

# Bases De Données

*TP PostgreSQL*

*par*

[julien.breyault@u-pec.fr](mailto:julien.breyault@u-pec.fr)

## Introduction

---

Le but de ce TP est :

- Mettre en place une **plateforme DEBIAN GNU/LINUX** servant de base à un **serveur LAPP (LAMP avec PostgreSQL)**
- Y ajouter les composants, notamment PostgreSQL
- Installer un **CMS**
- Administrer la base de données à l'aide d'**outils en ligne de commande et graphiques**

N'hésitez pas à créer des « snapshots » **à chaque étape-clé** afin de pouvoir revenir en arrière.

## Table de matières

Introduction.....	2
Système « de base ».....	4
Installation de la plateforme « LAMP ».....	5
Installation de tiny-tiny-RSS (agrégateur de flux RSS comme ex-google_reader).....	7
Réplication « streaming » PostgreSQL.....	8
Récupération de mot de passe oublié ;- ).....	10
Tuning bases en production.....	11



debian

## Systeme « de base »

---

- Mettre les machines en NAT avec réseau activé entre machines virtuelles (Réseau NAT).
- Utiliser la machine debian serveur de base avec les composants : SSH et composants usuels (pas poste de travail et serveur d'impression!)



debian

## Installation de la plateforme « LAMP »

---

```
# apt-get update
# apt-get install postgresql postgresql-contrib
postgresql-doc-9.6
```

- **Modifier le mot de passe de l'utilisateur système postgres :**

```
# passwd postgres
```

- **Modifier le mot de passe de l'utilisateur administrateur de la base postgres :**

```
# su - postgres
psql -d template1 -c "ALTER USER postgres WITH PASSWORD
'votremotdepasse'"
OU :
# psql
postgres=# \password postgres
```

- **Créer une base :**

```
# su - postgres
createdb mytestdb
```

- **Connectez-vous à cette base :**

```
psql mytestdb
```

- **Pour avoir de l'aide une fois connecté à la base :**

```
\h
```

- **Pour terminer la connection à la base :**

```
\q
```

- **Pour créer un utilisateur :**

```
# psql -d template1 -c "create user test with password
'motdepasse'"
```

- **Pour gérer la base à distance (comme avec PHPMYAdmin), il faut activer l'accès dans la base :**

```
# su - postgres
psql template1 < /usr/share/postgresql/9.6/extension/adminpack--
1.0.sql
```

```
exit
```

```
OU :
```

```
postgres=# CREATE EXTENSION adminpack;
CREATE EXTENSION
```

- **Activation des connexions TCP/IP :**

```
# nano /etc/postgresql/9.6/main/postgresql.conf
listen_addresses = 'localhost'
password_encryption = on

# service postgresql restart
```

- **Administration web :**

Ce n'est pas le sujet direct du TP, mais on aura besoin aussi d'un serveur web :

```
# apt-get install apache2
```

Allez sur votre navigateur avec une autre machine virtuelle et tapez l'adresse de votre serveur pour vérifier qu'il est bien lancé.

Nous ajoutons les composants PHP :

```
# apt-get install php libapache2-mod-php
```

Et rechargeons la nouvelle configuration apache :

```
# /etc/init.d/apache2 restart
```

On teste la configuration PHP :

```
# nano /var/www/info.php
<?php
phpinfo();
?>
```

Et on ouvre notre navigateur sur cette page (<http://ipduserveur/info.php>)

- **On installe l'interface d'administration PHP**

```
# apt-get install phppgadmin
```

```
# nano /etc/apache2/conf-available/phppgadmin.conf
```

décommenter (si présent, sinon rajouter le temps du TP) :  
allow from all

Activer la conf en mettant un lien symbolique dans sites-enabled vers ce fichier qui est dans conf-available

```
# service apache2 restart
```

```
# nano /usr/share/phppgadmin/conf/config.inc.php
et indiquer aussi :
$config['extra_login_security'] = false;
```

```
# service postgresql restart
# service apache2 restart
Connectez-vous ici : http://ipduserveur/phppgadmin
```



debian

## **Installation de tiny-tiny-RSS (agrégateur de flux RSS comme ex-google\_reader)**

---

Les modules php à installer :

