

BTS Services informatiques aux organisations - SISR**Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin****N° candidat : 02145639104****Contexte de la réalisation professionnelle**

Le service « après vente » de Scani fonctionne uniquement par mail dans une seule et même boîte mail contact qui redirige sur toutes les adresses mail des actifs. Chaque actif a accès à sa boîte et répond de manière non optimisée (cherche si il y a eu une réponse, un début de mail, renvoyer la réponse sur le mail contact pour que tout le monde puisse avoir un aperçu de la conversation mais souvent l'on peut oublier)

Pour palier à ce problème, nous voulons installer un serveur de tickets.

Pour que les personnes qui ne savent pas comment prendre en main OTRS nous avons fait une Documentation sur le site de Netwaze (Reproduite sur le site de Scani) : <https://doc.netwaze.fr/books/la-gestion-de-tickets-otrs>

Intitulé de la réalisation professionnelle**Installation d'un gestionnaire de ticket - OTRS****Période de réalisation : DU 13/07/21 AU 16/07/21****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- Collecter, suivre et orienter des demandes
- Traiter des demandes concernant les services réseau et système, applicatifs
- Traiter des demandes concernant les applications
- Participer à l'évolution d'un site Web exploitant les données de l'organisation.

Conditions de réalisation

- **Ressources présentes** **Pour les besoins de suivi d'incident nous utilisons la boîte mail contact de Scani**
- **Résultats attendus** **La solution que nous allons mettre en place est un gestionnaire de ticket OpenSource : OTRS pour simplifier les échanges entre les actifs et les coopérateurs et d'avoir un vrai suivi.**
- **Durée de réalisation** **2H**

Modalités d'accès à cette réalisation professionnelle.**Site internet : www.netwaze.fr****Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot**

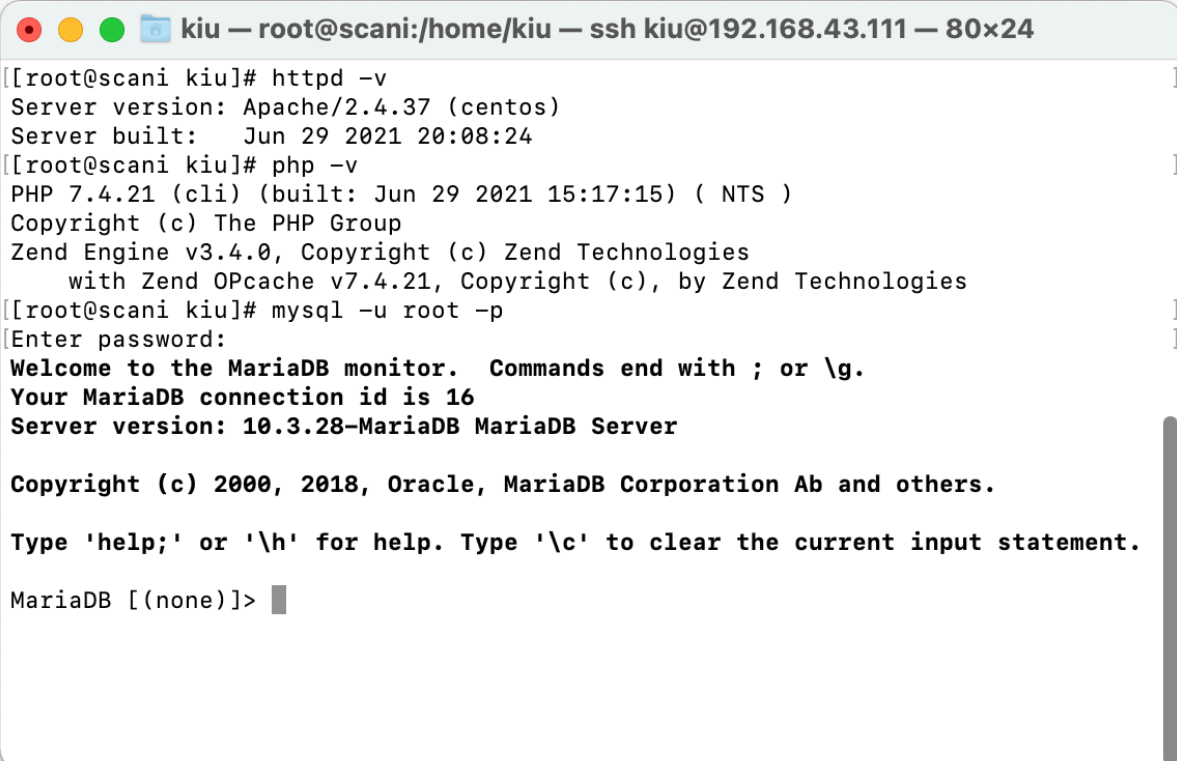
PRÉREQUIS MATÉRIEL

- Un Serveur Linux sous Centos, RockyLinux, AlmaLinux ou RedHat
- Serveur WEB Apache
- MariaDB ou Mysql
- PHP 7.4
- Perl (installation plus bas)

