

Zammad Ticket System

Implantación de sistemas operativos ASIR1.

Alexandre Carballo Pereira

INDICE

1. FUNCIONAMIENTO DE ZAMMAD
2. REQUISITOS PARA LA INSTALACIÓN DE ZAMMAD
3. INSTALACIÓN DE NODE.JS Y YARN
4. INSTALANDO RUBY CON RBENV
5. INSTALACIÓN DE NGINX
6. INSTALACIÓN POSTGRESQL
7. INSTALACIÓN DE ELASTICSEARCH
8. INSTALACIÓN ZAMMAD
9. CONFIGURANDO ZAMMAD
10. BIBLIOGRAFÍA

FUNCIONAMIENTO DE ZAMMAD

Zammad es una plataforma de código abierto para la gestión de tickets y soporte al cliente. Esta herramienta proporciona a las empresas y organizaciones una solución completa para la gestión de tickets, que les permite gestionar y seguir de manera eficiente y efectiva todas las solicitudes de soporte de los clientes.

Entre las principales funcionalidades de Zammad se encuentran la gestión de tickets, la integración con múltiples canales de comunicación, la automatización de procesos, la gestión de conocimiento, el análisis de datos y la colaboración en equipo.

La gestión de tickets de Zammad permite a los equipos de soporte al cliente crear, actualizar y cerrar tickets de manera rápida y sencilla. Además, los tickets pueden ser etiquetados y categorizados para una mejor organización y seguimiento.

Zammad también permite la integración con múltiples canales de comunicación, como correo electrónico, chat, redes sociales y telefonía, lo que permite una gestión centralizada y eficiente de todas las solicitudes de soporte.

La gestión de conocimiento en Zammad permite la creación y almacenamiento de bases de datos de información para la solución de problemas comunes, lo que ayuda a los agentes de soporte a responder de manera rápida y efectiva a las solicitudes de los clientes, la colaboración en equipo en Zammad permite a los equipos de soporte trabajar juntos de manera eficiente y efectiva, lo que mejora la calidad del servicio al cliente y aumenta la satisfacción del cliente.

En resumen, Zammad es una solución completa para la gestión de tickets y soporte al cliente, que proporciona una amplia gama de funcionalidades para la gestión eficiente y efectiva de todas las solicitudes de soporte.

REQUISITOS PARA LA INSTALACIÓN DE ZAMMAD

Para la correcta instalación de Zammad necesitamos un ordenador con un procesador con mínimo 4 núcleos y 4 GB de RAM para un uso básico de la plataforma, para un entorno más profesional se recomienda mínimo 8GB de RAM, en cuanto almacenamiento con 20GB sería suficiente, se requiere conexión a Internet estable para acceder a la plataforma.

Los requisitos mínimos de Software, necesitamos un navegador web actualizado, ya que puede dar errores en navegadores antiguos.

También necesitaremos tener instalado Node.js, Yarn, Ruby con rbnb, un servidor web como Nginx, una base de datos como PostgreSQL y por último para mejorar la utilización de la base de datos con el sistema de tickets Elasticsearch.

INSTALACIÓN DE NODE.JS Y YARN

Empezaremos por instalar Node.js en el server con el paquete yarn. La versión más reciente de Node.js no sale en los repositorios de Debian 11 así que tendremos que añadir al repositorio fuente al APT usando el siguiente comando:

```
sudo curl -sL https://deb.nodesource.com/setup_16.x |sudo bash -
```

Después de añadir el repositorio usaremos el siguiente comando para instalar Node.js:

```
sudo apt-get install nodejs -y
```

Después añadiremos Yarn a los repositorios con el siguiente comando

```
sudo curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg |sudo apt-key add -
```

```
sudo echo "deb https://dl.yarnpkg.com/debian/ stable main" |sudo tee  
/etc/apt/sources.list.d/yarn.list
```

Una vez tenemos añadidos los repositorios usaremos los siguiente comandos para hacer una actualización y la instalación de Yarn:

```
sudo apt-get update -y  
sudo apt-get install yarn -y
```

INSTALANDO RUBY CON RBENV

Lo primero será instalar los siguientes paquetes para poder llevar acabo todo lo que necesitamos para una correcta instalación:

```
sudo apt-get install curl gnupg2 dirmngr git-core libcurl4-openssl-dev software-properties-common zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libffi-dev -y
```

```
Configurando cpp (4:10.2.1-1) ...
Configurando libpolkit-agent-1-0:amd64 (0.105-31+deb11u1) ...
Configurando libc6-dev:amd64 (2.31-13+deb11u5) ...
Configurando libfontconfig1:amd64 (2.13.1-4.2) ...
Configurando libicu-dev:amd64 (67.1-7) ...
Configurando policykit-1 (0.105-31+deb11u1) ...
Configurando binutils-x86-64-linux-gnu (2.35.2-2) ...
Configurando libncurses-dev:amd64 (6.2+20201114-2) ...
Configurando libstdc++-10-dev:amd64 (10.2.1-6) ...
Configurando binutils (2.35.2-2) ...
Configurando libreadline-dev:amd64 (8.1-1) ...
Configurando dpkg-dev (1.20.12) ...
Configurando gir1.2-packagekitglib-1.0 (1.2.2-2) ...
Configurando libxml2-dev:amd64 (2.9.10+dfsg-6.7+deb11u4) ...
Configurando libsqlite3-dev:amd64 (3.34.1-3) ...
Configurando python3-gi (3.38.0-2) ...
Configurando libgd3:amd64 (2.3.0-2) ...
Configurando gcc-10 (10.2.1-6) ...
Configurando zlib1g-dev:amd64 (1:1.2.11.dfsg-2+deb11u2) ...
Configurando packagekit (1.2.2-2) ...
Created symlink /etc/systemd/user/sockets.target.wants/pk-debconf-helper.socket → /usr/lib/systemd/user/pk-debconf-helper.socket.
Configurando libc-devtools (2.31-13+deb11u5) ...
Configurando packagekit-tools (1.2.2-2) ...
Configurando g++-10 (10.2.1-6) ...
Configurando libxslt1-dev:amd64 (1.1.34-4+deb11u1) ...
Configurando software-properties-common (0.96.20.2-2.1) ...
Configurando gcc (4:10.2.1-1) ...
Configurando g++ (4:10.2.1-1) ...
update-alternatives: utilizando /usr/bin/g++ para proveer /usr/bin/c++ (c++) en modo automático
Configurando build-essential (12.9) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para dbus (1.12.24-0+deb11u1) ...
Procesando disparadores para libc-bin (2.31-13+deb11u5) ...
root@debian:~# apt install curl gnupg2 dirmngr git-core libcurl4-openssl-dev software-properties-common zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libffi-dev -y_
```