<https://tt-rss.org/gitlab/fox/tt-rss/wikis/PhpCompatibilityNotes>

Pour récupérer les sources, vous aurez besoin de GIT (facile à mettre à jour par « GIT PULL » ensuite lorsque TT-RSS vous alerte que quelque-chose a été modifié dans le code) :

```
# apt-get install git-core
```

Et maintenant, vous continuez tout seuls ;-)

<https://tt-rss.org/gitlab/fox/tt-rss/wikis/InstallationNotes>

Ensuite pour la mise à jour des flux :

<https://tt-rss.org/gitlab/fox/tt-rss/wikis/UpdatingFeeds>



debian

## Réplication « streaming » PostgreSQL

---

Vous avez besoin de deux serveurs DEBIAN avec PostgreSQL installé (le second ne doit pas être une copie du premier après toutes les étapes mais un serveur postgre « vierge ») :

**Maître** 192.168.0.1

**Esclave** 192.168.0.2

- **SUR LE MAITRE :**

```
nano -w /etc/postgresql/9.6/main/postgresql.conf
```

```
listen_addresses = '*'
```

```
wal_level = hot_standby
```

```
wal_keep_segments = 10 (augmenter si la dernière commande donne une table d'état de réplication vide)
```

```
max_wal_senders = 3
```

- **On crée un utilisateur spécifique pour la réplication:**

```
psql -h localhost -U postgres -W -c "CREATE USER utilrepl WITH REPLICATION PASSWORD 'motdepasse';"
```

- **Et on l'autorise à se connecter :**

```
nano -w /etc/postgresql/9.6/main/pg_hba.conf
```

```
host replication utilrepl 192.168.0.2/24 md5
```

- **On arrête le service :**

```
service postgresql stop
```

- **SUR L'ESCLAVE :**

```
nano -w /etc/postgresql/9.6/main/postgresql.conf
```

```
listen_addresses = '*'
```

```
hot_standby = on
```

- **On arrête l'esclave et on détruit tous les fichiers de la base**

```
/etc/init.d/postgresql stop
```

```
cd /var/lib/postgresql/9.6/main/
```

```
rm -rf *
```

```
nano -w /var/lib/postgresql/9.6/main/recovery.conf
```

```
primary_conninfo = 'host=192.168.0.1 port=5432 user=utilrepl password=motdepasse'
```

```
standby_mode = on
```

- **On copie tout du maître vers l'esclave (installer rsync si pas dispo) :**

```
rsync -av /var/lib/postgresql/9.6/main/* 192.168.0.2:/var/lib/postgresql/9.6/main/
```

- **On peut maintenant démarrer les deux serveurs :**

service postgresql start (sur les deux)

- **On vérifie l'état de la réplication :**

```
psql -h localhost -U postgres -W -c "select * from pg_stat_replication;"
```



debian

## Récupération de mot de passe oublié ;-)

---

```
# service postgresql stop
# nano /usr/lib/postgresql/data/pg_hba.conf
```

- **Chercher :**

```
local all all ...
```

- **Et modifier pour avoir ça :**

```
local all postgres trust
```

```
# service postgresql start
# su - postgres
$ psql -d template1 -U postgres
alter user postgres with password 'votrenouveaumotdepasse';
```

ou en une seule commande :

```
# psql -U postgres template1 -c "alter user postgres with password 'votrenouveaumotdepasse';"
```

- **Remettre le fichier suivant à l'état initial :**

```
# nano /usr/lib/postgresql/data/pg_hba.conf
```

C'est à dire changer :

```
local all all trust
```

Pour :

```
local all postgres ident
```

```
# service postgresql start
```

**Sauvés ;-)**

## Tuning bases en production

---

<http://www.dsfc.net/logiciel-libre/postgresql/tuning-postgresql-configuration-memoire-linux/>

## Backup

---

Utilisation des solutions présentées dans la partie MariaDB possible.

Une solutions spécifique à PostgreSQL existe (sauvegarde au fil de l'eau) :

<http://www.pgbarman.org/>



Ce document est publié sous la licence libre **Creative Commons-BY-SA** :

<http://creativecommons.org/licenses/by-sa/3.0/deed.fr>

**BY : Paternité.** Vous devez citer le nom de l'auteur original.

**SA : Partage des Conditions Initiales à l'Identique.** Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

En outre, à chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création. Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.