



1ère INSTALLATION QGIS-SERVER + LIZMAP

Légende

commande	à écrire dans la console du serveur
manipulation	action à faire
texte	texte à rajouter dans un éditeur de texte ouvert
description	description des différentes manipulations

Cette documentation est dérivée de la documentation officielle (<https://docs.3liz.com/fr/>) mais dans le contexte du PnCalanques et avec PHP7.

Il est conseillé de se référer à la documentation officielle en complément et pour être à jour.

CONFIGURATION DU SERVEUR

Configurer les locales du serveur et le connecter à un serveur de temps

```
sudo locale-gen fr_FR.UTF-8

sudo dpkg-reconfigure locales
décocher (en utilisant la barre espace) en_US.UTF 8 UTF 8
cocher (en utilisant la barre espace) fr_FR.UTF-8 UTF-8
ok (tab puis Entrée)
choisir fr_FR.UTF8
ok

sudo dpkg-reconfigure tzdata
choisir Europe > ok
choisir Paris

sudo apt-get install ntp ntpdate
```

Installation des paquets nécessaires

Faire un update de la liste des paquets pour s'assurer de télécharger et installer les dernières versions disponibles :

```
sudo apt-get update
```

Installation des paquets nécessaires :

```
sudo apt-get install xauth htop curl apache2 libapache2-mod-fcgid libapache2-mod-php
php-cgi php-gd php7.0-sqlite php-curl php-xmlrpc python-simplejson python-software-
properties
```

Configuration de php 7.0

Configuration du serveur apache/php :

```
sudo nano /etc/apache2/conf-available/php7.0-cgi.conf
```

ajouter à la fin :

```
<Directory /usr/share>
    AddHandler fcgid-script .php
    FCGIWrapper /usr/lib/cgi-bin/php7.0 .php
    Options ExecCGI FollowSymLinks Indexes
</Directory>

<Files ~ (\.php)>
    AddHandler fcgid-script .php
    FCGIWrapper /usr/lib/cgi-bin/php7.0 .php
    Options +ExecCGI
    allow from all
</Files>
```

Pour enregistrer : Ctrl + O > Entrée

Pour quitter l'éditeur de texte : Ctrl + X

Enregistre la configuration du module php d'Apache :

```
sudo a2enconf php7.0-cgi
```

Configuration du mpm-worker d'apache

On modifie le fichier de configuration d'Apache pour adapter les options du mpm_worker à la configuration du serveur.

On modifie la configuration d'Apache :

```
sudo nano /etc/apache2/apache2.conf
```

Ajouter à la fin :

```
<IfModule mpm_worker_module>
    StartServers      4
    MinSpareThreads   25
    MaxSpareThreads   100
    ThreadLimit        64
    ThreadsPerChild    25
    MaxClients         150
    MaxRequestsPerChild 0
</IfModule>
```

Configuration de mod_fcgid d'apache

QGIS Server fonctionne en mode fcgi. Il faut donc configurer le mod_fcgid d'Apache pour l'adapter aux capacités du serveur.

```
sudo nano /etc/apache2/mods-enabled/fcgid.conf
```

Ajouter à la fin :

```
<IfModule mod_fcgid.c>
  AddHandler    fcgid-script .fcgi
  FcgidConnectTimeout 300
  FcgidIOTimeout 300
  FcgidMaxProcessesPerClass 50
  FcgidMinProcessesPerClass 20
  FcgidMaxRequestsPerProcess 500
  IdleTimeout   300
  BusyTimeout   300
</IfModule>
```

Configuration de la compression

```
sudo nano /etc/apache2/mods-available/deflate.conf
```

intégrer dans la balise <IfModule mod_filter.c>

```
    <Location />
        # Insert filter
        SetOutputFilter DEFLATE
        # Netscape 4.x encounters some problems ...
        BrowserMatch ^Mozilla/4 gzip-only-text/html
        # Netscape 4.06-4.08 encounter even more problems
        BrowserMatch ^Mozilla/4\.0[678] no-gzip
        # MSIE pretends it is Netscape, but all is well
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
        # Do not compress images
        SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-
vary
        # Ensure that proxy servers deliver the right content
        Header append Vary User-Agent env=!dont-vary
    </Location>
```