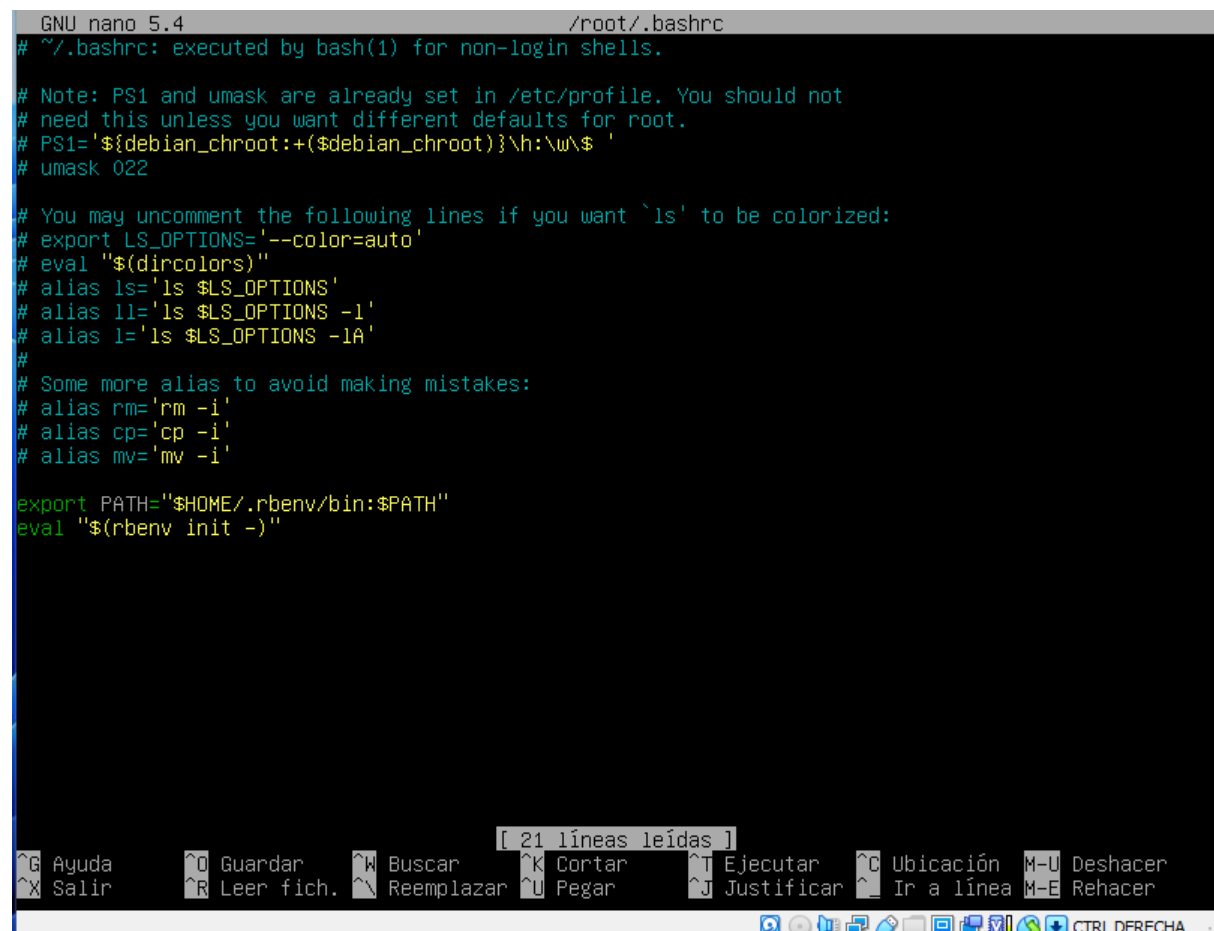
La aplicación rbenv nos facilita la instalación y control de diferentes versiones de Ruby, así que conseguiremos el código fuente de un repositorio de Git:

```
sudo git clone https://github.com/rbenv/rbenv.git ~/.rbenv
```

```
Archivo  maquina  vc  Entada  dispositivos  Ayuda
root@debian:~# git clone http://github.com/rbenv/rbenv.git
Clonando en 'rbenv'...
warning: redirigiendo a https://github.com/rbenv/rbenv.git/
remote: Enumerating objects: 3189, done.
remote: Counting objects: 100% (339/339), done.
remote: Compressing objects: 100% (198/198), done.
remote: Total 3189 (delta 189), reused 256 (delta 129), pack-reused 2850
Recibiendo objetos: 100% (3189/3189), 642.32 KiB | 3.89 MiB/s, listo.
Resolviendo deltas: 100% (1979/1979), listo.
root@debian:~# ls
rbenv
root@debian:~# pwd
/root
root@debian:~# ls
rbenv
root@debian:~# mv rbenv/ .rbenv
root@debian:~# ls
root@debian:~# ls -l
total 0
root@debian:~# ls -la
total 28
drwx----- 4 root root 4096 abr 21 17:11 .
drwxr-xr-x 18 root root 4096 feb  1 20:56 ..
-rw----- 1 root root  576 abr 21 16:47 .bash_history
-rw-r--r-- 1 root root  571 abr 10  2021 .bashrc
drwxr-xr-x 3 root root 4096 feb  1 21:00 .local
-rw-r--r-- 1 root root  161 jul  9  2019 .profile
drwxr-xr-x 11 root root 4096 abr 21 17:11 .rbenv
root@debian:~# _
```

Después usaremos el siguiente comando para exportar el PATH de rbenv:

```
sudo echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
sudo echo 'eval "$(rbenv init -)"' >> ~/.bashrc
exec $SHELL
```



```
GNU nano 5.4 /root/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

export PATH="$HOME/.rbenv/bin:$PATH"
eval "$(rbenv init -)"

[ 21 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación  M-U Deshacer
^X Salir  ^R Leer fich.  ^N Reemplazar  ^U Pegar  ^J Justificar  ^_ Ir a línea  M-E Rehacer
CTRL DERECHA
```

Ahora pillaremos del repositorio de Git la build de Ruby.

```
sudo git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
```

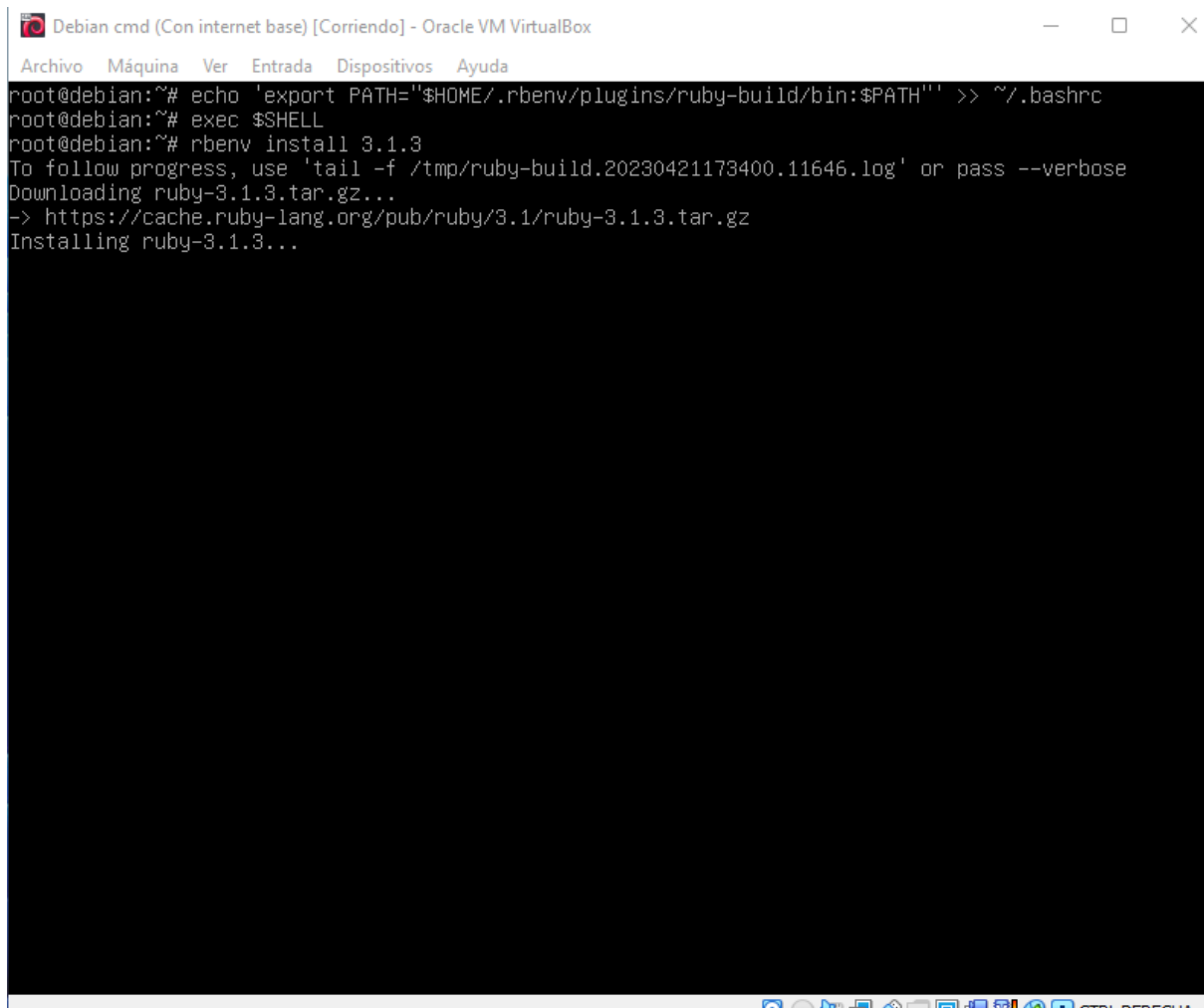
```
root@debian:~# git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
Clonando en '/root/.rbenv/plugins/ruby-build'...
remote: Enumerating objects: 13936, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 13936 (delta 55), reused 31 (delta 18), pack-reused 13850
Recibiendo objetos: 100% (13936/13936), 2.78 MiB | 12.67 MiB/s, listo.
Resolviendo deltas: 100% (9537/9537), listo.
root@debian:~#
```

