

Comment installer le système de support client Zammad sur Debian12

Zammad est une solution open source de support client/billetterie qui permet de gérer les communications entre divers canaux tels que le courrier électronique, le chat et les médias sociaux.

Dans ce guide, vous apprendrez comment installer le système de support client Zammad avec PostgreSQL et Nginx sur le Serveur Debian 12. Vous intégrerez également Zammad à Elasticsearch et Redis, alors restez à l'écoute pour cela.

Conditions préalables

Avant de continuer, assurez-vous d'être prêt avec les éléments suivants :

- Un serveur Debian 12 avec 4 Go de mémoire ou plus.
- Un utilisateur non root avec des privilèges d'administrateur.
- Un nom de domaine pointait vers une adresse IP de serveur.

Installation des dépendances

La première étape consiste à installer les dépendances du package pour le système de support client Zammad. Cela comprend les éléments suivants paquets:

- Recherche élastique
- Serveur de base de données PostgreSQL
- Rédis
- Serveur Web Nginx
- Noeud.js
- UFW

Ces packages que vous installerez sur votre serveur Debian.

Tout d'abord, exécutez la commande suivante pour installer les packages de base permettant de gérer le référentiel. Tapez y pour confirmer le installation.

```
sudo apt install curl apt-transport-https gnupg
```

```
root@bookworm64:~#  
root@bookworm64:~# sudo apt install curl apt-transport-https gnupg  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
curl is already the newest version (7.88.1-10+deb12u5).  
gnupg is already the newest version (2.2.40-1.1).  
gnupg set to manually installed.  
The following NEW packages will be installed:  
  apt-transport-https  
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.  
Need to get 25.2 kB of archives.  
After this operation, 35.8 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

Exécutez maintenant la commande ci-dessous pour ajouter le référentiel Elasticsearch 7.x à votre serveur Debian 12.

```
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
```

Ajoutez ensuite le référentiel Elasticsearch de clé GPG à l'aide de la commande ci-dessous.

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

Ensuite, mettez à jour votre référentiel Debian à l'aide de la commande suivante.

```
sudo apt update
```

```

root@bookworm64:~#
root@bookworm64:~# 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
root@bookworm64:~#
root@bookworm64:~# curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | \
gpg --dearmor | tee /etc/apt/trusted.gpg.d/elasticsearch.gpg > /dev/null
root@bookworm64:~#
root@bookworm64:~# sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Get:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.6 kB]
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://security.debian.org/debian-security bookworm-security InRelease
Get:5 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [122 kB]
0% [Working]

```

Une fois le référentiel mis à jour, installez les dépendances pour Zammand à l'aide de la commande suivante. Saisissez Y pour confirmer l'installation.

```
sudo apt install libimlib2 elasticsearch postgresql postgresql-common redis-server nginx nodejs locales ufw
```

