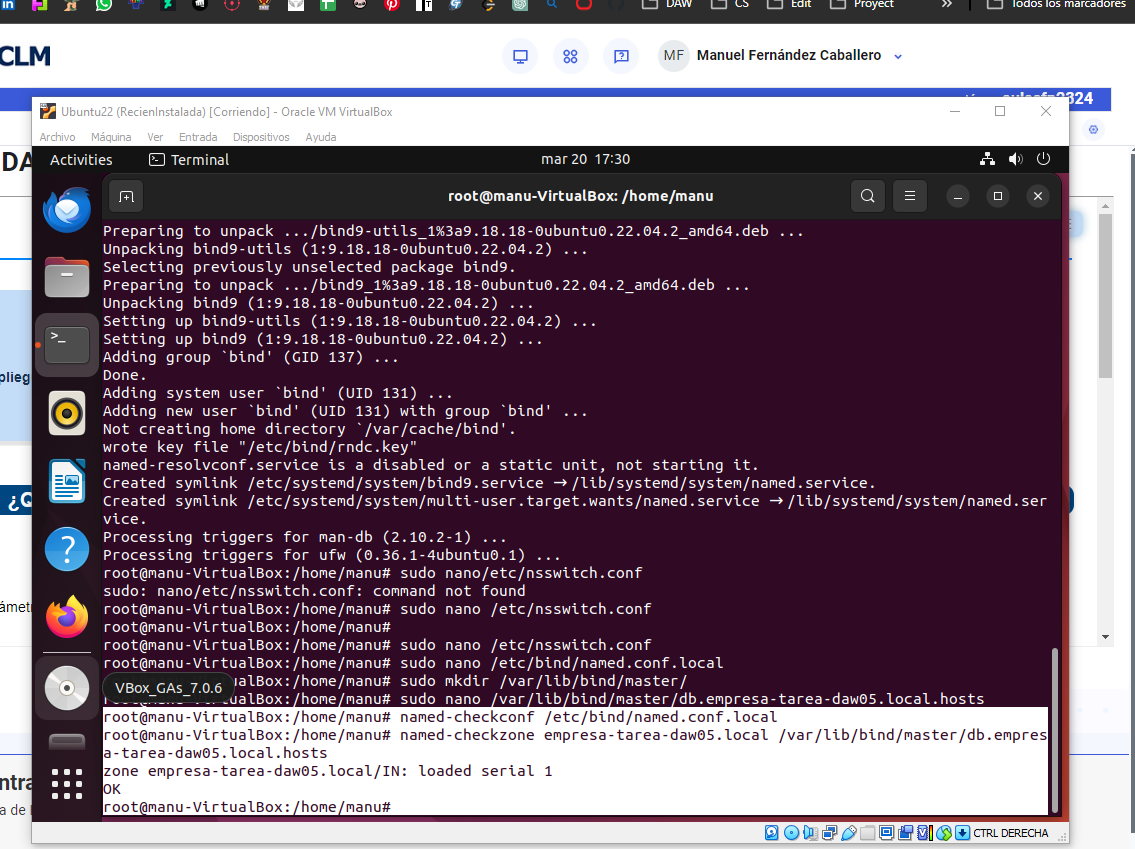
Con estos pasos, configuraremos la zona maestra de BIND y crearemos el archivo de registro de la zona en la ubicación especificada.

Verificamos la sintaxis del archivo de configuración de BIND ejecutando:

**named-checkconf /etc/bind/named.conf.local**

Luego, verificamos la zona creada en el archivo de registro usando:

**named-checkzone empresa-tarea-daw05.local /var/lib/bind/master/db.empresa-tarea-daw05.local.hosts**

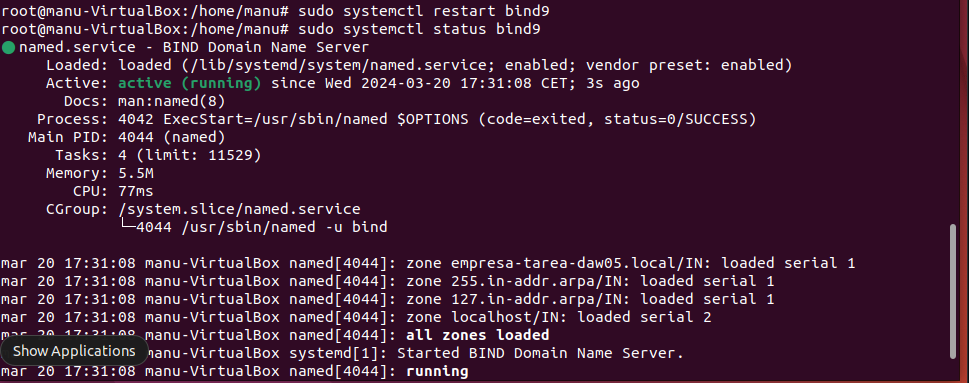


Si ambos comandos no devuelven ningún mensaje de error, reiniciamos el servicio BIND ejecutando:

**sudo systemctl restart bind9**

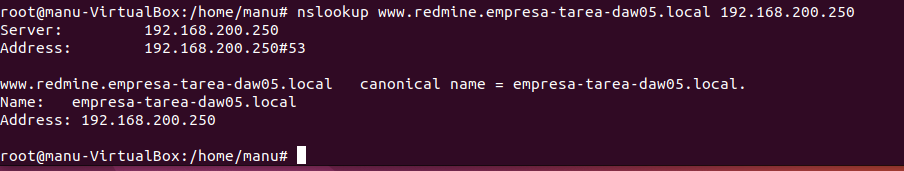
Para verificar que BIND está funcionando correctamente, verificamos su estado con:

**sudo systemctl status bind9**



Para confirmar que las consultas DNS se realizan correctamente, podemos utilizar tanto dig como nslookup. Por ejemplo:

nslookup www.redmine.empresa-tarea-daw05.local 192.168.200.250



Después, procedemos a instalar Redmine, pero antes instalamos Apache2 y el servidor MySQL ejecutando:

**sudo apt install apache2 mysql-server**

Después de la instalación, procedemos a crear un usuario y una contraseña en MySQL, así como una nueva base de datos para Redmine, para lo cual necesitaremos identificarnos como root con:

**mysql -u root -p**

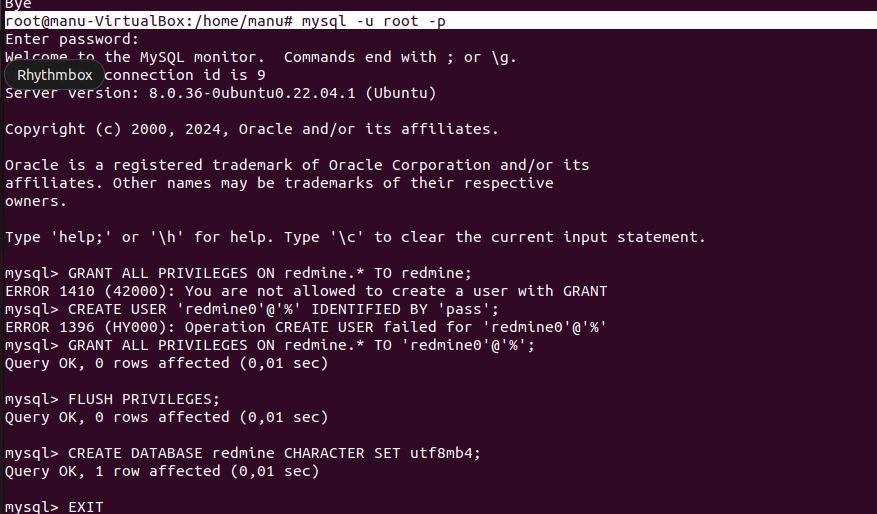
CREATE USER 'redmine0'@'%' IDENTIFIED BY 'pass';

GRANT ALL PRIVILEGES ON redmine.\* TO redmine;

FLUSH PRIVILEGES;

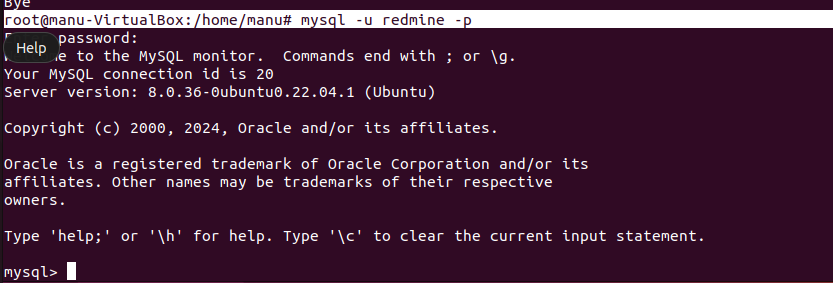
CREATE DATABASE redmine CHARACTER SET utf8mb4;

EXIT



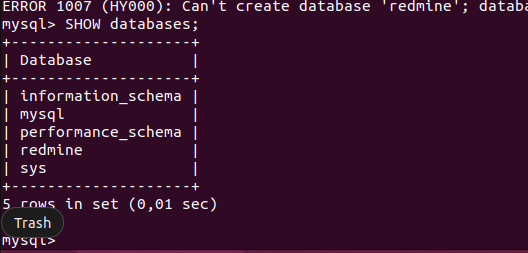
Para comprobar que tanto el usuario como la tabla se han creado correctamente:

**mysql -u redmine -p**



Comprobamos la base de datos:

Con **show databases;**



Instalo **redmine** siguiendo el ODT proporcionado en la tarea y sigo todos los pasos, escribo cada comando tal cuál pero no ha funcionado ya que está deprecated.

* + Actualizar los repositorios del sistema operativo.

# apt-get update

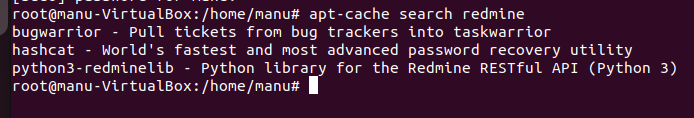
* + Actualiza el sistema operativo.

# apt-get upgrade

* + Buscar si existe el paquete redmine.

# apt-cache search redmine

No aparece en mi versión de Ubuntu así que mostraré una instalación alternativa más adelante



* + Descargar e instalar la aplicación redmine.