Ahora con el siguiente comando para añadir el path de Ruby-build

```
sudo echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
exec $SHELL
```

Ahora usaremos el siguiente comando para instalar la versión mas reciente de Ruby

```
rbenv install 3.1.3
```



```
Debian cmd (Con internet base) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@debian:~# echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
root@debian:~# exec $SHELL
root@debian:~# rbenv install 3.1.3
To follow progress, use 'tail -f /tmp/ruby-build.20230421173400.11646.log' or pass --verbose
Downloading ruby-3.1.3.tar.gz...
-> https://cache.ruby-lang.org/pub/ruby/3.1/ruby-3.1.3.tar.gz
Installing ruby-3.1.3...
```


Ahora instalaremos por último el paquete rails, el cual facilita el desarrollo de aplicaciones web en Ruby:

```
gem install rails
```

```
Parsing documentation for marcel-1.0.2
Installing ri documentation for marcel-1.0.2
Parsing documentation for activemodel-7.0.4.3
Installing ri documentation for activemodel-7.0.4.3
Parsing documentation for activerecord-7.0.4.3
Installing ri documentation for activerecord-7.0.4.3
Parsing documentation for globalid-1.1.0
Installing ri documentation for globalid-1.1.0
Parsing documentation for activejob-7.0.4.3
Installing ri documentation for activejob-7.0.4.3
Parsing documentation for activestorage-7.0.4.3
Installing ri documentation for activestorage-7.0.4.3
Parsing documentation for actiontext-7.0.4.3
Installing ri documentation for actiontext-7.0.4.3
Parsing documentation for mail-2.8.1
Installing ri documentation for mail-2.8.1
Parsing documentation for actionmailer-7.0.4.3
Installing ri documentation for actionmailer-7.0.4.3
Parsing documentation for actionmailbox-7.0.4.3
Installing ri documentation for actionmailbox-7.0.4.3
Parsing documentation for websocket-extensions-0.1.5
Installing ri documentation for websocket-extensions-0.1.5
Parsing documentation for websocket-driver-0.7.5
Installing ri documentation for websocket-driver-0.7.5
Parsing documentation for nio4r-2.5.9
Installing ri documentation for nio4r-2.5.9
Parsing documentation for actioncable-7.0.4.3
Installing ri documentation for actioncable-7.0.4.3
Parsing documentation for rails-7.0.4.3
Installing ri documentation for rails-7.0.4.3
Done installing documentation for zeitwerk, thor, method_source, concurrent-ruby, tzinfo, i18n, activesupport, nokogiri, crass, loofah, rails-html-sanitizer, rails-dom-testing, rack, rack-test, erubi, builder, actionview, actionpack, railties, mini_mime, marcel, activemodel, activerecord, globalid, activejob, activestorage, actiontext, mail, actionmailer, actionmailbox, websocket-extensions, websocket-driver, nio4r, actioncable, rails after 25 seconds
35 gems installed
root@debian:~# gem install rails
```

Por último haremos una actualización de RubyGem a su última versión:

```
gem update --system
```

También podremos comprobar la versión con `gem -v` y es recomendable desactivar la instalación del paquete Ruby de documentación a través de RubyGem:

```
Echo "gem: --no-document" << ~/.gemrc
```

```
* Add `mswin` support for cargo builder. Pull request [#6167] (https://github.com/rubygems/rubygems/pull/6167) by ianks
* Validate Cargo.lock is present for Rust based extensions. Pull request [#6151] (https://github.com/rubygems/rubygems/pull/6151) by simi
* Clean built artifacts after building extensions. Pull request [#6133] (https://github.com/rubygems/rubygems/pull/6133) by deivid-rodriguez
* Installs bundler 2.4.0 as a default gem.

## Bug fixes:

* Fix crash due to `BundlerVersionFinder` not defined. Pull request [#6152] (https://github.com/rubygems/rubygems/pull/6152) by deivid-rodriguez
* Don't leave corrupted partial package download around when running out of disk space. Pull request [#5681] (https://github.com/rubygems/rubygems/pull/5681) by duckinator

-----

RubyGems installed the following executables:
  /root/.rbenv/versions/3.1.3/bin/gem
  /root/.rbenv/versions/3.1.3/bin/bundle
  /root/.rbenv/versions/3.1.3/bin/bundler

Ruby Interactive (ri) documentation was installed. ri is kind of like man
pages for Ruby libraries. You may access it like this:
  ri Classname
  ri Classname.class_method
  ri Classname#instance_method
If you do not wish to install this documentation in the future, use the
--no-document flag, or set it as the default in your ~/.gemrc file. See
'gem help env' for details.

RubyGems system software updated
root@debian:~# gem update --system_
```

INSTALACIÓN DE NGINX

Ahora instalaremos Nginx que nos sirve para servir contenidos web y aplicaciones usando el siguiente comando:

```
apt install nginx
```

```
Seleccionando el paquete libnginx-mod-http-xslt-filter previamente no seleccionado.
Preparando para desempaquetar .../05-libnginx-mod-http-xslt-filter_1.18.0-6.1+deb11u3_amd64.deb ...
Desempaquetando libnginx-mod-http-xslt-filter (1.18.0-6.1+deb11u3) ...
Seleccionando el paquete libnginx-mod-mail previamente no seleccionado.
Preparando para desempaquetar .../06-libnginx-mod-mail_1.18.0-6.1+deb11u3_amd64.deb ...
Desempaquetando libnginx-mod-mail (1.18.0-6.1+deb11u3) ...
Seleccionando el paquete libnginx-mod-stream previamente no seleccionado.
Preparando para desempaquetar .../07-libnginx-mod-stream_1.18.0-6.1+deb11u3_amd64.deb ...
Desempaquetando libnginx-mod-stream (1.18.0-6.1+deb11u3) ...
Seleccionando el paquete libnginx-mod-stream-geoip previamente no seleccionado.
Preparando para desempaquetar .../08-libnginx-mod-stream-geoip_1.18.0-6.1+deb11u3_amd64.deb ...
Desempaquetando libnginx-mod-stream-geoip (1.18.0-6.1+deb11u3) ...
Seleccionando el paquete nginx-core previamente no seleccionado.
Preparando para desempaquetar .../09-nginx-core_1.18.0-6.1+deb11u3_amd64.deb ...
Desempaquetando nginx-core (1.18.0-6.1+deb11u3) ...
Seleccionando el paquete nginx previamente no seleccionado.
Preparando para desempaquetar .../10-nginx_1.18.0-6.1+deb11u3_all.deb ...
Desempaquetando nginx (1.18.0-6.1+deb11u3) ...
Configurando nginx-common (1.18.0-6.1+deb11u3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Configurando libnginx-mod-http-xslt-filter (1.18.0-6.1+deb11u3) ...
Configurando libgeoip1:amd64 (1.6.12-7) ...
Configurando geoip-database (20191224-3) ...
Configurando libnginx-mod-mail (1.18.0-6.1+deb11u3) ...
Configurando libnginx-mod-http-image-filter (1.18.0-6.1+deb11u3) ...
Configurando libnginx-mod-stream (1.18.0-6.1+deb11u3) ...
Configurando libnginx-mod-stream-geoip (1.18.0-6.1+deb11u3) ...
Configurando libnginx-mod-http-geoip (1.18.0-6.1+deb11u3) ...
Configurando nginx-core (1.18.0-6.1+deb11u3) ...
Upgrading binary: nginx.
Configurando nginx (1.18.0-6.1+deb11u3) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u5) ...
root@debian:~# apt install nginx
```

