
GUIDE D'INSTALLATION DE L'APPLICATION GeCOL SUR SERVEUR XUBUNTU 12.04 LTS

INTRODUCTION

Le guide d'installation qui va suivre va tenter d'amener de la simplicité et de centraliser toutes les informations nécessaires à l'installation de l'application GeCol. En effet, une documentation importante existe déjà sur internet, néanmoins, celle-ci reste éparse et imprécise. Ainsi, des directives précises pour l'installation sur système Xubuntu 12.04 LTS seront énoncés successivement afin d'amener l'application à être fonctionnelle le plus rapidement possible.

Dans un premier temps, nous décrirons l'installation de la pile LAPP (**L**inux **A**pache **P**HP **P**ostgreSQL) qui constituera l'environnement d'exécution de l'application.

Nous verrons ensuite l'installation de l'application GeCol proprement dite avec la configuration des composants de la pile LAPP.

Enfin, nous expliquerons l'installation du serveur de carte géographique qui a pour but de fournir des cartes qu'importe si notre serveur Xubuntu a un accès internet ou pas.

Notes Importantes:

Toute ligne débutant avec le symbole “#” dans ce guide d'installation vous indique qu'il faut entrer la commande qui suit dans un terminal. En effet, n'entrez surtout pas le symbole “#”.

Certaines commandes seront à entrer en tant qu'utilisateur **root**, c'est à dire, l'administrateur du serveur. Pour se faire, rien de plus simple :

Se connecter en tant que root

```
[<nom utilisateur>@<nom-ordinateur> <répertoire courant>]$ sudo su - root
```

Entrez alors le mot de passe de l'utilisateur courant:

```
Mot de passe :
```

Remarque : <nom utilisateur> = le nom que vous avez créé à l'installation de linux

<nom-ordinateur> = nom de la machine

<répertoire courant> = le répertoire dans lequel vous êtes

le # ou le \$ sont une information sur le type d'utilisateur : # pour root et \$ pour les autres utilisateurs. C'est après cet élément que l'on tape nos commandes linux

Ces 4 éléments ne sont pas à taper (ils s'affichent à chaque ligne) avant les commandes, ils font partie de ce que l'on appelle l'invite de commande dans un terminal linux.

Cependant, d'autres manipulations que nous verrons par la suite nécessiteront de reprendre le terminal avec d'autres utilisateurs.

Ce guide s'applique parfaitement bien à Ubuntu, Xubuntu étant une déclinaison d'Ubuntu avec l'interface graphique Xfce.

1) PREREQUIS A L'APPLICATION GeCOL : LA PILE LAPP

La pile LAPP est essentielle à l'application GeCol. Nous allons guider l'installation de chacune des composantes de la pile étape par étape.

Avant tout, faisons une mise à jour du dépôt :

```
[root@<nom-ordinateur>:<répertoire courant>]# apt-get update
```

1.1 INSTALLATION DU SERVEUR WEB APACHE

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

Commandes réutilisables

Pour démarrer le serveur Apache :

```
#service apache2 start
```

ou

```
#!/etc/init.d/apache2 start
```

Pour le redémarrer, tout simplement :

```
#service apache2 restart
```

1.2 INSTALLATION DU LANGAGE DE SCRIPT PHP

```
#apt-get install php5 php-pear php5-pgsql php5-gd php5-ldap php5-mcrypt php5-xsl
```

Remarque : pour ubuntu : pas de package php5-mcrypt

1.3 INSTALLER CLEINT SUBVERSION

```
apt-get install subversion
```

1.4 INSTALLATION DU SERVEUR POSTGRESQL

Donc pour télécharger le package de la version 9.1 :

```
#apt-get install postgresql-9.1 postgresql-9.1-debversion libpq-dev
```

INITIALISATION DE POSTGRESQL:

Un cluster est initialisé par défaut, nous démarrerons ce dernier.

Le propriétaire de ce cluster étant **postgres** (voir l'explication qui suit dans la section concernant la définition du mot de passe pour l'utilisateur *postgres*), on entre la commande qui suit avant de démarrer le serveur PostgreSQL :

DEFINITION DU MOT DE PASSE POUR L'UTILISATEUR **POSTGRES**:

PostgreSQL, en s'installant, crée un utilisateur Linux par défaut de nom **postgres** en plus de l'utilisateur du même nom propre au serveur PostgreSQL. Le compte utilisateur Linux **postgres** est le seul compte permettant d'accéder à l'invite de commande SQL via la commande bash *psql*. Ni l'utilisateur Linux **postgres** et ni l'utilisateur **postgres** du serveur PostgreSQL n'ont de mot de passe. Le problème étant que l'absence de mot de passe envoie systématiquement une erreur pour les deux profils utilisateur. Nous allons ici modifier uniquement le mot de passe de l'utilisateur **postgres** sur le serveur PostgreSQL car il est indispensable lorsque l'application demandera à se connecter au serveur de données PostgreSQL. Comme dit précédemment, les mécanismes de sécurité de PostgreSQL font que l'absence de mot de passe mène systématiquement à une erreur lorsque l'application souhaite se connecter au serveur de données même si, répété également, l'utilisateur est créé par défaut sans mot de passe.