Redémarrage d'Apache

```
sudo ln -sf /etc/apache2/mods-available/headers.load /etc/apache2/mods-
enabled/headers.load
sudo service apache2 restart
```

INSTALLATION DE POSTGRESQL

Installer les paquets:

```
sudo apt-get install postgresql postgresql-contrib postgis php-pgsql
```

Créer un super-utilisateur Lizmap:

```
sudo su - postgres
```

```
createuser lizmap -superuser
? mettre peut-être plutôt lizmap_admin ?
```

Modifier les mots de passe :

```
psql
```

```
ALTER USER postgres WITH ENCRYPTED PASSWORD '*****'; (modifier le mot de passe.)
```

Attention : penser bien à ajouter le point virgule en fin d'instruction pour que la commande s'exécute correctement !)

```
ALTER USER lizmap WITH ENCRYPTED PASSWORD '*****'; (modifier le mot de passe)
```

```
\q (pour sortir de la console postgresql)
```

```
exit (pour sortir de la session d'utilisateur postgres et revenir à l'utilisateur connecté au serveur)
```

!/ \ REMARQUE : les parties géolocalisation, pg_tune, pure-ftpd ont été omises ici par rapport au document d'installation de Lizmap.

INSTALLATION DE QGIS-SERVER

On ajoute les sources pour qgis-server :

```
sudo gpg --recv-key CAEB3DC3BDF7FB45 (refaire l'opération si plantage à la première exécution)
```

```
sudo gpg --export --armor CAEB3DC3BDF7FB45 | sudo apt-key add -
```

```
sudo nano /etc/apt/sources.list.d/debian-gis.list
```

Y coller le contenu suivant :

```
deb http://qgis.org/debian xenial main
```

```
deb-src http://qgis.org/debian xenial main
```

Après avoir paramétré le lien vers le serveur contenant les paquets à jour, on peut installer qgis-server :

```
sudo apt-get update
```

```
sudo apt-get install qgis-server python-qgis
```

On active les modules suivants d'Apache :

```
sudo a2enmod rewrite
```

```
sudo a2enmod cgi
```

INSTALLATION DE UNZIP (s'il n'est pas déjà installé)

```
sudo apt-get install unzip
```

INSTALLATION DE LIZMAP

```
cd /var/www/
```

On crée une constante MYAPP dans laquelle on met le mot « lizmap-web-client » (= le nom de l'application qu'on va installer) on refait la même chose avec le numéro de la version qu'on va installer :

```
MYAPP=lizmap-web-client
```

VERSION=3.1.11 - aller voir sur le site la dernière version de lizmap-web-client (<https://github.com/3liz/lizmap-web-client/releases>)

On télécharge le fichier correspondant au nom d'application et au numéro de version indiqué plus haut :

```
sudo wget https://github.com/3liz/$MYAPP/archive/$VERSION.zip
```

On le dézippe dans le répertoire « www » :

```
sudo unzip $VERSION.zip
```

On la renomme en enlevant le numéro de version :

```
sudo mv $MYAPP-$VERSION $MYAPP
```

On crée un raccourci (lien symbolique) pour simplifier les URL dans le navigateur :

```
sudo ln -s /var/www/$MYAPP/lizmap/www/ /var/www/html/lm
```

On supprime le fichier .zip:

```
sudo rm $VERSION.zip
```

Lizmap donne à Apache les droits d'écriture dans son arborescence :

```
cd /var/www/$MYAPP/
```

```
sudo lizmap/install/set_rights.sh www-data www-data
```

Dans le répertoire config, on copie les fichiers de configuration par défaut pour en conserver une copie (qui comportera l'extension .dist) :

```
cd lizmap/var/config
```

```
sudo cp lizmapConfig.ini.php.dist lizmapConfig.ini.php
```

```
sudo cp localconfig.ini.php.dist localconfig.ini.php
```

```
sudo cp profiles.ini.php.dist profiles.ini.php
```