INSTALACIÓN POSTGRESQL

Haremos el siguiente comando para instalar la base de datos de PostgreSQL:

```
apt install postgresql postgresql-contrib
```

```
creando subdirectorios ... hecho
seleccionando implementación de memoria compartida dinámica ...posix
seleccionando el valor para max_connections ... 100
seleccionando el valor para shared_buffers ... 128MB
seleccionando el huso horario por omisión ... Europe/Madrid
creando archivos de configuración ... hecho
ejecutando script de inicio (bootstrap) ... hecho
realizando inicialización post-bootstrap ... hecho
sincronizando los datos a disco ... hecho

Completado. Ahora puede iniciar el servidor de bases de datos usando:

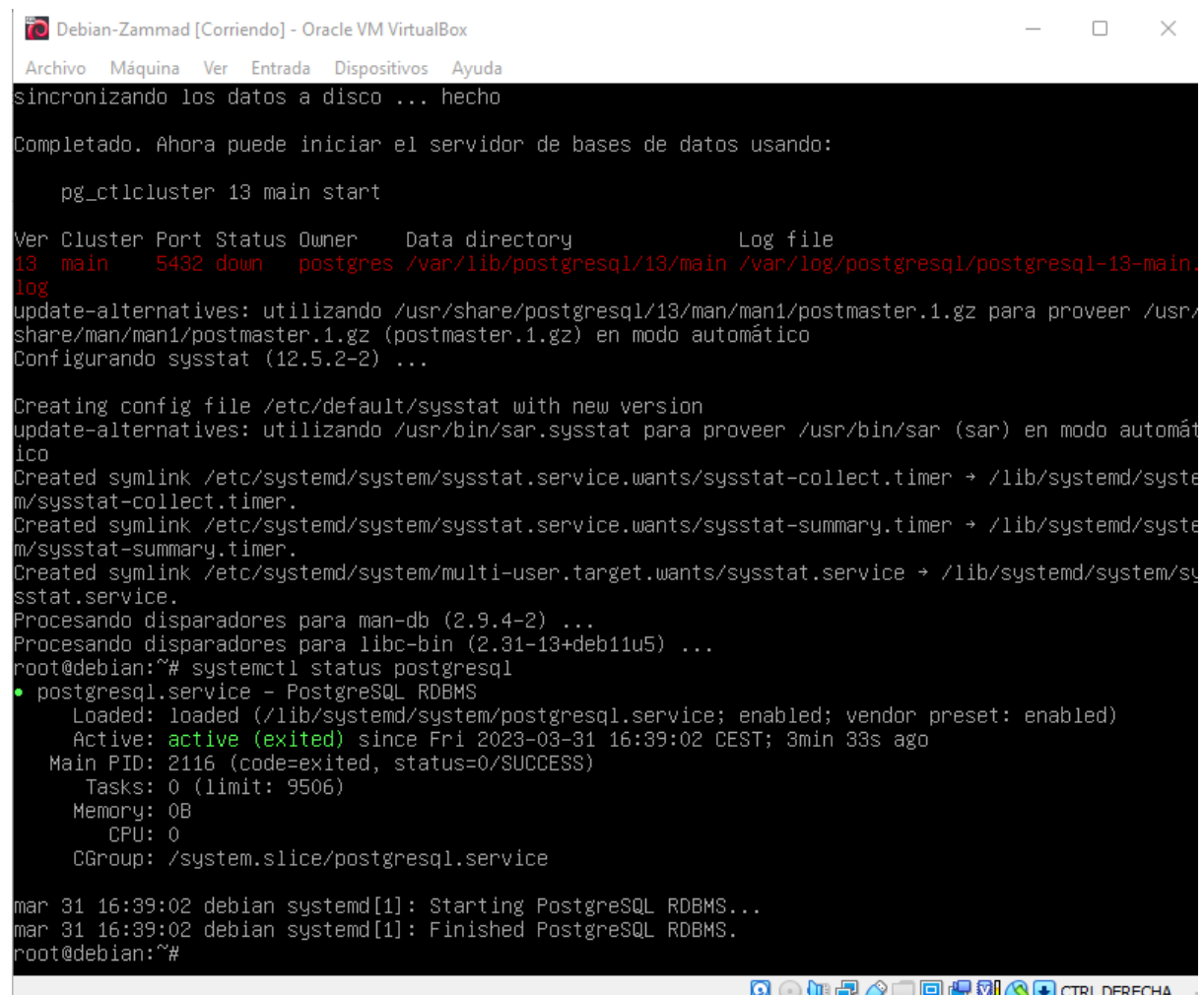
    pg_ctlcluster 13 main start

Ver Cluster Port Status Owner    Data directory          Log file
13  main    5432  down   postgres /var/lib/postgresql/13/main /var/log/postgresql/postgresql-13-main.log
update-alternatives: utilizando /usr/share/postgresql/13/man/man1/postmaster.1.gz para proveer /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) en modo automático
Configurando postgresql (13+225) ...
Configurando sysstat (12.5.2-2) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: utilizando /usr/bin/sar.sysstat para proveer /usr/bin/sar (sar) en modo automático
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-collect.timer → /lib/systemd/systemd/systemd/sysstat-collect.timer.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-summary.timer → /lib/systemd/systemd/sysstat-summary.timer.
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/systemd/sysstat.service.
Configurando postgresql-contrib (13+225) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u5) ...
root@debian:~# apt install postgresql postgresql-contrib_
```

Aquí podemos usar el siguiente comando para ver el estado de la base de datos recién instalada:

```
Systemctl status postgresql
```



```
Debian-Zammad [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
sincronizando los datos a disco ... hecho
Completado. Ahora puede iniciar el servidor de bases de datos usando:

    pg_ctlcluster 13 main start

Ver Cluster Port Status Owner    Data directory          Log file
13  main    5432 down   postgres /var/lib/postgresql/13/main /var/log/postgresql/postgresql-13-main.log
update-alternatives: utilizando /usr/share/postgresql/13/man/man1/postmaster.1.gz para proveer /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) en modo automático
Configurando sysstat (12.5.2-2) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: utilizando /usr/bin/sar.sysstat para proveer /usr/bin/sar (sar) en modo automático
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-collect.timer → /lib/systemd/system/sysstat-collect.timer.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-summary.timer → /lib/systemd/system/sysstat-summary.timer.
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/system/sysstat.service.
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u5) ...
root@debian:~# systemctl status postgresql
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2023-03-31 16:39:02 CEST; 3min 33s ago
   Main PID: 2116 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 9506)
   Memory: 0B
   CPU: 0
   CGroup: /system.slice/postgresql.service

mar 31 16:39:02 debian systemd[1]: Starting PostgreSQL RDBMS...
mar 31 16:39:02 debian systemd[1]: Finished PostgreSQL RDBMS.
root@debian:~#
```

INSTALACIÓN DE ELASTICSEARCH

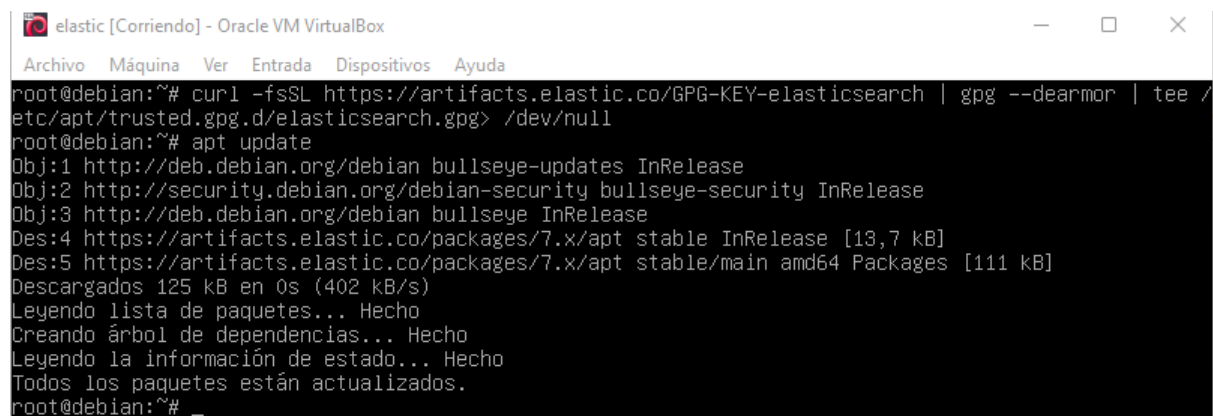
Empezaremos por agregar el repositorio al sistema operativo con el siguiente comando:

```
echo "deb [signed-by=/etc/apt/trusted.gpg.d/elasticsearch.gpg] \  
https://artifacts.elastic.co/packages/7.x/apt stable main" | \  
tee -a /etc/apt/sources.list.d/elastic-7.x.list > /dev/null
```