```

root@bookworm64:~#
root@bookworm64:~# sudo apt install libimlib2 elasticsearch postgresql postgresql-common redis-server nginx nodejs locales ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
locales is already the newest version (2.36-9+deb12u3).
The following additional packages will be installed:
fontconfig fonts-droid-fallback fonts-noto-mono fonts-urw-base35 iptables libavahi-client3 libavahi-common-data libavahi-common3 libc-ares2
libcairo-gobject2 libcairo2 libcommon-sense-perl libcups2 libdatrie1 libfontenc1 libfribidi0 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libglib2.0-0 libglib2.0-data libgraphite2-3 libgs-common libgs10 libgs10-common libharfbuzz0b libhwy1 libice6
libid3tag0 libidn12 libijs-0.35 libip6tc2 libjbig2dec0 libjson-perl libjson-xs-perl libjxl0.7 liblcms2-2 libllvm14 liblzf1
libnetfilter-contrack3 libnfnetlink0 libnode108 libopenjp2-7 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpaper-utils libpaper1
libpixman-1-0 libpq5 librsync2-2 librsync2-common libsm6 libspectre1 libthai-data libthai0 libtypes-serialiser-perl libwebpdemux2 libx11-xcb1
libxcb-render0 libxcb-shm0 libxrender1 libxslt1.1 libxt6 libz3-4 nginx-common node-acorn node-busboy node-cjs-module-lexer node-undici node-xtend
nodejs-doc poppler-data postgresql-15 postgresql-client-15 postgresql-client-common redis-tools shared-mime-info x11-common xdg-user-dirs
xfonts-encodings xfonts-utils
Suggested packages:
fonts-noto fonts-freefont-otf | fonts-freefont-ttf fonts-texgyre firewalld cups-common low-memory-monitor liblcms2-utils librsync2-bin fcgiwrap
nginx-doc npm poppler-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic
fonts-archic-ukai fonts-archic-uming fonts-nanum postgresql-doc postgresql-doc-15 ruby-redis rsyslog
The following NEW packages will be installed:
elasticsearch fontconfig fonts-droid-fallback fonts-noto-mono fonts-urw-base35 iptables libavahi-client3 libavahi-common-data libavahi-common3
libc-ares2 libcairo-gobject2 libcairo2 libcommon-sense-perl libcups2 libdatrie1 libfontenc1 libfribidi0 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libglib2.0-0 libglib2.0-data libgraphite2-3 libgs-common libgs10 libgs10-common libharfbuzz0b libhwy1 libice6
libid3tag0 libidn12 libijs-0.35 libimlib2 libip6tc2 libjbig2dec0 libjson-perl libjson-xs-perl libjxl0.7 liblcms2-2 libllvm14 liblzf1
libnetfilter-contrack3 libnfnetlink0 libnode108 libopenjp2-7 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpaper-utils libpaper1
libpixman-1-0 libpq5 librsync2-2 librsync2-common libsm6 libspectre1 libthai-data libthai0 libtypes-serialiser-perl libwebpdemux2 libx11-xcb1
libxcb-render0 libxcb-shm0 libxrender1 libxslt1.1 libxt6 libz3-4 nginx nginx-common node-acorn node-busboy node-cjs-module-lexer node-undici
node-xtend nodejs-doc poppler-data poppler-data postgresql postgresql-15 postgresql-client-15 postgresql-client-common postgresql-common redis-server
redis-tools shared-mime-info ufw x11-common xdg-user-dirs xfonts-encodings xfonts-utils
0 upgraded, 91 newly installed, 0 to remove and 7 not upgraded.
Need to get 422 MB of archives.
After this operation, 937 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

Une fois que tout est installé, vérifiez chacun des services pour vous assurer que le service est en cours d'exécution et activé.

Vérifiez le serveur PostgreSQL à l'aide de la commande.

```
sudo systemctl is-enabled postgresql
sudo systemctl status postgresql
```

Ci-dessous, vous pouvez voir que PostgreSQL est en cours d'exécution et activé.

```

root@bookworm64:~#
root@bookworm64:~# sudo systemctl is-enabled postgresql
enabled
root@bookworm64:~# sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since
   Main PID: 6234 (code=exited, status=0/SUCCESS)
   CPU: 8ms

```

Vérifiez le serveur Redis à l'aide de la commande suivante.

```
sudo systemctl is-enabled redis
sudo systemctl status redis
```

Vous pouvez voir ci-dessous que Redis est en cours d'exécution.

```

root@bookworm64:~#
root@bookworm64:~# sudo systemctl is-enabled redis
alias
root@bookworm64:~# sudo systemctl status redis
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; preset: enabled)
   Active: active (running) since
   Docs: http://redis.io/documentation,
   man:redis-server(1)
   Main PID: 6920 (redis-server)
   Status: "Ready to accept connections"
   Tasks: 5 (limit: 4645)

```

Vérifiez maintenant Nginx à l'aide de la commande ci-dessous.

```
sudo systemctl is-enabled nginx
sudo systemctl status nginx
```