PRÉREQUIS

Tout d'abord nous devons mettre à jour notre distribution Linux : `$ dnf update`
Sachant que le serveur web de Scani est différent de celui de Netwaze, OTRS n'est pas compatible avec NGINX.

Notre serveur web tourne sur un Apache avec un MariaDB et PHP 7.4.21



```
kiu — root@scani:/home/kiu — ssh kiu@192.168.43.111 — 80x24
[[root@scani kiu]# httpd -v
Server version: Apache/2.4.37 (centos)
Server built:   Jun 29 2021 20:08:24
[[root@scani kiu]# php -v
PHP 7.4.21 (cli) (built: Jun 29 2021 15:17:15) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.21, Copyright (c), by Zend Technologies
[[root@scani kiu]# mysql -u root -p
[Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 10.3.28-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Pour n'avoir aucun problème, nous allons désactiver SELINUX.

```
$ nano /etc/selinux/config
```

On recherche la ligne "SELINUX=enforcing" et on la modifie en « SELINUX=disabled"

On redémarre Centos pour appliquer les changements.

```
$ shutdown -r now
```

Pour que OTRS fonctionne bien nous devons modifier le fichier configuration de MariaDB dans /etc/my.cnf.d/mariadb-server.cnf pour rajouter en dessous de [mysqld] :

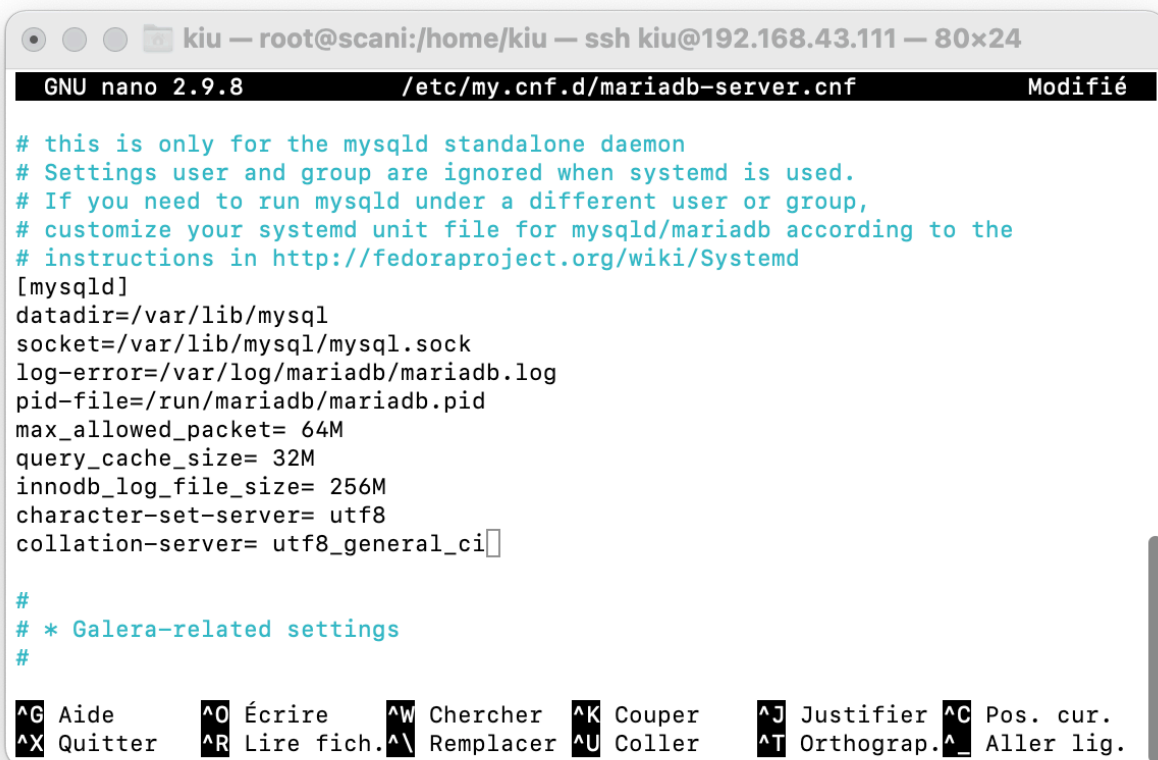
```
max_allowed_packet= 64M
```