Modification des fichiers de configuration de PostGreSQL pour y intégrer le module d'authentification de Lizmap

Ouvrir l'adressage de postgresql pour qu'il écoute les connexions qui sont faites depuis le réseau :

```
sudo nano /etc/postgresql/9.5/main/pg_hba.conf
```

Ajouter au niveau des connexions IPv4 une ligne avec l'adresse du réseau à ouvrir :

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		peer
# IPv4 local connections:					
host	all		all	127.0.0.1/32	md5

```
host    all             all             192.168.186.0/24      md5    -- à modifier en
fonction du réseau.
```

Ouvrir le fichier :

```
sudo nano /etc/postgresql/9.5/main/postgresql.conf
```

Décommenter la ligne `listen_addresses` (enlever le `#` au début) et remplacer `'localhost'` par `'*'` pour que le serveur écoute l'ensemble des connexions :

```
#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
```

On redémarre PostgreSQL :

```
sudo service postgresql restart
```

```
sudo su - postgres
```

On crée une base de données pour stocker les tables nécessaires :

```
psql
CREATE DATABASE lizmap_admin;
\q
exit
```

Utilisation de PostgreSQL en tant que base de données d'administration (Optionnel)

```
sudo nano /var/www/lizmap-web-client/lizmap/var/config/profiles.ini.php
```

Commenter les 6 lignes suivantes en ajoutant un « ; » devant :

```
:[jdb:jauth]
;driver=sqlite3
;database="var:db/jauth.db"
```

```
:[jdb:lizlog]
;driver=sqlite3
;database="var:db/logs.db"
```

Ajouter à la fin du fichier :

```
[jdb:jauth]
driver=pgsql
database="lizmap_admin"
```

```
host="localhost"
user="postgres"
password="*****"
port=5432

[jdb:lizlog]
driver=pgsql
database="lizmap_admin"
host="localhost"
user="postgres"
password="*****"
port=5432
```

On active le module postgresql pour PHP :

```
sudo nano /etc/php/7.0/apache2/php.ini
```

On décommente les lignes correspondant à pgsql (enlève le ;) :

```
extension=php_pdo_pgsql.dll
;extension=php_pdo_sqlite.dll
extension=php_pgsql.dll
;extension=php_shmop.dll
```

```
sudo apt-get install php7.0-xml
sudo service apache2 restart
```

Conseil : faire un snapshot avant l'installation de la version de demo pour pouvoir revenir dessus si jamais.

Installation de la version de démo

Pour activer le répertoire des démos :

```
sudo nano /var/www/lizmap-web-client/lizmap/var/config/localconfig.ini.php
```

Ajouter à la fin du fichier :

```
[modules]
lizmap.installparam=demo
```

On revient au répertoire de départ:

```
cd ../../..
```

INSTALLATION DE LIZMAP WEB CLIENT

On lance l'installateur :

```
sudo php lizmap/install/installer.php
```

/!\ Erreur rencontrée lors de l'installation:

*[error] An error occurred during the installation of the module jacl2db: invalid query (ERROR: insert or update on table "jacl2_user_group" violates foreign key constraint "jacl2_user_group_id_aclgrp_fkey"
DETAIL: Key (id_aclgrp)=(__priv_admin) is not present in table "jacl2_group".(INSERT INTO "jacl2_user_group" ("login","id_aclgrp") VALUES ('admin','__priv_admin')))*

Si on veut, on peut intégrer manuellement la ligne:

```
INSERT INTO public.jacl2_group(id_aclgrp, name, grouptype, ownerlogin)  
VALUES ('__priv_admin', 'admin', 2, 'admin');
```

Ensuite, relancer l'installation.