Le serveur Web Nginx doit être exécuté et activé comme ceci :

```
root@bookworm64:~#
root@bookworm64:~# sudo systemctl is-enabled nginx
enabled
root@bookworm64:~# sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since
     Docs: man:nginx(8)
  Main PID: 5893 (nginx)
    Tasks: 3 (limit: 4645)
   Memory: 2.3M
      CPU: 102ms
```

Enfin, exécutez la commande ci-dessous pour démarrer et activer Elasticsearch.

```
sudo systemctl start elasticsearch
sudo systemctl enable elasticsearch
```

Vérifiez-le ensuite à l'aide de la commande ci-dessous. Vous devriez voir qu'Elasticsearch est en cours d'exécution et activé.

```
sudo systemctl status elasticsearch
```

```
root@bookworm64:~#
root@bookworm64:~# sudo systemctl start elasticsearch
root@bookworm64:~# sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/syst
root@bookworm64:~#
root@bookworm64:~# sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
   Active: active (running) since
     Docs: https://www.elastic.co
  Main PID: 6980 (java)
    Tasks: 63 (limit: 4645)
   Memory: 2.2G
      CPU: 1min 17.771s
   CGroup: /system.slice/elasticsearch.service
           └─6980 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=
             └─7162 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
```

Configuration des dépendances

Après avoir installé les dépendances du package pour Zammad, l'étape suivante consiste à configurer chacune des dépendances pour Zammad. Vous configurerez les éléments suivants :

- PostgreSQL : augmentez la valeur `max_connections` par défaut à 2 000.
- Elasticsearch : installation de la pièce jointe d'ingestion du plugin nécessaire à Zammad.
- Configurez les paramètres régionaux par défaut
- sur `en_US.UTF-8` UFW : démarrez et activez, ouvrez les ports SSH, HTTP et HTTPS.

Configurer PostgreSQL

Ouvrez la configuration PostgreSQL par défaut ci-dessous. `/etc/postgresql/15/main/postgresql.conf` en utilisant la commande de l'éditeur nano

```
sudo nano /etc/postgresql/15/main/postgresql.conf
```

Modifiez le `max_connections` par défaut dans PostgreSQL à 2000 m, ce qui est requis par Zammad.

```
max_connections = 2000
```

Enregistrez le fichier et quittez l'éditeur.

Exécutez maintenant la commande `systemctl` suivante pour redémarrer PostgreSQL et appliquer les modifications,


```
locale-gen en_US.UTF-8
```

Configurez ensuite les paramètres régionaux par défaut sur en_US.UTF-8 en ajoutant une nouvelle configuration du dépôt /etc/default/locale

```
echo "LANG=en_US.UTF-8" > /etc/default/locale
```

```
root@bookworm64:~#  
root@bookworm64:~# locale-gen en_US.UTF-8  
Generating locales (this might take a while)...  
  en_US.UTF-8... done  
Generation complete.  
root@bookworm64:~#  
root@bookworm64:~# echo "LANG=en_US.UTF-8" > /etc/default/locale  
root@bookworm64:~#
```

Configurer UFW

Exécutez maintenant la commande ufw ci-dessous pour activer les profils OpenSSH et WWW Full.

```
sudo ufw allow OpenSSH  
sudo ufw allow "WWW Full"
```

Activez ensuite UFW à l'aide de la commande ci-dessous. tapez y pour confirmer et démarrer UFW.

```
sudo ufw enable
```

Une fois activé, vous obtiendrez ce qui suit :

```
root@bookworm64:~#  
root@bookworm64:~# sudo ufw allow OpenSSH  
Rules updated  
Rules updated (v6)  
root@bookworm64:~#  
root@bookworm64:~# sudo ufw allow "WWW Full"  
Rules updated  
Rules updated (v6)  
root@bookworm64:~#  
root@bookworm64:~# sudo ufw enable  
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y  
Firewall is active and enabled on system startup  
root@bookworm64:~#
```