```
query_cache_size= 32M
```

```
innodb_log_file_size= 256M
```

```
character-set-server= utf8
```

```
collation-server= utf8_general_ci
```



```
GNU nano 2.9.8 /etc/my.cnf.d/mariadb-server.cnf Modifié

# this is only for the mysqld standalone daemon
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld/mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/run/mariadb/mariadb.pid
max_allowed_packet= 64M
query_cache_size= 32M
innodb_log_file_size= 256M
character-set-server= utf8
collation-server= utf8_general_ci

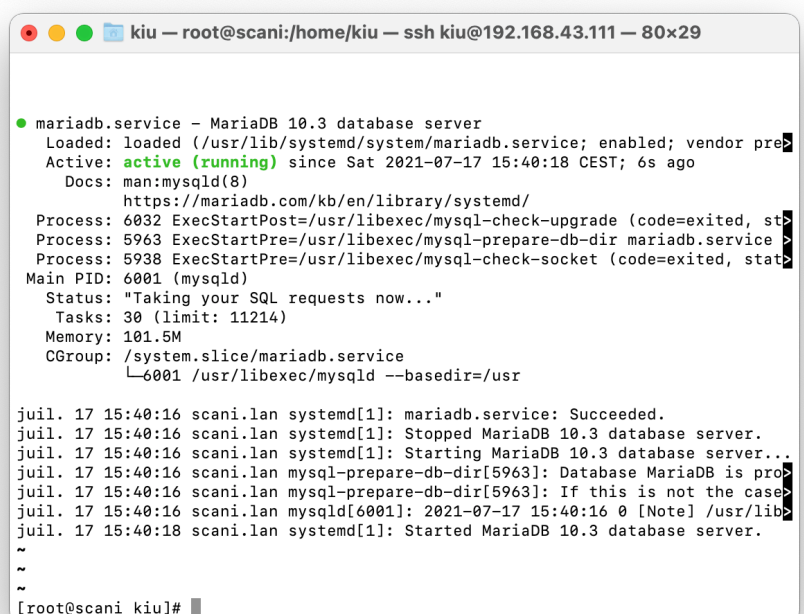
#
# * Galera-related settings
#

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^T Orthograp.^_ Aller lig.
```

On relance notre MariaDB et nous vérifions que le MariaDB fonctionne bien.

```
$ systemctl restart mariadb
```

```
$ systemctl status mariadb
```



```
kiu — root@scani:/home/kiu — ssh kiu@192.168.43.111 — 80x29

● mariadb.service - MariaDB 10.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor pre
   Active: active (running) since Sat 2021-07-17 15:40:18 CEST; 6s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 6032 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, st
   Process: 5963 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mariadb.service >
   Process: 5938 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, stat
   Main PID: 6001 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 30 (limit: 11214)
    Memory: 101.5M
   CGroup: /system.slice/mariadb.service
           └─6001 /usr/libexec/mysql --basedir=/usr