Continuamos con un comando para descargar e instalar la clave pública de GPG (GNU privacy Guard) de ElasticSearch en el sistema operativo:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | \  
gpg --dearmor | tee /etc/apt/trusted.gpg.d/elasticsearch.gpg> /dev/null
```

Y hacemos un apt update seguido



```
elastic [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@debian:~# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --dearmor | tee /etc/apt/trusted.gpg.d/elasticsearch.gpg> /dev/null
root@debian:~# apt update
Obj:1 http://deb.debian.org/debian bullseye-updates InRelease
Obj:2 http://security.debian.org/debian-security bullseye-security InRelease
Obj:3 http://deb.debian.org/debian bullseye InRelease
Des:4 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13,7 kB]
Des:5 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [111 kB]
Descargados 125 kB en 0s (402 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
root@debian:~# _
```

Y casi terminamos con el siguiente comando:

```
apt install elasticsearch
```

```

Descargados 125 kB en 0s (402 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
root@debian:~# apt install elasticsearch
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  linux-image-5.10.0-18-amd64
Utilice «apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
  elasticsearch
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 317 MB de archivos.
Se utilizarán 530 MB de espacio de disco adicional después de esta operación.
Des:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.10 [317 MB]
Descargados 317 MB en 4s (74,8 MB/s)
Seleccionando el paquete elasticsearch previamente no seleccionado.
(Leyendo la base de datos ... 37866 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../elasticsearch_7.17.10_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Desempaquetando elasticsearch (7.17.10) ...
Configurando elasticsearch (7.17.10) ...
### NOT starting on installation, please execute the following statements to configure elasticsearch
service to start automatically using systemd
  sudo systemctl daemon-reload
  sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
  sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
root@debian:~# apt install elasticsearch

```

Por último instalaremos el plugin ingest-attachment que nos permitirá procesar y extraer contenido de archivos adjuntos en documentos indexados en Elasticsearch.

```
/usr/share/elasticsearch/bin/elasticsearch-plugin install ingest-attachment
```

```
root@debian:~# /usr/share/elasticsearch/bin/elasticsearch-plugin install ingest-attachment
-> Installing ingest-attachment
-> Downloading ingest-attachment from elastic
[=====] 100%
#####
@ WARNING: plugin requires additional permissions @
#####
* java.lang.RuntimePermission accessClassInPackage.sun.java2d.cmm.kcms
* java.lang.RuntimePermission accessDeclaredMembers
* java.lang.RuntimePermission getClassLoader
* java.lang.reflect.ReflectPermission suppressAccessChecks
* java.security.SecurityPermission createAccessControlContext
See https://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.

Continue with installation? [y/N]y
-> Installed ingest-attachment
-> Please restart Elasticsearch to activate any plugins installed
```

Ahora tendremos que reiniciar elasticsearch (systemctl restart elasticsearch) para que esté en funcionamiento, ahora tendremos que instalar Zammad primero para poder terminar de unir ElasticSearch a Zammad.

INSTALACIÓN ZAMMAD

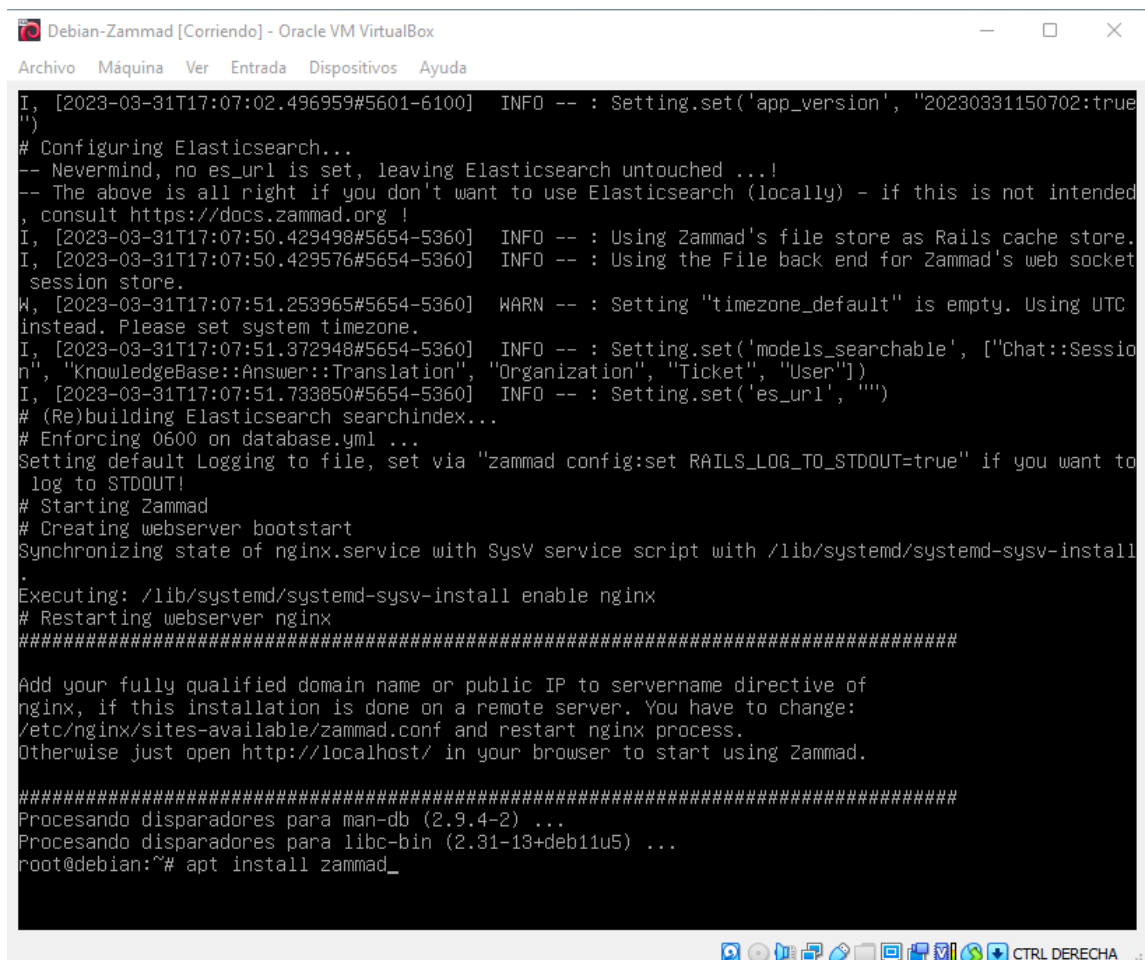
Empezaremos por añadir la llave pública al repositorio con el siguiente comando:

```
curl -fsSL https://dl.packager.io/srv/zammad/zammad/key | \
  gpg --dearmor | tee /etc/apt/trusted.gpg.d/pkgr-zammad.gpg> /dev/null
```

Ahora con el siguiente comando añadiremos al repositorio de paquetes al sistema operativo:

```
echo "deb [signed-by=/etc/apt/trusted.gpg.d/pkgr-zammad.gpg]
https://dl.packager.io/srv/deb/zammad/zammad/stable/debian 11 main" | \
tee /etc/apt/sources.list.d/zammad.list > /dev/null
```

Continuaremos haciendo un apt update y luego un apt install zammad.