Avec UFW activé, exécutez la commande ci-dessous pour le vérifier.

```
sudo ufw status
```

Vous pouvez voir UFW avec le statut Actif et les profils activés OpenSSH et WWW Full.

```
root@bookworm64:~#  
root@bookworm64:~# sudo ufw status  
Status: active  
  
To Action From  
-- --  
OpenSSH ALLOW Anywhere  
WWW Full ALLOW Anywhere  
OpenSSH (v6) ALLOW Anywhere (v6)  
WWW Full (v6) ALLOW Anywhere (v6)
```

Installation du système de support client Zammad

À ce stade, vous avez terminé l'installation et la configuration des dépendances pour Zammad. Maintenant tu vas commencer l'installation de Zammad, qui peut être effectuée en utilisant plusieurs méthodes. Le moyen le plus simple consiste à utiliser les packages du site officiel Dépôt Zammad.

Exécutez la commande ci-dessous pour ajouter la clé GPG pour le référentiel Zammad.

```
curl -fsSL https://dl.packager.io/srv/zammad/zammad/key | \
```

```
gpg --dearmor | tee /etc/apt/trusted.gpg.d/pkgr-zammad.gpg> /dev/null
```

Ajoutez maintenant le référentiel Zammad pour Debian 12 à l'aide de la commande suivante.

```
echo "deb [signed-by=/etc/apt/trusted.gpg.d/pkgr-zammad.gpg] https://dl.packager.io/srv/deb/zammad/zammad/stable/debian 12 main" | \ tee /etc/apt/sources.list.d/zammad.list > /dev/null
```

Mettez à jour et actualisez votre référentiel Debian à l'aide de la commande suivante.

```
sudo apt update
```

```
root@bookworm64:~#  
root@bookworm64:~# curl -fsSL https://dl.packager.io/srv/zammad/zammad/key | \  
gpg --dearmor | tee /etc/apt/trusted.gpg.d/pkgr-zammad.gpg> /dev/null  
root@bookworm64:~#  
root@bookworm64:~# echo "deb [signed-by=/etc/apt/trusted.gpg.d/pkgr-zammad.gpg] https://dl.packager.io/srv/deb/zammad/zammad/stable/debian 12 main" | \  
tee /etc/apt/sources.list.d/zammad.list > /dev/null  
root@bookworm64:~#  
root@bookworm64:~# sudo apt update  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Hit:4 https://artifacts.elastic.co/packages/7.x/apt stable InRelease  
Get:5 https://dl.packager.io/srv/deb/zammad/zammad/stable/debian 12 InRelease [1,845 B]  
26% [Working]
```

Une fois le référentiel mis à jour, installez Zammad avec la commande ci-dessous.

```
sudo apt install zammad
```

Ci-dessous l'installation de Zammad :

```
root@bookworm64:~#  
root@bookworm64:~# sudo apt install zammad  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  zammad  
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.  
Need to get 162 MB of archives.  
After this operation, 568 MB of additional disk space will be used.  
Get:1 https://dl.packager.io/srv/deb/zammad/zammad/stable/debian 12/main amd64 zammad amd64 1.10.0-1 [162 MB]  
0% [1 zammad 770 kB/162 MB 0%]
```

Lors de l'installation, la base de données PostgreSQL et l'utilisateur seront créés, parallèlement à la migration de la base de données.

```
--> done.  
# Enabling Zammad on boot  
# Stopping Zammad  
# Installing zammad on postgresql  
# Creating postgresql bootstart  
Synchronizing state of postgresql.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable postgresql  
# Restarting postgresql server  
# Creating zammad postgresql user  
CREATE ROLE  
# Creating zammad postgresql db  
# Grant privileges to new postgresql user  
GRANT  
# Updating database.yml  
# ... Fixing permission database.yml  
I, [ ] INFO -- : ActionController is using the redis instance at redis://localhost:6379.  
I, [ ] INFO -- : Using Zammad's file store as Rails cache store.  
I, [ ] INFO -- : Using the File back end for Zammad's web socket session store.  
I, [ ] INFO -- : Migrating to CreateBase (20120101000001)  
== 20120101000001 CreateBase: migrating =====  
-- create_table(:sessions, {:id=>:integer})  
-> 0.0518s  
-- add_index(:sessions, :session_id)  
-> 0.0086s  
-- add_index(:sessions, :updated_at)
```