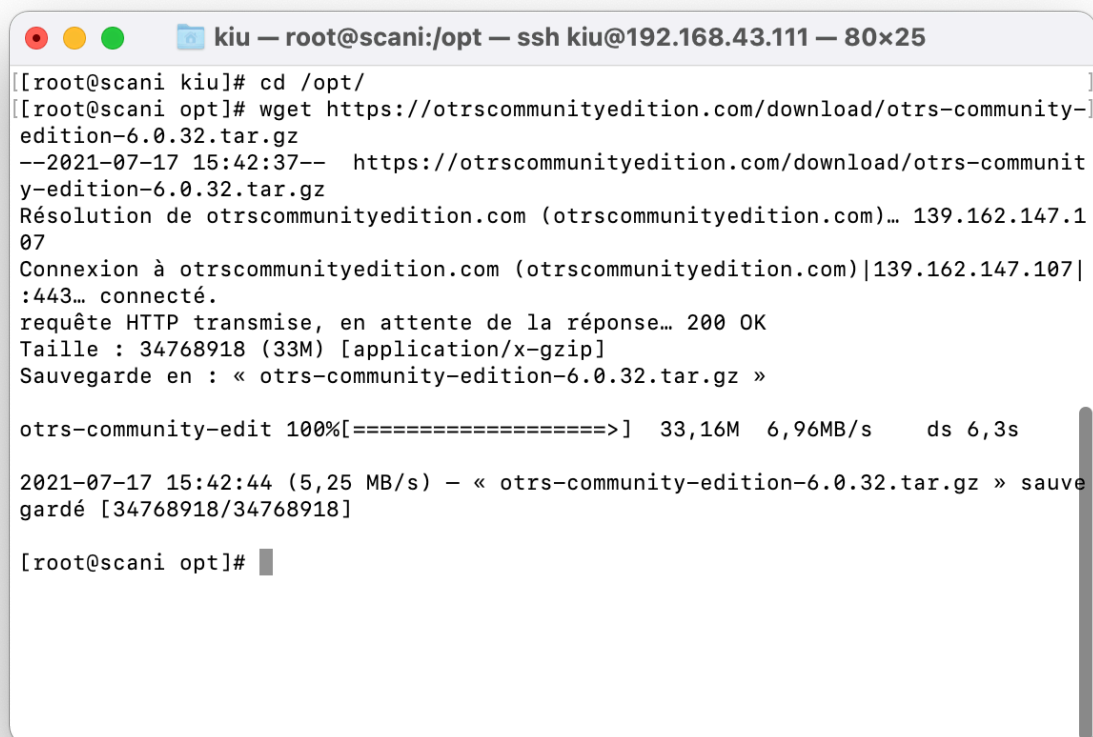
juil. 17 15:40:16 scani.lan systemd[1]: mariadb.service: Succeeded.
juil. 17 15:40:16 scani.lan systemd[1]: Stopped MariaDB 10.3 database server.
juil. 17 15:40:16 scani.lan systemd[1]: Starting MariaDB 10.3 database server...
juil. 17 15:40:16 scani.lan mysql-prepare-db-dir[5963]: Database MariaDB is pro
juil. 17 15:40:16 scani.lan mysql-prepare-db-dir[5963]: If this is not the case
juil. 17 15:40:16 scani.lan mysqld[6001]: 2021-07-17 15:40:16 0 [Note] /usr/lib
juil. 17 15:40:18 scani.lan systemd[1]: Started MariaDB 10.3 database server.
~
~
[root@scani kiu]#
```

INSTALLATION DE OTRS

On télécharge OTRS avec la commande wget dans /opt. Pour installer wget : \$ dnf install wget

\$ cd /opt

\$ wget https://otrscommunityedition.com/download/otrs-community-edition-6.0.32.tar.gz



```
kiu — root@scani:/opt — ssh kiu@192.168.43.111 — 80x25
[[root@scani kiu]# cd /opt/
[[root@scani opt]# wget https://otrscommunityedition.com/download/otrs-community-
edition-6.0.32.tar.gz
--2021-07-17 15:42:37-- https://otrscommunityedition.com/download/otrs-communit
y-edition-6.0.32.tar.gz
Résolution de otrscommunityedition.com (otrscommunityedition.com)... 139.162.147.1
07
Connexion à otrscommunityedition.com (otrscommunityedition.com)|139.162.147.107|
:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 34768918 (33M) [application/x-gzip]
Sauvegarde en : « otrs-community-edition-6.0.32.tar.gz »

otrs-community-edit 100%[=====] 33,16M 6,96MB/s ds 6,3s

2021-07-17 15:42:44 (5,25 MB/s) - « otrs-community-edition-6.0.32.tar.gz » sauve
gardé [34768918/34768918]

[[root@scani opt]# █
```

Puis on le décompresse et on change le nom du dossier.

\$ tar xzf otrs-community-edition-6.0.32.tar.gz

\$ mv otrs-community-edition-6.0.32/ otrs

On supprime l'archive : \$ rm otrs-community-edition-6.0.32.tar.gz

Pour pouvoir configurer OTRS nous devons initialiser le fichier de configuration qui est non reconnu a la base. Donc nous copions le fichier de base et le renommons comme il se doit pour être reconnu par OTRS.