```
Debian-Zammad [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

I, [2023-03-31T17:07:02.496959#5601-6100] INFO -- : Setting.set('app_version', "20230331150702:true")
# Configuring Elasticsearch...
-- Nevermind, no es_url is set, leaving Elasticsearch untouched ...!
-- The above is all right if you don't want to use Elasticsearch (locally) - if this is not intended
, consult https://docs.zammad.org !
I, [2023-03-31T17:07:50.429498#5654-5360] INFO -- : Using Zammad's file store as Rails cache store.
I, [2023-03-31T17:07:50.429576#5654-5360] INFO -- : Using the File back end for Zammad's web socket
session store.
W, [2023-03-31T17:07:51.253965#5654-5360] WARN -- : Setting "timezone_default" is empty. Using UTC
instead. Please set system timezone.
I, [2023-03-31T17:07:51.372948#5654-5360] INFO -- : Setting.set('models_searchable', ["Chat::Sessio
n", "KnowledgeBase::Answer::Translation", "Organization", "Ticket", "User"])
I, [2023-03-31T17:07:51.733850#5654-5360] INFO -- : Setting.set('es_url', "")
# (Re)building Elasticsearch searchindex...
# Enforcing 0600 on database.yml ...
Setting default Logging to file, set via "zammad config:set RAILS_LOG_TO_STDOUT=true" if you want to
log to STDOUT!
# Starting Zammad
# Creating webserver bootstart
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install
.
Executing: /lib/systemd/systemd-sysv-install enable nginx
# Restarting webserver nginx
#####

Add your fully qualified domain name or public IP to servername directive of
nginx, if this installation is done on a remote server. You have to change:
/etc/nginx/sites-available/zammad.conf and restart nginx process.
Otherwise just open http://localhost/ in your browser to start using Zammad.

#####
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u5) ...
root@debian:~# apt install zammad_
```


Una vez tengamos terminada la instalación tendremos que copiar el archivo de `zammad_ssl.conf` a la ruta correspondiente, ya que es la configuración necesaria para permitir el acceso a Zammad a través de HTTPS, con el siguiente comando.

```
cp opt/contrib/nginx/zammad_ssl.conf /etc/nginx/sites-available/zammad.conf
```

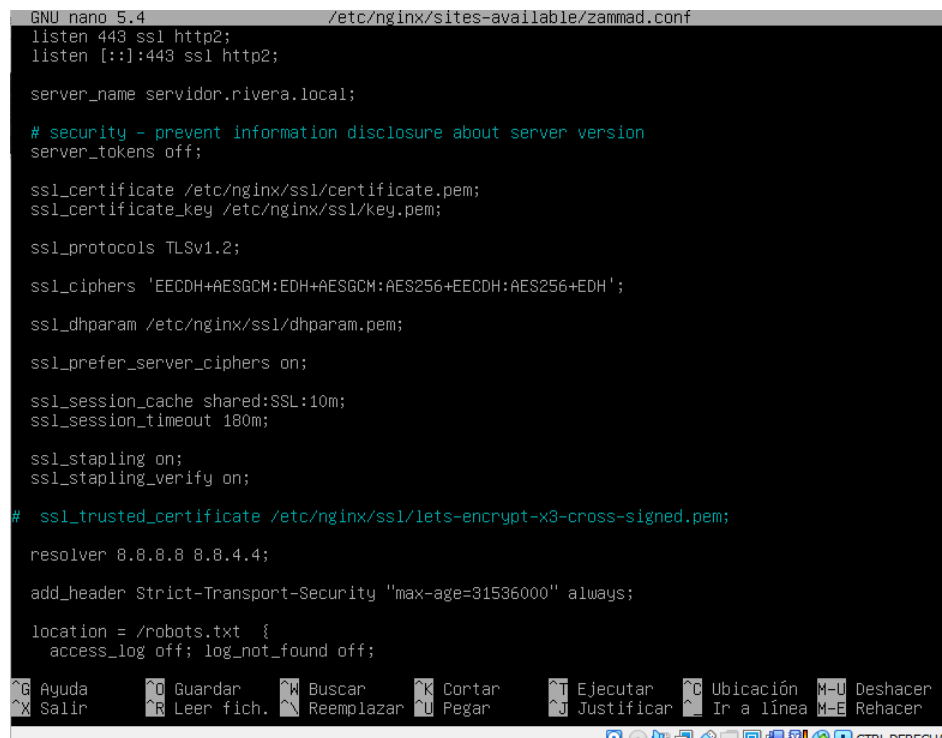
Crearemos ahora un nuevo directorio con el siguiente comando:, que es donde almacenaremos los certificados SSL utilizado para proporcionar conexiones seguras.

```
mkdir /etc/nginx/ssl
```

Ahora generaremos unas claves RSA y un certificado autofirmado x.509. este certificado nos permitirá hacer pruebas para un entorno de desarrollo o pruebas, para un entorno de producción tendríamos que obtener un certificado válido emitido por una entidad certificadora.

Luego en el mismo comando le especificamos en dónde se almacenará la clave privada.

```
openssl req -newkey rsa:4096 -nodes -x509 -days 1825 -keyout  
/etc/nginx/ssl/key.pem -out /etc/nginx/ssl/certificate.pem  
openssl dhparam -out /etc/nginx/ssl/dhparam.pem 409
```



```
GNU nano 5.4 /etc/nginx/sites-available/zammad.conf
listen 443 ssl http2;
listen [::]:443 ssl http2;

server_name servidor.rivera.local;

# security - prevent information disclosure about server version
server_tokens off;

ssl_certificate /etc/nginx/ssl/certificate.pem;
ssl_certificate_key /etc/nginx/ssl/key.pem;

ssl_protocols TLSv1.2;

ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';

ssl_dhparam /etc/nginx/ssl/dhparam.pem;

ssl_prefer_server_ciphers on;

ssl_session_cache shared:SSL:10m;
ssl_session_timeout 180m;

ssl_stapling on;
ssl_stapling_verify on;

# ssl_trusted_certificate /etc/nginx/ssl/lets-encrypt-x3-cross-signed.pem;

resolver 8.8.8.8 8.8.4.4;

add_header Strict-Transport-Security "max-age=31536000" always;