De plus, la configuration du bloc du serveur Nginx sera créée.

```
Setting default logging to file, see via zammad config:set rails_log_level true if you
# Starting Zammad
# Creating webserver bootstart
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-
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.

#####
root@bookworm64:~#
```

Une fois l'installation terminée, exécutez la commande zammad ci-dessous pour configurer l'intégration avec Elasticsearch et reconstruire l'index de recherche dans Elasticsearch.

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

```
root@bookworm64:~#
root@bookworm64:~# zammad run rails r "Setting.set('es_url', 'http://localhost:9200')"
root@bookworm64:~# 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.
```

Ensuite, exécutez la commande ci-dessous pour redémarrer les services zammad. Zammad est livré avec plusieurs services, notamment zammad, zammad-web, zammad-worker et zammad-websocket.

```
sudo systemctl restart zammad
sudo systemctl restart zammad-web
sudo systemctl restart zammad-worker
sudo systemctl restart zammad-websocket
```

Enfin, exécutez la commande ci-dessous pour vérifier l'état du service zammad.

```
sudo systemctl is-enabled zammad
sudo systemctl status zammad
```

Ci-dessous, vous pouvez voir que le service zammad est en cours d'exécution et activé sur le serveur Debian.

```
root@bookworm64:~#
root@bookworm64:~# sudo systemctl is-enabled zammad
enabled
root@bookworm64:~# sudo systemctl status zammad
● zammad.service
   Loaded: loaded (/etc/systemd/system/zammad.service; enabled; preset: enabled)
   Active: active (running) since
 Main PID: 9539 (sleep)
    Tasks: 1 (limit: 4645)
   Memory: 168.0K
      CPU: 5ms
   CGroup: /system.slice/zammad.service
           └─9539 /bin/sleep infinity
```

Configuration du serveur Web Nginx

Par défaut, zammad génère la configuration du bloc du serveur Nginx située dans /etc/nginx/sites-available/zammad.conf. Pour configurer Zammad en production, vous devez vous assurer de modifier la configuration du domaine. De plus, vous devez sécuriser avec HTTPS en production.

Maintenant, ouvrez le fichier /etc/nginx/sites-available/zammad.conf en utilisant la commande suivante de l'éditeur nano.

```
sudo nano /etc/nginx/sites-available/zammad.conf
```

Changez le nom de domaine par défaut dans le nom du serveur option comme celle-ci :

```
server_name help.hwdomain.io;
```

Enregistrez et quittez le fichier.

Ensuite, exécutez la commande ci-dessous pour vérifier la syntaxe Nginx et redémarrez le service Nginx pour prendre effet.

```
sudo nginx -t  
sudo systemctl restart nginx
```

Ci-dessous, vous pouvez voir que le test de syntaxe Nginx est réussi.

```
root@bookworm64:~#  
root@bookworm64:~# sudo nano /etc/nginx/sites-available/zammad.conf  
root@bookworm64:~#  
root@bookworm64:~# sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
root@bookworm64:~#  
root@bookworm64:~# sudo systemctl restart nginx  
root@bookworm64:~#
```

Sécuriser Zammad

Si vous déployez en production, assurez-vous d'activer HTTPS via Letsencrypt.