```
$ cp /opt/otrs/Kernel/Config.pm.dist /opt/otrs/Kernel/Config.pm
```

A présent nous devons créer un lien symbolique avec Apache pour que OTRS fonctionne. Pour se faire nous devons taper la commande : `$ sudo ln -s /opt/otrs/scripts/apache2-httpd.include.conf /etc/httpd/conf.d/zzz_otrs.conf`

Pour que OTRS fonctionne nous devons installer le Language Perl et ses dépendances qui se trouvent dans le dépôt EPEL : `$ dnf install epel-release`.

```
$ dnf install mod_epel
```

```
$ dnf install perl-core
```

Pour savoir si OTRS a tout les logiciels et dépendances nécessaires pour son exécution on tape la commande : `$ sudo perl /opt/otrs/bin/otrs.CheckModules.pl`

```
kiu — root@scani:opt — ssh kiu@192.168.43.111 — 80x30
chanism in IMAP connections.)
  o ModPerl::Util.....ok (v2.000011)
  o Moo.....Not installed! To install, you can use: 'yum
install "perl(Moo)". (required - Required for random number generator.)
  o Net::DNS.....Not installed! To install, you can use: 'yum
install "perl(Net::DNS)". (required)
  o Net::LDAP.....Not installed! To install, you can use: 'yum
install "perl(Net::LDAP)". (optional - Required for directory authentication.)
)
  o Net::SMTP.....ok (v3.11)
  o Template.....Not installed! To install, you can use: 'yum
install "perl(Template)". (required - Template::Toolkit, the rendering engine
of OTRS.)
  o Template::Stash::XS.....Not installed! To install, you can use: 'yum
install "perl(Template::Stash::XS)". (required - The fast data stash for Temp
late::Toolkit.)
  o Text::CSV_XS.....Not installed! To install, you can use: 'yum
install "perl(Text::CSV_XS)". (optional - Recommended for faster CSV handling
.)
  o Time::HiRes.....ok (v1.9758)
  o XML::LibXML.....Not installed! To install, you can use: 'yum
install "perl(XML::LibXML)". (required - Required for XML processing.)
  o XML::LibXSLT.....Not installed! To install, you can use: 'yum
install "perl(XML::LibXSLT)". (optional - Required for Generic Interface XSLT
mapping module.)
  o XML::Parser.....Not installed! To install, you can use: 'yum
install "perl(XML::Parser)". (optional - Recommended for XML processing.)
  o YAML::XS.....Not installed! To install, you can use: 'yum
install "perl(YAML::XS)". (required - Required for fast YAML processing.)
[root@scani opt]#
```

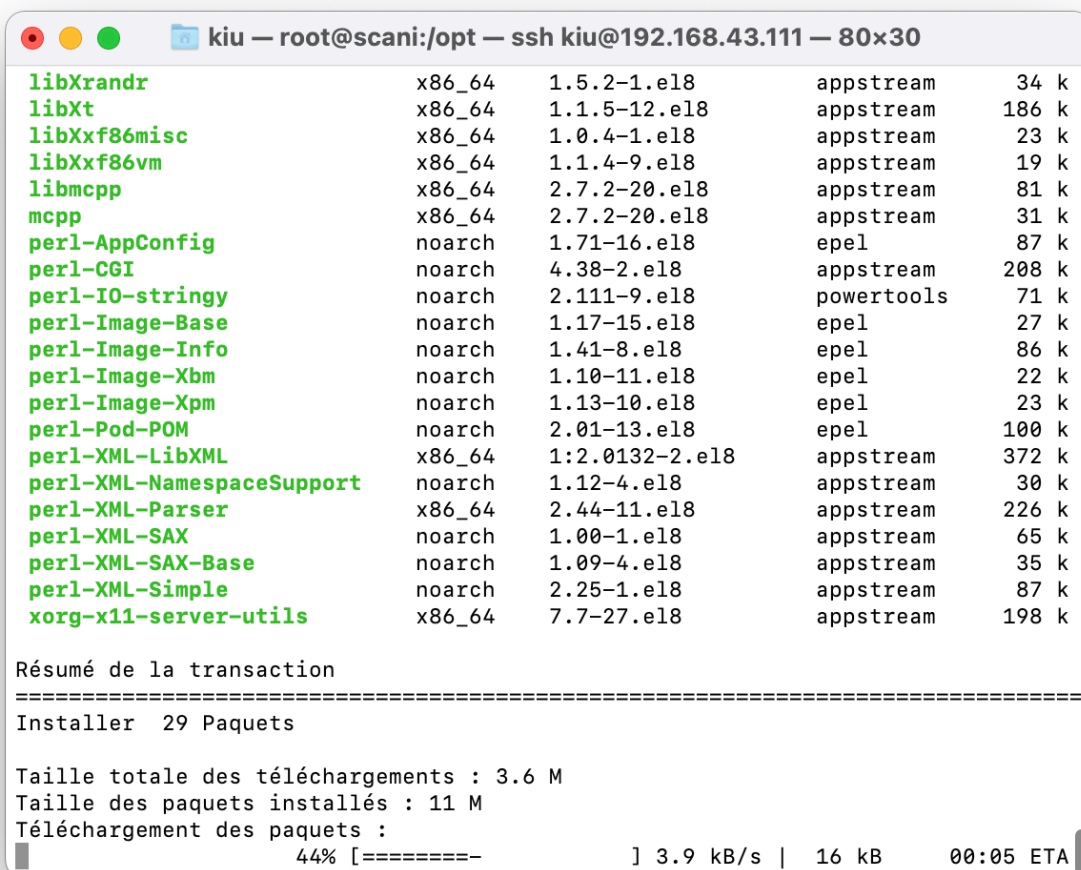
La commande nous retourne alors toutes les dépendances que nous devons installer si nous voulons que OTRS fonctionne avec toutes ces fonctionnalités.

Pour installer toutes les dépendances nous devons obligatoirement le faire manuellement.

On commence par activer les Powertools de Centos. : `dnf config-manager --set-enabled powertools`

Nous pouvons installer une bonne partie des dépendances de OTRS avec dnf :