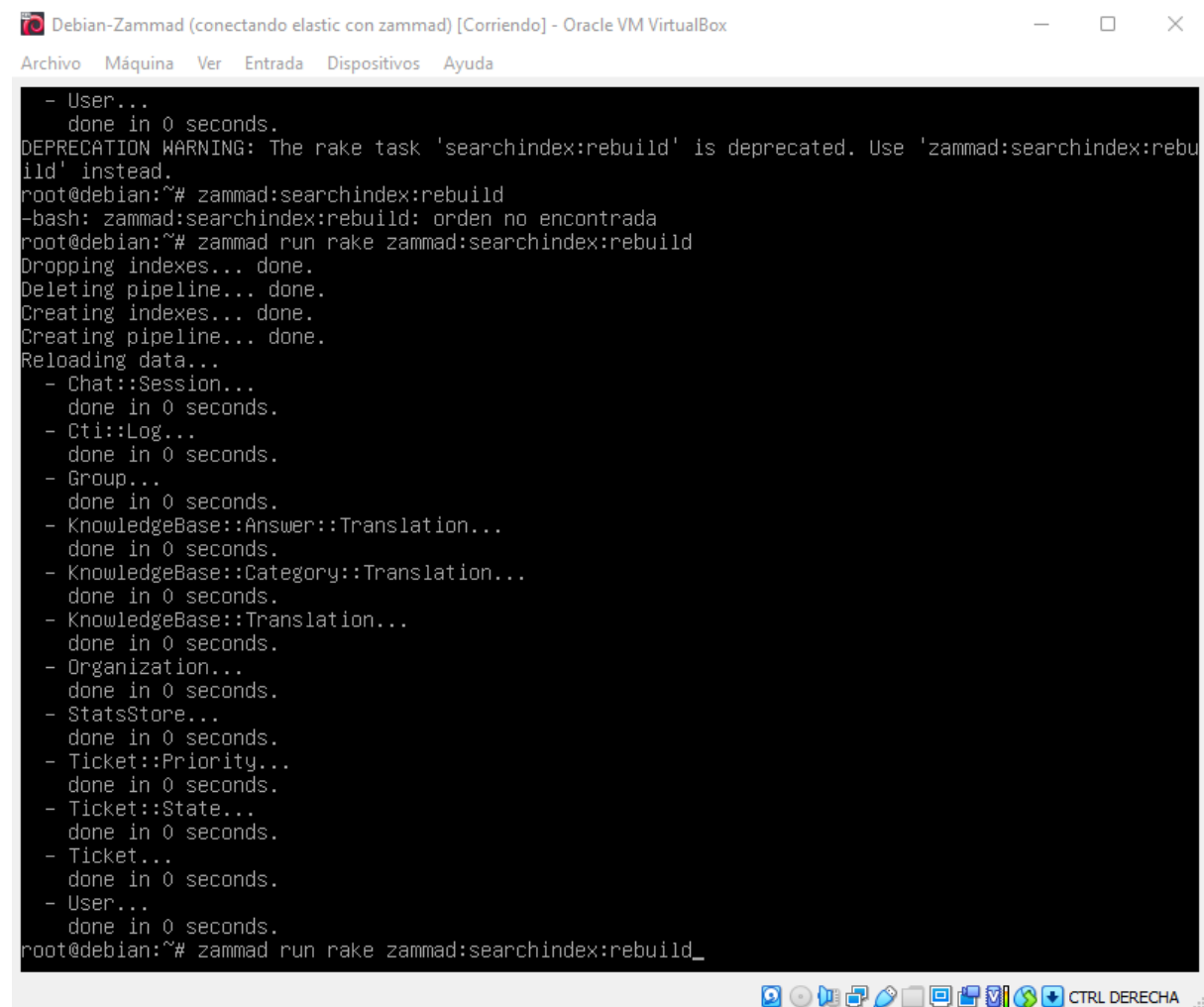
location = /robots.txt {
    access_log off; log_not_found off;

^G Ayuda      ^O Guardar   ^W Buscar    ^K Cortar    ^T Ejecutar  ^C Ubicación  M-U Deshacer
^X Salir      ^R Leer fich. ^E Reemplazar ^U Pegar      ^J Justificar ^I Ir a línea  M-E Rehacer
CTRI DREFCHA
```

Por último uniremos elasticsearch a zammad con los siguientes comandos para tener todo list para empezar a trabajar con zammad.

```
zammad run rails r "Setting.set('es_url', 'http://localhost:9200')"
```

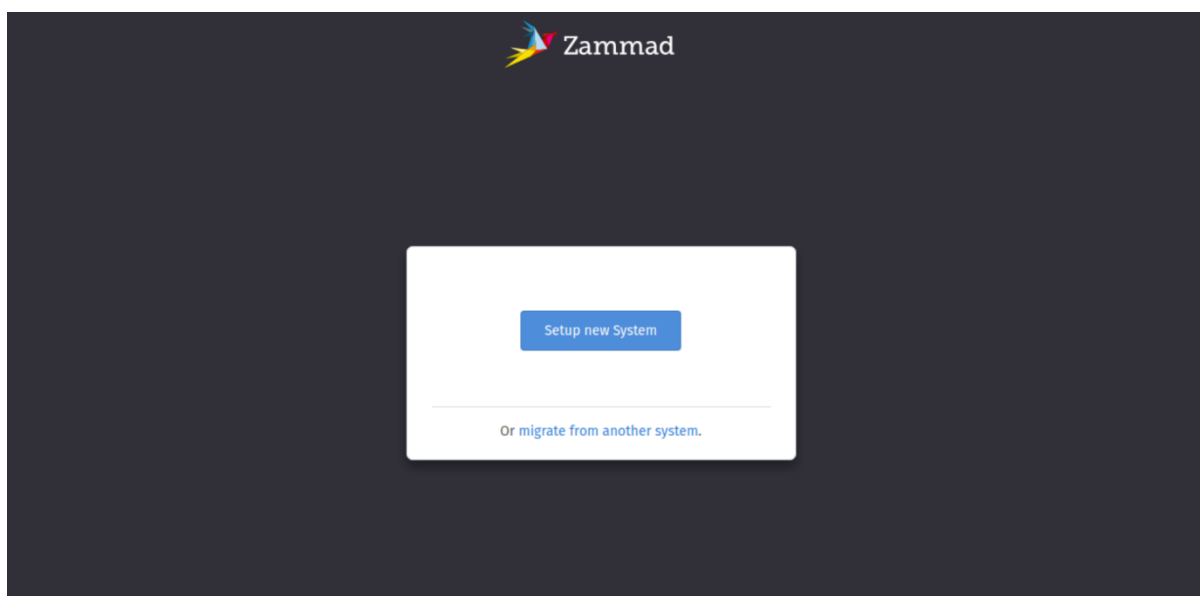
```
zammad run rake zammad:searchindex:rebuild
```