# apt-get install redmine

Configuración de redmine

¿Desea configurar la base de datos para redmine/instances/default con <<dbconfig-common>>? → Sí

El tipo de base de datos que se va autilizar para redmine/instances/default: → mysql

Contraseña del usuario de administración de la base de datos → root

Contraseña de aplicación MySQL para redmine/instances/defaul → redmine

Confirmación de contraseña → redmine

redmine-mysql package required … To finish that configuration, please install the redmine-mysql package, and reconfigure redmine using: dpkg-reconfigure -plow redmine → Aceptar

# apt-get install mysql-server redmine-mysql

Nueva contraseña para el usuario <<root>> de MySQL → root

Vuelva a introducir la contraseña para el usuario <<root>> de MySQL: → root

# dpkg-reconfigure -plow redmine

Las instancias de Redmine que se van a configurar o actualizar → default

¿Desea reinstalar la base de datos para redmine/instances/default? → Sí

El tipo de base de datos que se va autilizar para redmine/instances/default: → mysql

Método de conexión para la base de datos MySQL de redmin/instances/default: → socket unix

Nombre del usuario de administración de la base de datos: → root

Nombre de usuario para redmine/instances/default → redmine

Nombre de la base de datos para redmine/instances/default → redmine\_default

Idioma predeterminado para Redmine → es

Creating config file /etc/redmine/default/session.yml with new version

A new secret session key has been generated in /etc/redmine/default/session.yml

Populating database for redmine instance “default”.

# apt-get install gem

# gem install -v=0.4.2 i18n

# gem install rack -v=1.1.1

# RAILS\_ENV=production rake db:migrate

# cp -pv config/database.yml.example config/database.yml

* + Editar **database.yml** tal que así:

production:

adapter: mysql

database: redmine\_default

host: localhost

username: root

password: root

encoding: utf8

* + Ejecutar:

ruby script/server webrick -e production

* + Lanza en el navegador [http://192.168.200.250:3000](http://192.168.200.250:3000/) (o bien configura en tu sistema como servidor DNS la IP 192.168.200.250) y accede con el usuario por defecto 'admin' con contraseña 'admin'.

No ha funcionado así que utilizo una instalación alternativa:

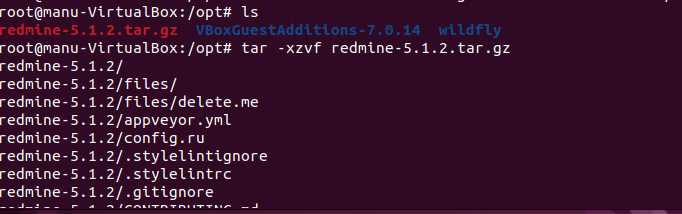
Accedo al directorio opt

*cd /opt/*

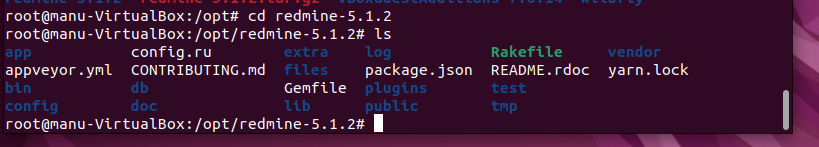
Descargo redmine 5.1.2 y descomprimo

*wget https://www.redmine.org/releases/redmine-5.1.2.tar.gz*

*tar -xvzf redmine-5.1.2.tar.gz  
sudo ln -s redmine-5.1.2 redmine*

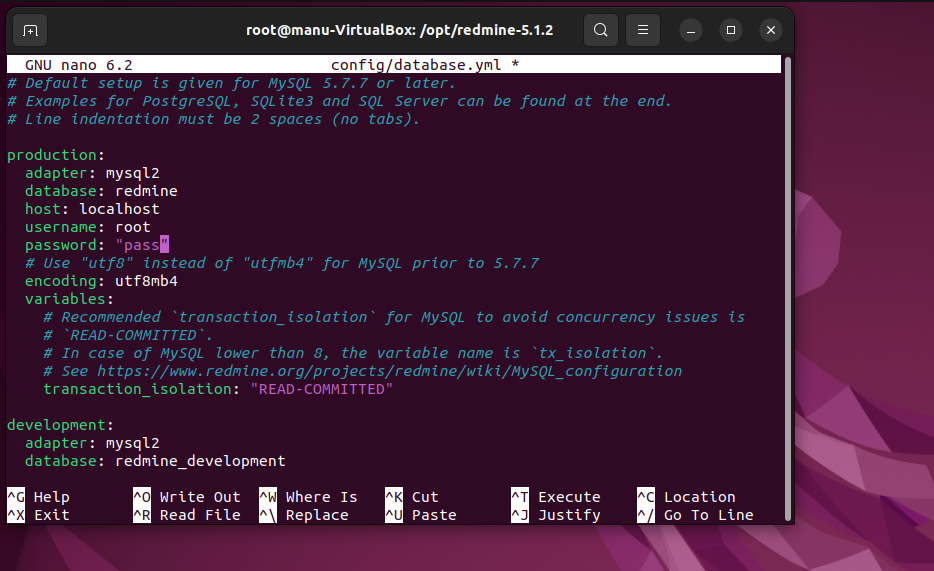


*cd /opt/redmine-5.1.2/*



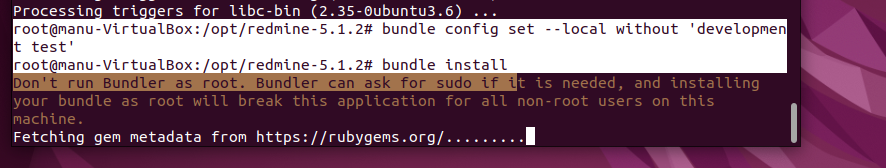
Copiamos la base de dato predefinida y seteamos el usuario previo con la contraseña previa para utilizar Mysql/mariaDB

*cp config/database.yml.example config/database.yml  
nano config/database.yml*



Instalamos las dependencias necesarias con: apt install ruby-bundler

*bundle config set --local without 'development test'  
bundle install*

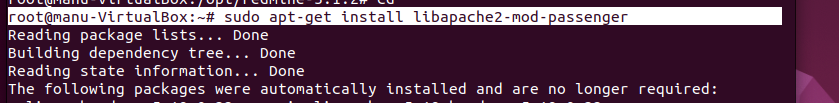


Una vez que se han instalado las dependencias de Redmine, ejecuta el siguiente comando para generar el token secreto de Redmine y migrar la base de datos.