```
$ dnf install "perl(DateTime)" -y && dnf install "perl(Crypt::Eksblowfish::Bcrypt)" -y && dnf install "perl(Date::Format)" -y && dnf install "perl(DateTime::TimeZone)" -y && dnf install "perl(Encode::HanExtra)" -y && dnf install "perl(JSON::XS)" -y && dnf install "perl(Authen::SASL)" -y && dnf install "perl(Authen::NTLM)" -y && dnf install "perl(Moo)" -y && dnf install "perl(Net::DNS)" -y && dnf install "perl(Net::LDAP)" -y && dnf install "perl(Template)" -y && dnf install "perl(Template::Stash::XS)" -y && dnf install "perl(Text::CSV_XS)" -y && dnf install "perl(XML::LibXML)" -y && dnf install "perl(XML::LibXSLT)" -y && dnf install "perl(XML::Parser)" -y && dnf install "perl(YAML::XS)" -y
```



```
kiu — root@scani:/opt — ssh kiu@192.168.43.111 — 80x30
libXrandr          x86_64      1.5.2-1.e18      appstream        34 k
libXt              x86_64      1.1.5-12.e18     appstream        186 k
libXxf86misc       x86_64      1.0.4-1.e18     appstream        23 k
libXxf86vm         x86_64      1.1.4-9.e18     appstream        19 k
libmcpp            x86_64      2.7.2-20.e18    appstream        81 k
mcpp               x86_64      2.7.2-20.e18    appstream        31 k
perl-AppConfig     noarch      1.71-16.e18     epel             87 k
perl-CGI           noarch      4.38-2.e18      appstream        208 k
perl-IO-stringy   noarch      2.111-9.e18     powertools      71 k
perl-Image-Base    noarch      1.17-15.e18     epel            27 k
perl-Image-Info    noarch      1.41-8.e18      epel            86 k
perl-Image-Xbm     noarch      1.10-11.e18     epel            22 k
perl-Image-Xpm     noarch      1.13-10.e18     epel            23 k
perl-Pod-POM       noarch      2.01-13.e18     epel            100 k
perl-XML-LibXML    x86_64      1:2.0132-2.e18  appstream        372 k
perl-XML-Namespac noarch      1.12-4.e18      appstream        30 k
perl-XML-Parser    x86_64      2.44-11.e18     appstream        226 k
perl-XML-SAX       noarch      1.00-1.e18      appstream        65 k
perl-XML-SAX-Base  noarch      1.09-4.e18      appstream        35 k
perl-XML-Simple    noarch      2.25-1.e18      appstream        87 k
xorg-x11-server-ut x86_64      7.7-27.e18      appstream        198 k

Résumé de la transaction
=====
Installer 29 Paquets

Taille totale des téléchargements : 3.6 M
Taille des paquets installés : 11 M
Téléchargement des paquets :
[=====] 44% [=====] 3.9 kB/s | 16 kB 00:05 ETA
```

On installe ensuite les dépendances que nous ne pouvons pas installer avec dnf. On les installe alors avec Perl. :