```
Debian-Zammad (conectando elastic con zammad) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
- User...
  done in 0 seconds.
DEPRECATION WARNING: The rake task 'searchindex:rebuild' is deprecated. Use 'zammad:searchindex:rebuild' instead.
root@debian:~# zammad:searchindex:rebuild
-bash: zammad:searchindex:rebuild: orden no encontrada
root@debian:~# zammad run rake zammad:searchindex:rebuild
Dropping indexes... done.
Deleting pipeline... done.
Creating indexes... done.
Creating pipeline... done.
Reloading data...
- Chat::Session...
  done in 0 seconds.
- Cti::Log...
  done in 0 seconds.
- Group...
  done in 0 seconds.
- KnowledgeBase::Answer::Translation...
  done in 0 seconds.
- KnowledgeBase::Category::Translation...
  done in 0 seconds.
- KnowledgeBase::Translation...
  done in 0 seconds.
- Organization...
  done in 0 seconds.
- StatsStore...
  done in 0 seconds.
- Ticket::Priority...
  done in 0 seconds.
- Ticket::State...
  done in 0 seconds.
- Ticket...
  done in 0 seconds.
- User...
  done in 0 seconds.
root@debian:~# zammad run rake zammad:searchindex:rebuild_
```

CONFIGURANDO ZAMMAD

Para poder configurar el sistema de tickets Zammad, tendremos que entrar a través de la IP del servidor, en este <https://192.25.0.10> y lo primero que nos saldrá una vista para configurar el sistema:



Hacemos clic en el botón de “configurar nuevo sistema” y seremos redirigidos a otra página donde deberemos proporcionar el nombre de administrador, correo electrónico, contraseña y hacer clic en el botón Create.

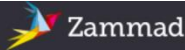


Administrator Account

FIRSTNAME *	LASTNAME *
<input type="text" value="hitesh"/>	<input type="text" value="jethva"/>
EMAIL *	PASSWORD *
<input type="text" value="admin@example.com"/>	<input type="password" value="*****"/>
PASSWORD (CONFIRM) *	
<input type="password" value="*****"/>	

[Go Back](#) [Create](#)


En la siguiente imagen veremos como hay que proporcionar el nombre de tu organización, la URL del sistema, cargar el logotipo.



Organization

ORGANIZATION NAME

LOGO



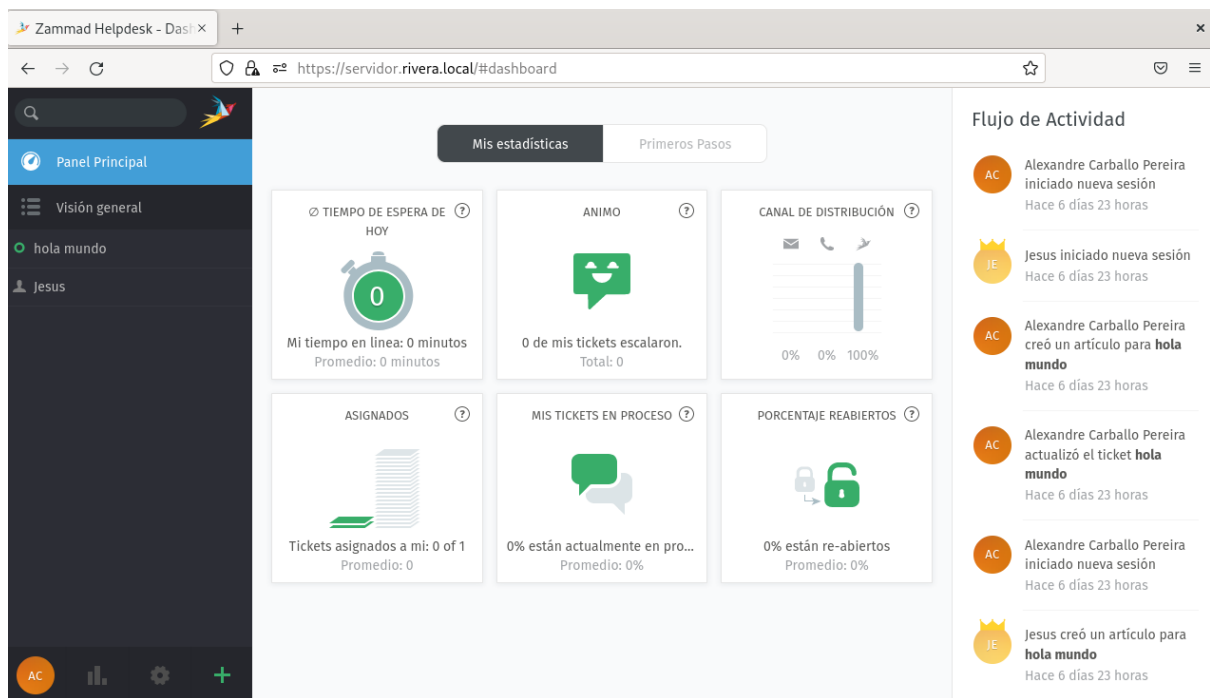
[Upload](#)

SYSTEM URL

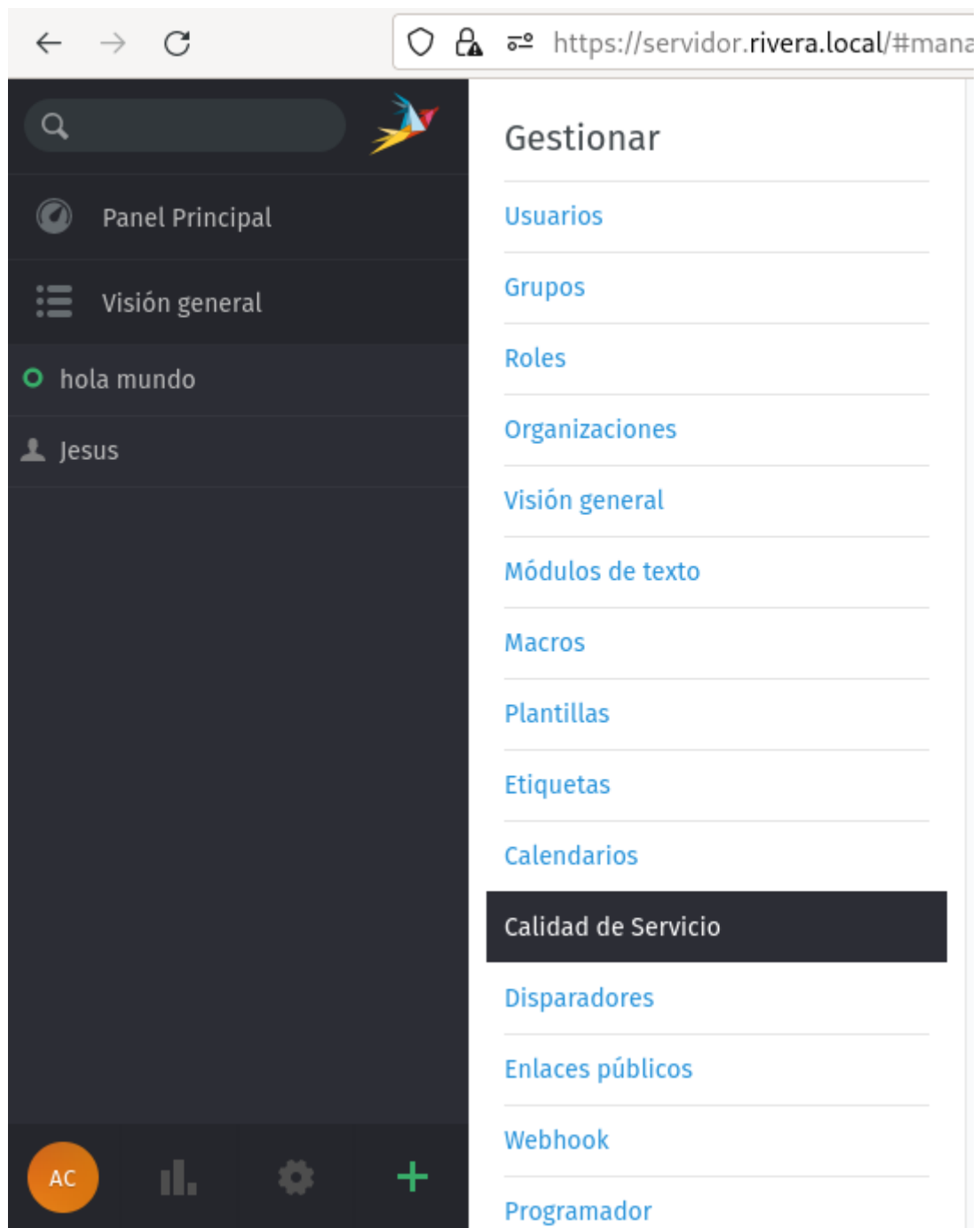
The URL to this installation of Zammad.

[Next](#)

Por último vendría la configuración del correo y podríamos empezar a usar Zammad.



En la pestaña de Gestion, podemos configurar nuevos usuarios, crear grupos de trabajo (admin,recursos humanos, dirección,etc..), roles (admin, trabajador, cliente) para repartir los tickets por secciones, darle prioridad a clientes VIP o a las necesidades de los mismos, también nos permitirá crear plantillas predefinidas para rellenar tickets con valores predefinidos, crear módulos de texto contestar de manera rápida y eficaz perdiendo menos tiempo escribiendo respuestas.



Otra parte muy interesante de Zammad es que nos permite crear canales de comunicación con redes sociales para poder recibir tickets a través de chats personalizados, emails, twitter, facebook, telegram....

El panel de control es muy sencillo y permitirá a los usuarios del mismo una utilización bastante intuitiva y sencilla.

Panel Principal

Visión general

hola mundo

Jesus

Welcome to Zammad!

AC

hola mundo

Ticket#64002 - creado Hace 6 días 23 horas

asdada

Hace 6 días 23 horas

recibido, haremos las pertinentes pesquisas

Hace 6 días 23 horas

Capturar nota o seleccionar adjunto...

JE

Ticket

PROPIETARIO

ESTADO *

cerrado

PRIORIDAD *

2 normal

ETIQUETAS

esto esta muy complicado

+ Añadir Etiqueta

LINKS

+ Agregar Link

NOTIFICACIONES

Cancelar suscripción

Mantenerse en la pestaña

Actualizar

BIBLIOGRAFÍA

- <https://www.eduardocollado.com/2018/01/30/podcast-104-zammad-gestion-de-tickets/>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html>
- <https://docs.zammad.org/en/latest/install/elasticsearch.html>
- <https://linuxhint.com/install-postgresql-debian/>
- <https://www.youtube.com/watch?v=oqmPiH0vd6s>
- <https://docs.zammad.org/en/latest/prerequisites/software.html#package-dependencies>
- <https://help.clouding.io/hc/es/articles/4404691800466-C%C3%B3mo-instalar-el-sistema-de-ticketing-Zammad-en-Ubuntu-20-04-LTS>
- <https://docs.zammad.org/en/latest/install/package.html>
- <https://docs.zammad.org/en/latest/install/elasticsearch.html>
- <https://docs.zammad.org/en/latest/prerequisites/hardware.html>
- <https://github.com/rbenv/rbenv.git>