Installez le plugin certbot et certbot nginx à l'aide de la commande suivante. Saisissez y pour confirmer l'installation.

```
sudo apt install certbot python3-certbot-nginx
```

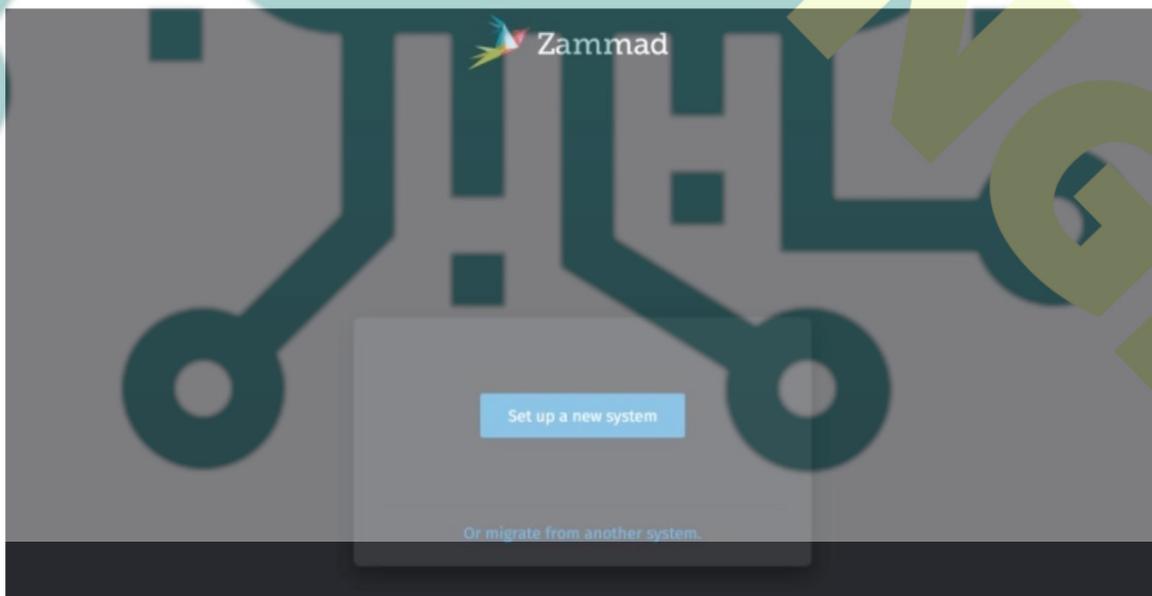
Une fois l'installation terminée, générez des certificats SSL/TLS pour votre nom de domaine Zammad à l'aide du certbot commande ci-dessous. Assurez-vous de modifier les informations du nom de domaine et de l'adresse e-mail dans cette commande.

```
sudo certbot --nginx --agree-tos --redirect --hsts --staple-ocsp --email alice@hwdomain.io -d help.hwdomain.io
```

Le processus de génération sera en cours d'exécution. Une fois terminé, vos certificats SSL seront disponibles sur /etc/letsencrypt/live/domain.com annuaire.

Installation de Zammad

Ouvrez votre navigateur Web et visitez l'URL d'installation de Zammad (<http://help.hwdomain.io>). Cliquez sur Configurer un nouveau système pour démarrez l'installation.



Saisissez maintenant les détails de votre utilisateur administrateur, de votre adresse e-mail et de votre mot de passe. Cliquez ensuite sur Créer pour confirmer.

Administrator Account

FIRST NAME

Alice

LAST NAME

Wonderland

EMAIL

alice@hwdomain.io

PASSWORD *

••••••••

PASSWORD (CONFIRM) *

••••••••

[Go Back](#)

Create

Saisissez le nom de votre organisation et cliquez sur Suivant.

Organization

ORGANIZATION NAME

Alice Inc

LOGO

Upload

SYSTEM URL

https://help.hwdomain.io

The URL to this installation of Zammad.

Next

Pour la notification par e-mail, cliquez sur Ignorer pour l'instant. Vous pourrez configurer cela plus tard.