```
$ perl -MCPAN -e shell
```

```
[[root@scani opt]# perl -MCPAN -e shell  
Terminal does not support AddHistory.
```

```
cpan shell -- CPAN exploration and modules installation (v2.18)  
Enter 'h' for help.
```

```
cpan[1]> █
```

```
> install Mail::IMAPClient  
> install DateTime::TimeZone  
> exit
```

On créer un utilisateur pour OTRS :

```
$ sudo useradd -d /opt/otrs -m otrs
```

On lui met un mot de passe complexe

```
$ passwd otrs
```

Puis nous associons ce profil au groupe apache

```
$ usermod -G apache otrs
```

Pour finir on lui applique les droits pour que l'utilisateur otrs soit autoriser a gérer OTRS

```
$ perl /opt/otrs/bin/otrs.SetPermissions.pl --web-group=apache
```

```
[[root@scani kiu]# perl /opt/otrs/bin/otrs.SetPermissions.pl --web-group=apache ]  
Setting permissions on /opt/otrs  
[root@scani kiu]# █
```

On redémarre Apache et MariaDB pour que tout soit actualiser.

```
$ systemctl restart httpd
```

```
$ systemctl restart mariadb
```

```
[[root@scani kiu]# systemctl restart httpd ]  
[[root@scani kiu]# systemctl restart mariadb ]  
[root@scani kiu]# █
```


FINALISATION DE OTRS

Nous pouvons maintenant nous connecter à notre interface web pour finaliser l'installation de OTRS avec l'adresse IP ou notre nom de domaine : XXX.XXX.XXX.XXX/otrs/installer.pl

((OTRS)) Community Edition

Étape 1
Licence

Étape 2
Réglages de Base de Données

Étape 3
Spécifications Générales et Réglages de Messagerie

Étape 4
Terminer

Bienvenue à ((OTRS)) Community Edition

Centuran Consulting
ul. Górczewska 156 lok. 32
01-460 Warszawa
Poland
Téléphone: +48 22 120 18 00

Site web: otrscommunityedition.com
E-mail: info@otrscommunityedition.com
GitHub repository: github.com/Centuran/otrs-community-edition

[Suivant](#)

Powered by ((OTRS)) Community Edition



Nous acceptons le contrat de licence et nous devons créer une base de données. Nous pouvons soit la créer depuis la console soit depuis l'interface.

((OTRS)) Community Edition

Étape 1
Licence

Étape 2
Réglages de Base de Données

Étape 3
Spécifications Générales et Réglages de Messagerie

Étape 4
Terminer

Sélection de la base de donnée (2/4)

Type: MySQL
Oracle
PostgreSQL

Type d'installation: Créer une nouvelle base de données pour OTRS
 Utiliser une base de données existante pour OTRS

[Suivant](#)

Powered by ((OTRS)) Community Edition



Étape 1 Licence	Étape 2 Réglages de Base de Données	Étape 3 Spécifications Générales et Réglages de Messagerie	Étape 4 Terminer
---------------------------	---	--	----------------------------

Configure MySQL (2/4)

Utilisateur:

Mot de Passe:

Si vous avez défini un mot de passe root pour votre base de données, il doit être indiqué ici. Sinon, laissez ce champ vide.

Hôte:

Résultat du contrôle de la base de données

✓ Contrôle de base de donnée effectué avec succès.

Utilisateur de la base de données (Nouveau)

Utilisateur:

Un nouvel utilisateur de la base de données sera créé avec des droits limités pour ce système OTRS.

Mot de Passe:

Répétez le mot de passe:

Mot de passe généré: **ea98WpLc1BO7wIY**

Base de données

Nom de la base de données:

Nous pouvons changer les informations de la base de données mais généralement les informations par défaut sont suffisantes.

Créer la base de données (2/4)

- ✓ CREATE DATABASE `otrs`
- ✓ GRANT ALL PRIVILEGES
- ✓ FLUSH PRIVILEGES
- ✓ Processing otrs-schema
- ✓ Processing otrs-initial_insert
- ✓ Processing post statements

Mise en place Base de données réussie!

Nous devons configurer notre domaine, notre adresse mail, le nom de société, nos fichier log, notre langue etc ...

Paramètres Système (3/4) Configuration de la messagerie (3/4)

ID Système: ✖
Identifiant du système. Chaque numéro de ticket et session HTTP contient

FQDN du système:
Nom de domaine pleinement qualifié de votre système.

E-mail administrateur:
Adresse e-mail de l'administrateur système.

Société:

Journal

Module de journalisation: ✖
Moteur de journalisation à utiliser.

Frontal web

Langue par défaut: ✖
Langue par défaut.

Vérifier les enregistrements MX: ✖
Les adresses emails entrées manuellement sont contre-vérifiées avec les enr de domaine. N'utilisez pas cette option si votre serveur de nom de domaine publiques.

Configurer le mail sortant

Type de mail sortant: ✖
Sélectionner le type de mail sortant

Port mail sortant:
Sélectionner le port mail sortant

Configurer mail entrant

Type de mail entrant: ✖
Sélectionner le type de mail entrant

Hôte mail entrant:
Hôte mail entrant.

Utilisateur mail entrant:
Utilisateur pour mail entrant.

Mot de passe mail entrant:
Mot de passe pour mail entrant.

ou

Une fois fait OTRS nous donne un administrateur et un mot de passe aléatoire.

Fin (4/4)

Page de démarrage: <http://192.168.43.111/otrs/index.pl>

Utilisateur: **root@localhost**

Mot de Passe: **x8heomlObGFY32VK**

Pour que OTRS fonctionne, nous devons activer des daemon. Pour le déclenchement des tickets, l'envoi de mails etc ... sinon nous auront cette erreur

Information à propos de la tâche de fond d'OTRS

The OTRS Daemon is a daemon process that performs asynchronous tasks, e.g. ticket escalation triggering, email sending, etc.

A running OTRS Daemon is mandatory for correct system operation.

Starting the OTRS Daemon

Pour activer les daemon il faut lancer le daemon et renommer les 2 fichiers cron de OTRS.