On prépare l'arborescence dans lequel on va venir ranger et synchroniser nos projets qgis :

```
sudo mkdir /var/www/lizmap-web-client/ftp
```

On crée un groupe pour se connecter en FTP, pour lui donner les droits en écriture sur notre répertoire FTP:

```
sudo addgroup client_ftp
```

Par défaut, le répertoire « var » n'est pas modifiable en FTP. On crée alors une exception qui permet à un utilisateur se connectant en FTP à ce dossier de pouvoir y ajouter des dossiers et d'y glisser-déposer des fichiers :

On crée un utilisateur pour mettre dans ce groupe :

```
sudo adduser lizmap_ftp
```

Lui attribuer mot de passe et cliquer sur entrée pour les infos demandées

On l'affecte à ce groupe :

```
sudo adduser lizmap_ftp client_ftp
```

On donne les droits d'écriture :

```
sudo chown lizmap_ftp:client_ftp /var/www/lizmap-web-client/ftp -R
```

Paramétrer la gestion du cache (optionnel)

A faire si on veut s'intéresser à la gestion du cache (pas testé au PnCal) :

```
sudo mkdir /var/www/lizmap-web-client/cache  
chmod 700 /var/www/lizmap-web-client/cache -R  
chown www-data:www-data /var/www/lizmap-web-client/cache -R
```

Activation de l'envoi de mail en smtp sécurisé (nécessaire pour modérer la création des comptes extérieurs)


```
sudo nano /var/www/lizmap-web-client/lizmap/var/config/mainconfig.ini.php
```

On paramètre les sessions [mailer] et [mailLogger] (cf. passages en gras) :

```
[mailer]
webmasterEmail="damien.frazzoni@calanques-parcnational.fr"
webmasterName="Damien FRAZZONI"

; how to send mail : "mail" (mail()), "sendmail" (call sendmail), or "smtp" (send
directly to a smtp)
mailerType=smtp
; Sets the hostname to use in Message-Id and Received headers
; and as default HELO string. If empty, the value returned
; by SERVER_NAME is used or 'localhost.localdomain'.
hostname=
sendmailPath="/usr/sbin/sendmail"

; if mailer = smtp , fill the following parameters

; SMTP hosts. All hosts must be separated by a semicolon :
"smtp1.example.com:25;smtp2.example.com"
smtpHost=smtp.espaces-naturels.fr
; default SMTP server port
smtpPort=465
; secured connection or not. possible values: "", "ssl", "tls"
smtpSecure="ssl"
; SMTP HELO of the message (Default is hostname)
smtpHelo=
; SMTP authentication
smtpAuth=on
smtpUsername="informatique@calanques-parcnational.fr"
smtpPassword="*****"
; SMTP server timeout in seconds
smtpTimeout=10

ET

[mailLogger]
email="damien.frazzoni@calanques-parcnational.fr"
emailHeaders="Content-Type: text/plain; charset=UTF-8\nFrom: damien.frazzoni@calanques-
parcnational.fr\nX-Mailer: Jelix\nX-Priority: 1 (Highest)\n"
```

Adaptation de la configuration de PostgreSQL (optionnel, à faire quand tout est ok pour une mise en production)

Nous allons utiliser pg_tune, un utilitaire qui permet de générer automatiquement un fichier de configuration de PostgreSQL adapté aux propriétés du serveur (mémoire vive, processeurs, etc.)

PostgreSQL Tuning with pg_tune

```
pgtune -i /etc/postgresql/9.5/main/postgresql.conf -o
/etc/postgresql/9.5/main/postgresql.conf.pgtune --type Web
cp /etc/postgresql/9.5/main/postgresql.conf
/etc/postgresql/9.5/main/postgresql.conf.backup
cp /etc/postgresql/9.5/main/postgresql.conf.pgtune
/etc/postgresql/9.5/main/postgresql.conf
nano /etc/postgresql/9.5/main/postgresql.conf
```

Restart to check any problems

```
service postgresql restart
```

If error messages, increase the linux kernel configuration variables

```
echo "kernel.shmall = 4294967296" >> /etc/sysctl.conf # to increas shred buffer param
in kernel
echo "kernel.shmmax = 4294967296" >> /etc/sysctl.conf
echo 4294967296 > /proc/sys/kernel/shmall
echo 4294967296 > /proc/sys/kernel/shmmax
sysctl -a | sort | grep shm
```

Restart PostgreSQL

```
service postgresql restart
```