Email Notification

SEND MAILS VIA *

Local MTA (Sendmail/Postfix/Exim/...) - use serve✓

[Go Back](#)

[Skip](#)

[Continue](#)

Cliquez sur Ignorer pour la configuration du canal de connexion. Cela peut également être fait plus tard, une fois la configuration terminée.

Connect Channels

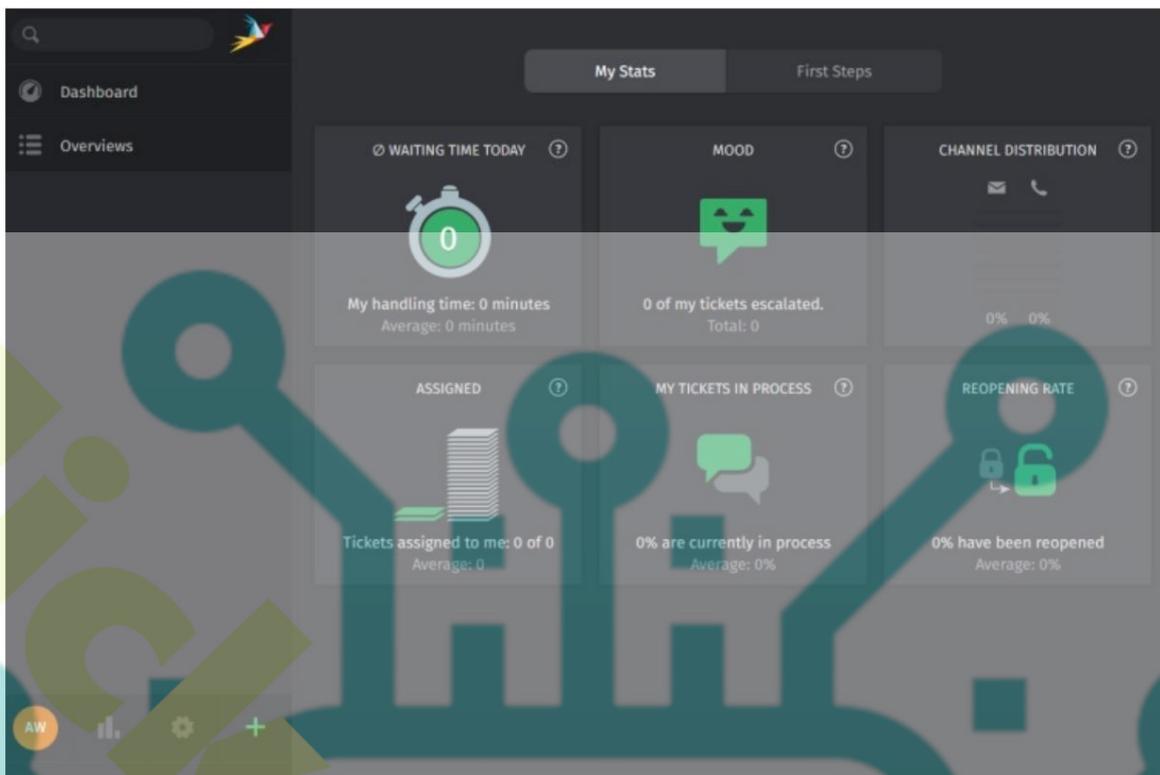
Set up the communication channels you want to use with your Zammad.

[Email](#)

[Go Back](#)

[Skip](#)

Une fois le processus terminé, vous pouvez voir le tableau de bord d'administration de Zammad comme suit :



Conclusion

Toutes nos félicitations! Vous avez maintenant installé avec succès le système de support client Zammad sur le serveur Debian 12. Vous avez installé zammad avec le backend de base de données PostgreSQL, le serveur Web Nginx, Elasticsearch pour le moteur de recherche principal et Redis pour la gestion du cache et des sessions. De plus, vous avez également appris comment activer HTTPS pour Zammad via Certbot et Letsencrypt.