*bundle exec rake generate\_secret\_token  
RAILS\_ENV=production bundle exec rake db:migrate*

Ahora, instalamos el módulo Passenger:

**sudo apt-get install libapache2-mod-passenger**

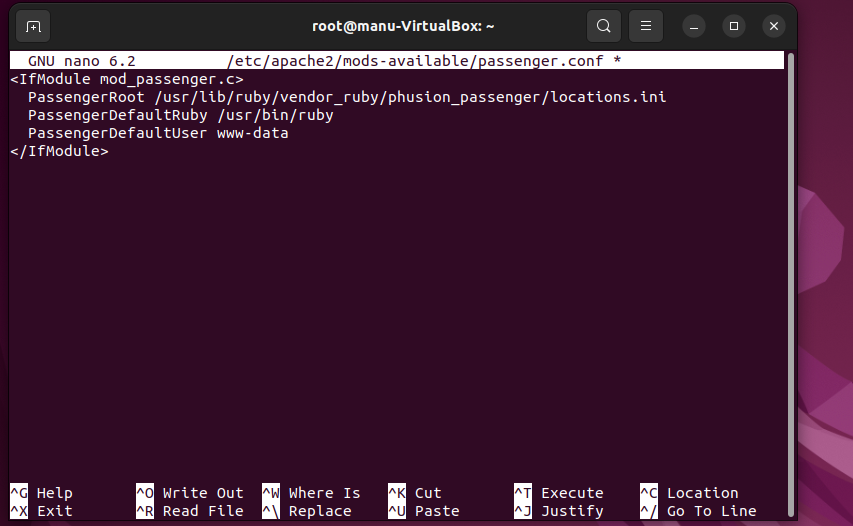


Y configuramos el archivo **passenger.conf** añadiendo la línea:

**sudo nano /etc/apache2/mods-available/passenger.conf**

Después de abrir el archivo con nano, añado la siguiente línea:

**PassengerDefaultUser www-data**



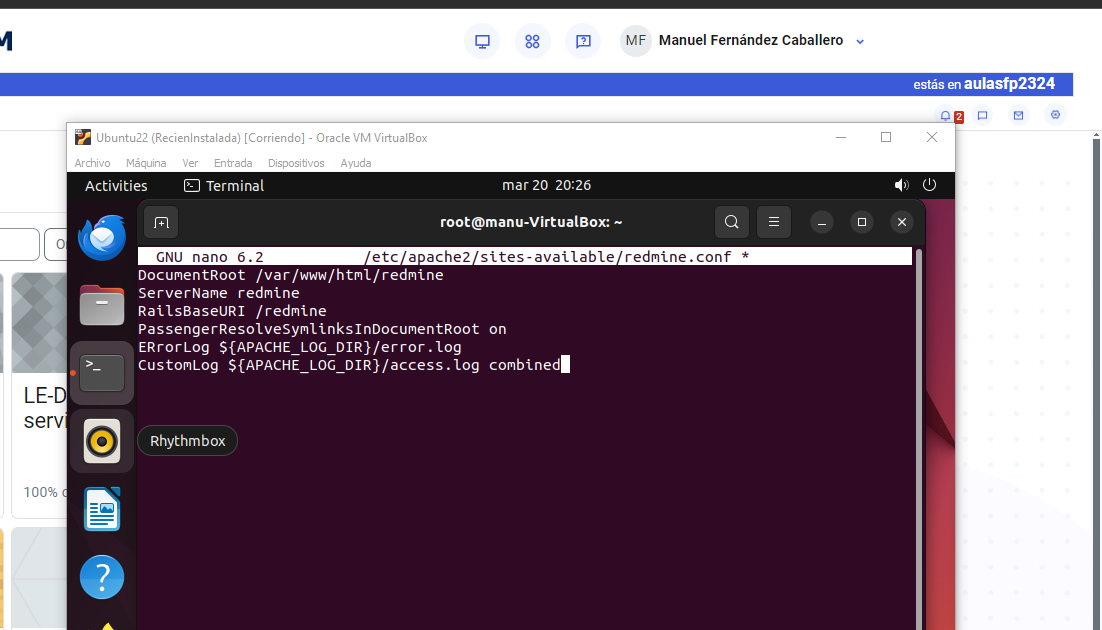
Creamos un enlace simbólico entre la carpeta de redmine y la carpeta de redmine que tenemos dentro del servidor apache:



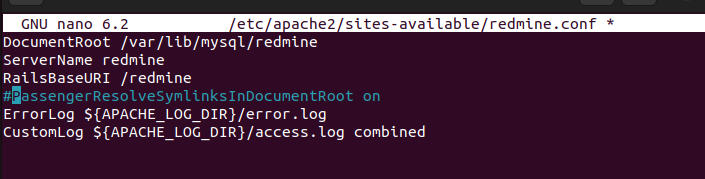
**sudo ln -s /usr/share/redmine/public/ /var/www/html/redmine**

Y configuramos un virtualhost para redmine:

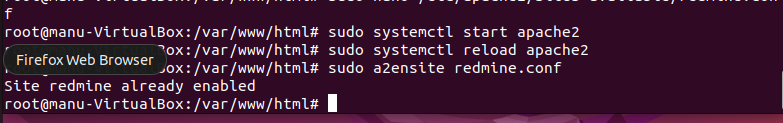
**sudo nano /etc/apache2/sites-available/redmine.conf**



Me da error el servidor apache, así que corrijo la ubicación de redmine y comento Passenger:



Ahora activamos el sitio con **sudo a2ensite redmine.conf** y reiniciamos el servicio apache con **sudo systemctl restart apache2** y no obtengo ningún error

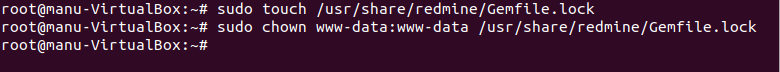


Creo el archivo .lock y situo al usuario www-data como dueño con:

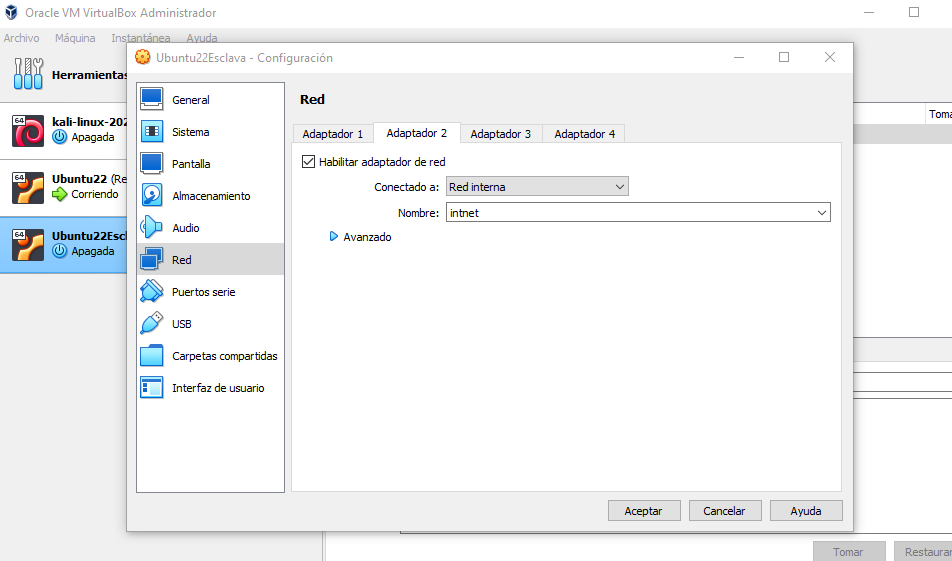
sudo touch /usr/share/redmine/Gemfile.lock

Permisos con:

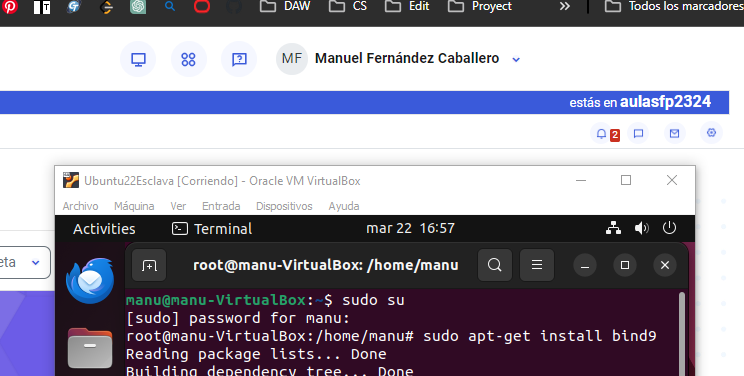
sudo chown www-data:www-data /usr/share/redmine/Gemfile.lock



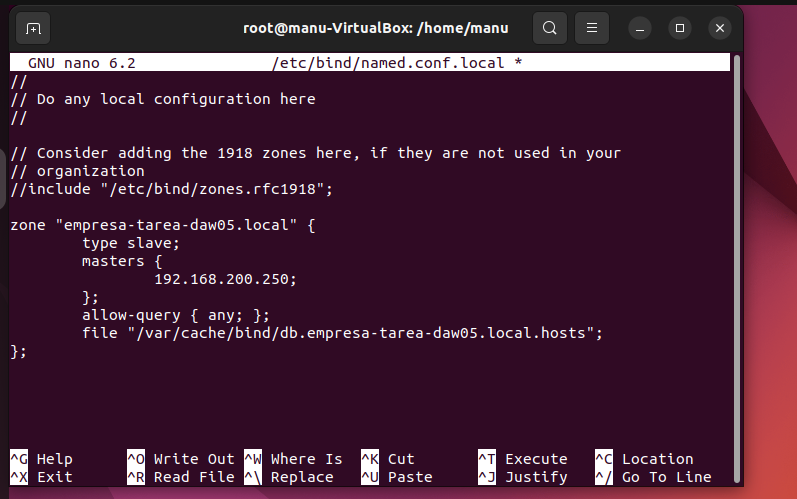
Ahora en la máquina virtual que actuará como servidor esclavo vamos a poner un segundo adaptador como red interna:



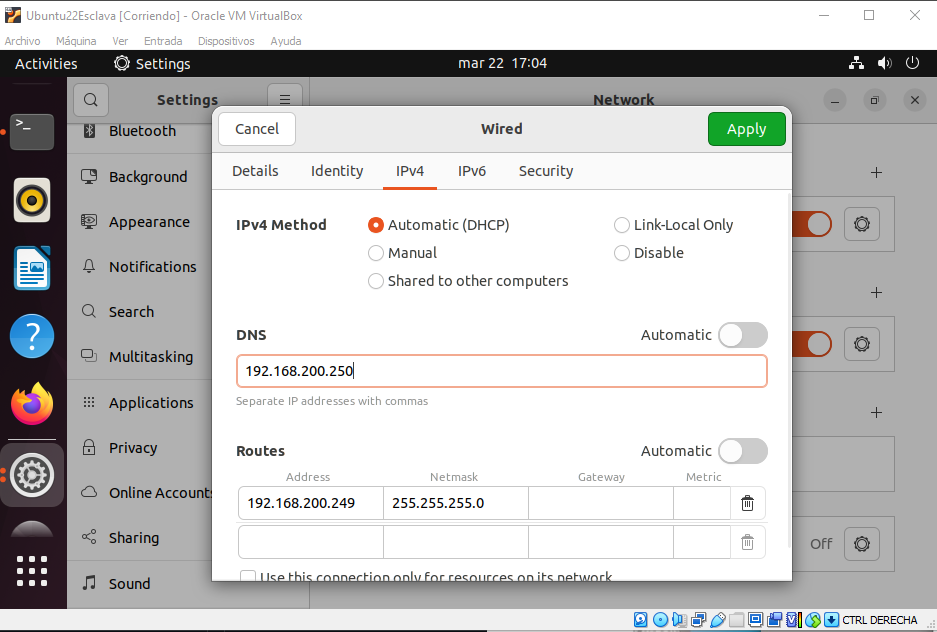
E instalamos **bind9** de la misma manera **con sudo apt-get install bind9** y lo coniguramos para actuar como servidor esclavo en el fichero:



**sudo nano /etc/bind/named.conf.local**



Ahora configuro la interfaz de red de forma gráfica desde ubuntu para que tenga **como IP 192.168.200.249, máscara de subred 255.255.255.0** y use como **servidor DNS el del servidor maestro, 192.168.200.250**

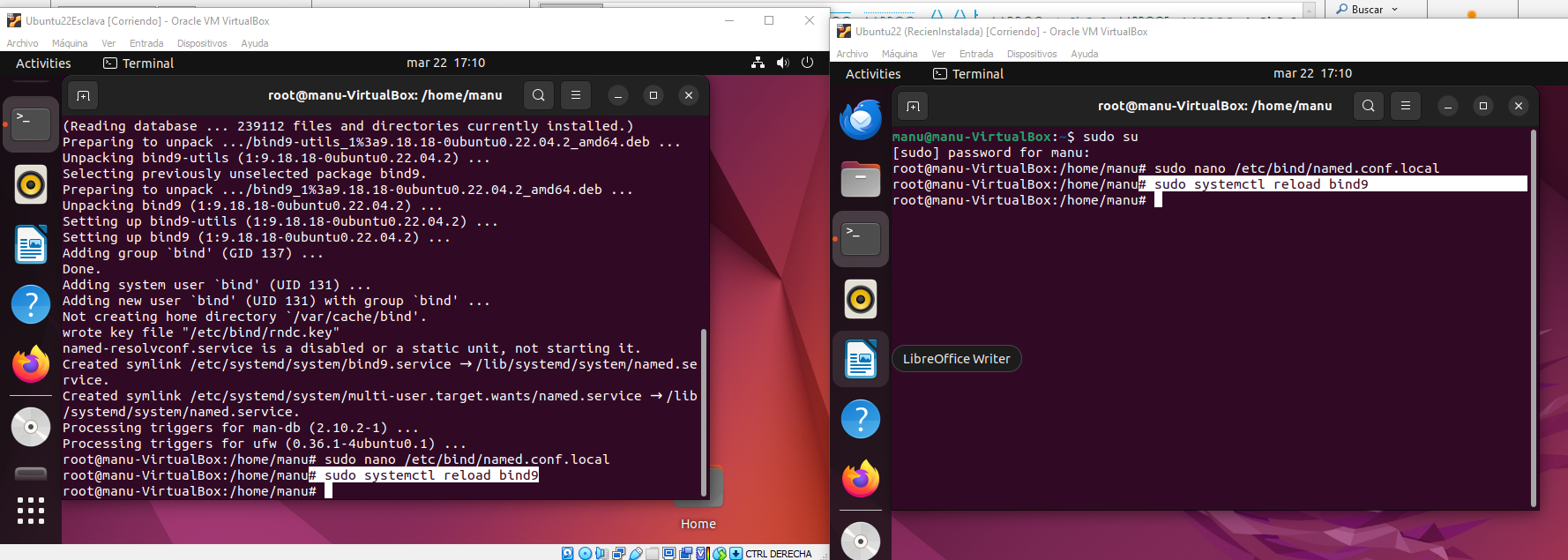


Ahora, en el servidor maestro modificamos el archivo de configuracion e indicamos que hay un servidor esclavo con la directiva allow-transer:

**sudo nano /etc/bind/named.conf.local**



Reinicio el servicio bind 9 en el servidor maestro y en el esclavo con **sudo systemctl reload bind9**