Pour ouvrir l'invite de commande SQL du serveur PostgreSQL qui va permettre le changement de mot de passe :

Se connecter avec l'utilisateur postgres de linux :

```
[root@<nom-ordinateur> <répertoire courant>]# sudo -i -u postgres
```

Puis se connecter au SGBD avec l'utilisateur postgres de Postgresql

```
[postgres@<nom-ordinateur> <répertoire courant>]# psql postgres
```

Puis, dans l'invite :

```
postgres=#alter user postgres with password 'postgres';
```

On prend **postgres** pour éviter les oublis.

Si tout se passe correctement, on devrait avoir :

```
ALTER ROLE
```

Quittez la console d'administration PostgreSQL en entrant :

```
postgres=# \q
```

Se déconnecter de l'utilisateur postgres linux :

```
[postgres@<nom-ordinateur> <répertoire courant>]# exit
```

Configuration du fichier *pg_hba.conf* de PostgreSQL :

Se mettre sur le répertoire et Ouvrez le fichier votre éditeur de texte préféré :

```
[root@<nom-ordinateur>:<répertoire courant>]#cd
/etc/postgresql/9.1/main/pg_hba.conf
[root@<nom-ordinateur> <répertoire courant>]#gedit pg_hba.conf &
```

Changer toutes les valeurs en colonne méthode à **md5** comme ceci :

```
#Database administrative login by Unix domain socket

local    all             postgres                                md5

# TYPE  DATABASE  USER  ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local    all             all                                md5

# IPv4 local connections:
host     all             all             127.0.0.1/32    md5

# IPv6 local connections:
host     all             all             ::1/128         md5
```

Pour que ces modifications prennent effet :

```
[root@<nom-ordinateur> <répertoire courant>]#/etc/init.d/postgresql restart
```

CONFIGURATION D'UN ACCES A DISTANCE DU SERVEUR DE DONNEES POSTGRESQL (OPTIONNELLE)

Les futurs besoins peuvent vous amener à vouloir héberger le serveur de données sur un autre serveur physique que celui de l'application GeCol. Dans ce cas, le fichier postgresql.conf dans le répertoire /var/lib/pgsql/9.1/data doit être modifié:

Ouvrez le fichier avec votre éditeur préféré, décommenter la ligne **listen_addresses** si elle ne l'est pas déjà et Remplacez **localhost** par **'*'** :

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

# - Connection Settings -
```

```
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost', '*' = all
                                # (change requires restart)
#port = 5432                     # (change requires restart)
```

Vous pouvez également restreindre l'accès à certaines adresses IP, à vous de les entrer en les séparant par une simple virgule.

Pour une meilleure sécurité, vous pouvez également changer le port par défaut à 5432. Décommenter la ligne concernant le port et faire les changements souhaités.

Redémarrer le serveur pour tenir compte des changements (en redémarrant le service)

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

Références : <http://doc.ubuntu-fr.org/postgresql>

2) INSTALLATION DE GECOL

Nous allons enfin pouvoir installer l'application GeCol proprement dite.

2.1 CREATION D'UN NOUVEL UTILISATEUR SUR LE SERVEUR POSTGRESQL (OPTIONNELLE)

L'utilisateur **postgres** est l'utilisateur par défaut du serveur. Il est de ce fait l'administrateur et le super utilisateur du serveur PostgreSQL. Vous pouvez néanmoins créer un nouvel utilisateur et l'associer (le rendre propriétaire) à la base de données que nous créerons dans les lignes qui vont suivre. Néanmoins, l'utilisateur **postgres** peut être propriétaire de la future base de données et, ainsi, le choix d'un nouvel utilisateur vous est laissé.

Pour la création d'un nouvel utilisateur, nous devons préalablement Ouvrez l'invite de commande SQL en tant qu'utilisateur **postgres** :

En tant qu'utilisateur root sous linux, se connecter au SGBD avec l'utilisateur postgres du SGBD

```
#psql -U postgres
```

Créons notre utilisateur :

```
postgres=# create user <nomutilisateur> with password '<motdepasse>';
```

où *nomutilisateur* est le nom choisi pour le nouvel utilisateur et *motdepasse* le mot de passe choisie.

Si tout se passe bien :

```
CREATE ROLE
```

2.2 CREATION DE LA BASE DE DONNEES

Et pour créer une nouvelle base de données :