```
$ su -c "/opt/otrs/bin/otrs.Daemon.pl start" -s /bin/bash otrs
```

```
$ sudo cp /opt/otrs/var/cron/aaa_base.dist /opt/otrs/var/cron/aaa_base
```

```
$ sudo cp /opt/otrs/var/cron/otrs_daemon.dist /opt/otrs/var/cron/otrs_daemon
```

On rajoute ensuite une tâche cron pour l'utilisateur otrs qui exécutera les fichiers cron toutes les 5 minutes.

```
$ su -c "/opt/otrs/bin/Cron.sh start" -s /bin/bash otrs
```

Partie 2 – Validation.

Après avoir apporté quelques modifications techniques nous pouvons voir que OTRS Scani fonctionne.

The screenshot displays the OTRS Community Edition interface. At the top, there is a navigation bar with the OTRS logo, a search bar, and the text "((OTRS)) Community Edition". Below the navigation bar, there are tabs for "Tableau de bord", "Ticket", and "Coopérateurs".

The main content area is divided into several sections:

- Nouveaux tickets:** A table showing a list of tickets with columns for "TICKET#", "ÂGE", and "TITRE". The table contains 10 rows of data, including ticket numbers like 42011094 and 42011093, and titles such as "[Msg web] (je...@free.fr)" and "Undelivered Mail Returned to Sender".
- Paramètres:** A section for settings, currently showing "Stats sur 7 jours".
- Stats sur 7 jours:** A line graph showing the number of tickets created (Créé) and closed (Fermés) over a seven-day period. The x-axis represents days from Sunday (Dim) to Saturday (Sam), and the y-axis represents the number of tickets, ranging from 0 to 87. A significant peak is visible on Monday (Lun).
- Utilisateurs actuellement connectés:** A section showing the number of operators (Opérateurs) and cooperators (Coopérateurs) currently connected. It lists 4 operators and 0 cooperators.
- Tickets avec date de rappel atteinte:** A section showing tickets where the reminder date has been reached. It currently shows "aucun" (none).
- Tickets ouverts:** A section showing open tickets. It currently shows 101 tickets.

Nous recevons les tickets par mail qui les ouvrent directement par un ticket. La gestion à été considérablement favorisée et nous avons diminué nos temps de réponse.

Partie 3 – Veille technologique.

A la place de OTRS nous aurions pu utiliser GLPI (mais il pose problème avec notre SI) ou une solution propriétaire comme Zendesk.

Link

OTRS : <https://otrscommunityedition.com/>

GLPI : <https://glpi-project.org/fr/>

ZenDesk : <https://www.zendesk.fr/>