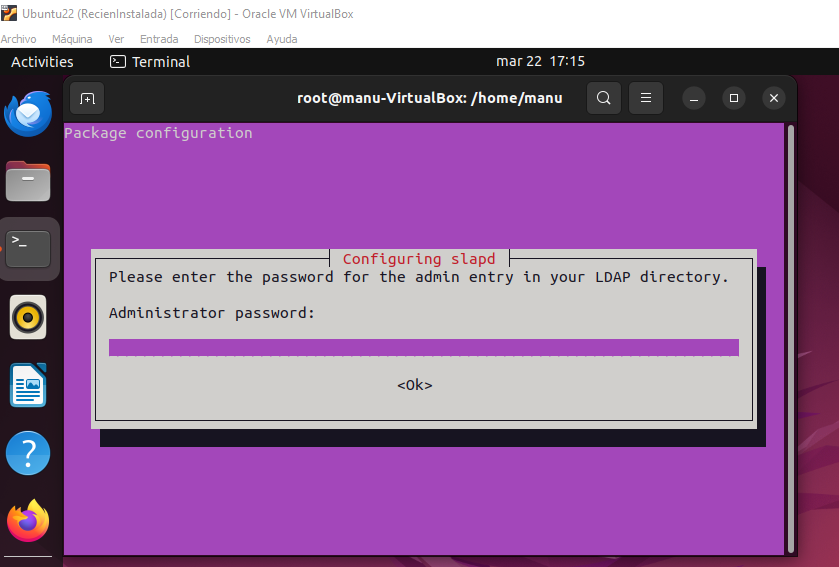
Con esto el esclavo se descarga el host del maestro.

1. Configurar un servidor OpenLDAP con:
   1. El nombre de **dominioempresa-tarea-daw05.local.**
   2. Una estructura básica para atender a una unidad organizativa que contenga el departamento de atención al cliente.
   3. Un usuario que pertenezca al departamento de atención al cliente: **op1** con contraseña **oper**.

Vuelvo al servidor maestro e instalo OpenLDAP

**sudo apt-get install slapd ldap-utils**

Nos pedira pass de administrador:

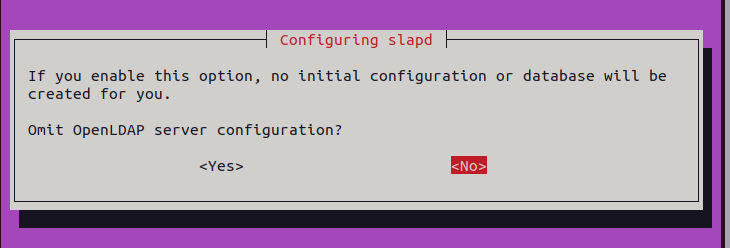


Una vez finalizada la isntalacion, añadimos el nombre del dominio con el comando

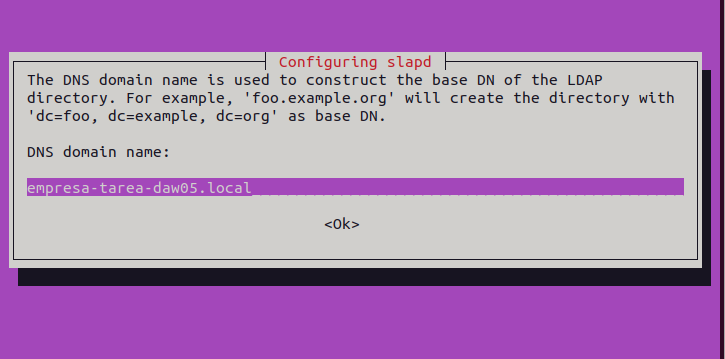
**sudo dpkg-reconfigure slapd**

y constamos a las siguientes preguntas:

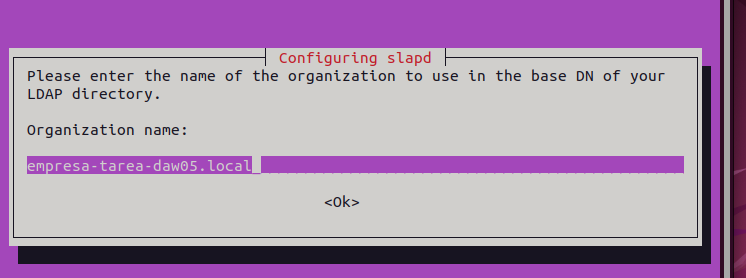
¿Desea omitir la configuración?



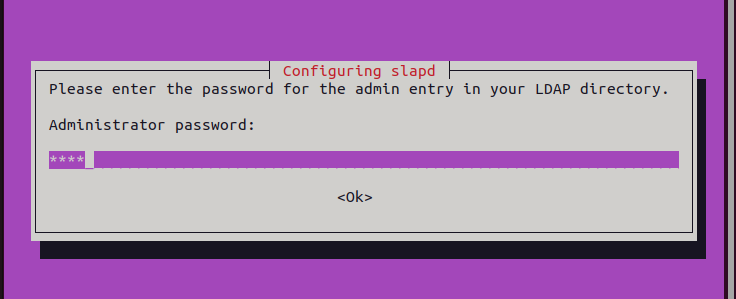
Nombre del dominio DNS:



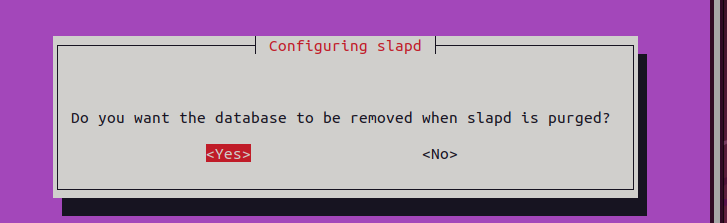
Nombre de la organización:



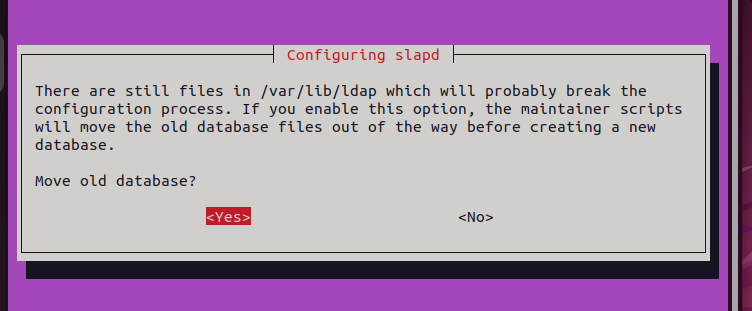
Nos pide contraseña para el administrador:



¿Desea que se borre la base de datos cuando se purgue el paquete slapd? Sí

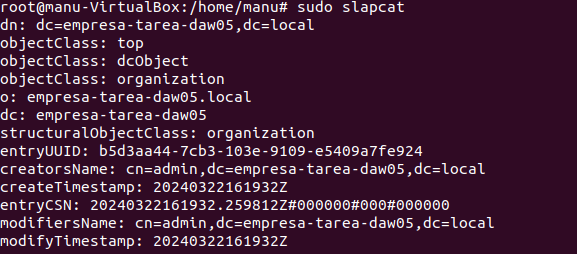


¿Desea mover la base de datos antigua? Sí



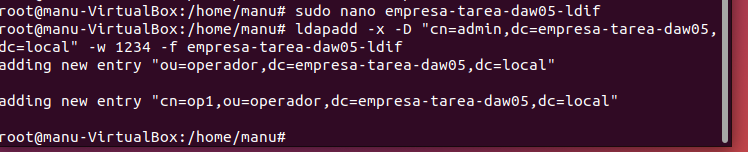
Comprobamos lo que hay en el dominio con

sudo slapcat



Lo importamos con el comando :

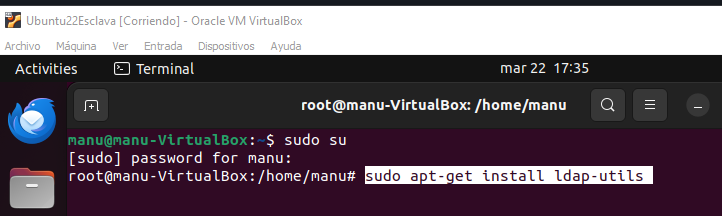
**ldapadd -x -D "cn=admin,dc=empresa-tarea-daw05,dc=local" -w 1234 -f empresa-tarea-daw05-ldif**



1. Revisar toda la estructura **DIT** del dominio **empresa-tarea-daw05.local** a través del servidor DNS esclavo.

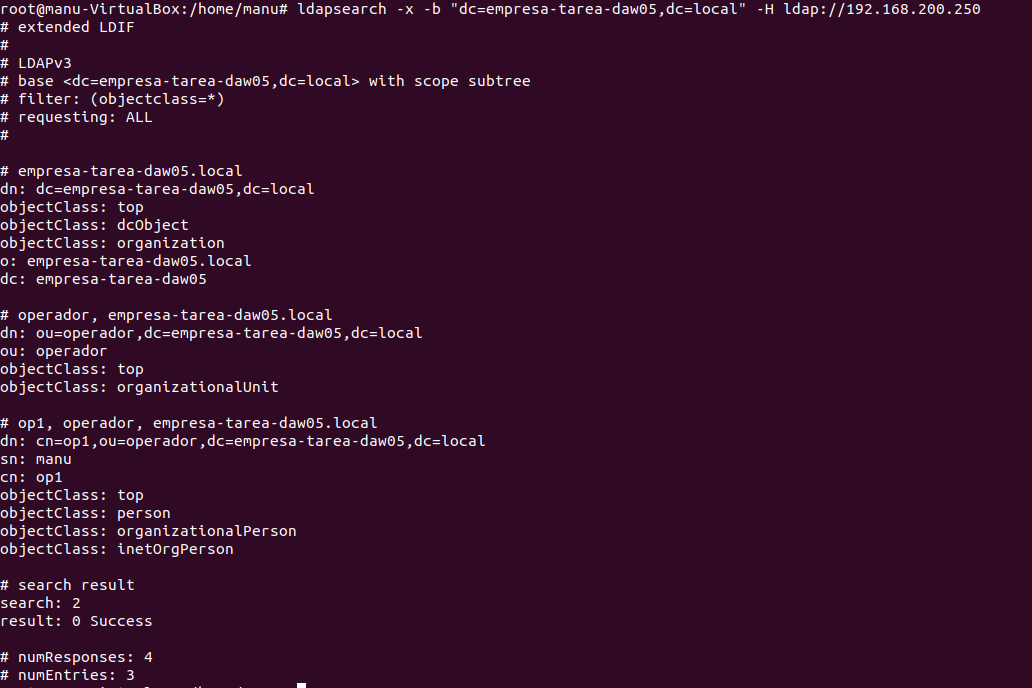
Instalamos de la misma manera ldap-utils en el servidor esclavo:

**sudo apt-get install ldap-utils**



Una vez instalado podremos llamar al servidor maestro para que nos devuelva informacion sobre la estructura del LDAP con

**ldapsearch -x -b "dc=empresa-tarea-daw05,dc=local" -H ldap://192.168.200.250**



Y comprobamos que es la misma información que teníamos en el servidor.