```
postgres=# create database <nombd> owner=<nomutilisateur>;
```

où *nombd* est le nom choisi pour votre nouvelle base de données (par ex : arim) et *nomutilisateur* est le nom d'utilisateur propriétaire de la base de données à créer. Ce nom peut être **postgres**, l'utilisateur par défaut, ou vous pouvez associer la base de données à un autre utilisateur. Pour cela, revenez à la section précédente 2.1.

2.3 CHARGEMENT DE LA BASE DE DONNEES

Sortez de l'invite de commande SQL avec :

```
postgres=# \q
```

Récupérer les dumps (base de données vide et bases de données avec dictionnaires) de la base de données gecol au niveau de /var/www/

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

1. soit sur le svn public:

```
svn co https://svn.mpl.ird.fr/arhem-pins/tags/data/database/
```

2. Soit sur le site www.gecol.ird.fr (dump_gecol-v1.zip ou dump_gecol_schema_v1.zip) et dézipper l'archive dans le dossier /var/www/database/

N'oublier pas de créer le dossier data auparavant (si il n'existe pas) : `mkdir /var/www/database`

Se mettre en tant qu'utilisateur **root** si ce n'est pas déjà le cas.

```
[postgres@<nom-ordinateur>:<répertoire courant>]$exit
```

Pour installer la base de données souhaitée, entrez la commande dans le terminal :

```
[root@<n><wd>]#psql -d <nomdb> -U <nomUtilisateurAyantPermission> -f  
/var/www/database/<nomFichier>.sql
```

Password : Taper le mot de passe de l'utilisateur

où *nomdb* désigne la base de données à alimenter (la base de données doit être au préalable créée comme précédemment - section 2.2), *nomUtilisateurAyantPermission* désigne le nom d'utilisateur qui fait l'opération (Bien évidemment, l'utilisateur doit être autorisé à faire cette opération en étant soit propriétaire de la base de données soit super utilisateur **postgres**) et *nomFichier.sql* désigne donc le fichier contenant les instructions d'installation de la base de données (instructions de création de la structure de la base de données et instructions d'insertion des enregistrements).

L'opération est longue surtout si le fichier SQL est important.

2.4 CONFIGURATION DU SERVEUR APACHE ET DES FICHIERS RELATIFS AU LANGAGE PHP

Les directives suivantes vont en majeure partie vous inviter à modifier des fichiers de configuration. Choisissez votre éditeur de texte préféré.

2.4.1 CONFIGURATION DU FICHIER PHP.INI

Ouvrez le fichier php.ini dans le répertoire /etc/ :

```
#sudo gedit /etc/php5/apache2/php.ini &
```

Vérifiez et Modifiez si besoin les valeurs suivantes : (Utiliser  +  pour la fonction recherche dans **gedit**):

```
file_uploads = On  
post_max_size = 50M
```



```
upload_max_filesize = 50M
max_input_time = -1
memory_limit = 256M
```

2.4.2 METTRE EN PLACE LE DOSSIER DE SOURCE

Récupérer le dossier de source via SVN dans le dossier /var/www/

Cd /var/www/

1. Soit sur le dépôt privé (branches de l'application en développement) :

```
svn co https://svn.mpl.ird.fr/arhem-pins/branches/pinsarim/app/
svn co https://svn.mpl.ird.fr/arhem-pins/branches/pinsarim/cake/
svn co https://svn.mpl.ird.fr/arhem-pins/branches/pinsarim/data/
```

2. sinon le dépôt public :

```
svn co https://svn.mpl.ird.fr/arhem-pins/tags/gecol-4.3.1rc0-beta
svn co https://svn.mpl.ird.fr/arhem-pins/tags/cake/
svn co https://svn.mpl.ird.fr/arhem-pins/tags/data/
```

3. Sinon, télécharger les 3 dossiers (gecol-4.3.1rc0-beta.zip, cake.zip, data.zip) sur le site <http://www.gecol.ird.fr>

Nous avons donc dans le dossier /var/www/ 3 dossiers :

/var/www/app ou /var/www/gecol-4.3.1rc0-beta

/var/www/cake

/var/www/data

Renommer le dossier app (ou gecol-4.3.1rc0-beta) avec le nom que vous voulez donner à votre application. Ce sera ce même nom utilisé pour l'alias.

Exemple : arim, pins, gecol, arim-filariose, etc.

```
mv app <nom application>
```

Ou

```
mv gecol-4.3.1rc0-beta <nom application>
```

Nous désignerons le nom de ce dossier racine par *<nomDossierRacineSource>*

2.4.3 CONFIGURATION DES FICHIERS CONF D'APACHE

Créer le lien symbolique pour le module rewrite en se plaçant dans le répertoire `/etc/apache2/mods-enabled` :

```
ln -s ../mods-available/rewrite.load rewrite.load
```

Dans le fichier `/etc/apache2/sites-available/default`, créer un alias avec la forme suivante :

```
Alias /<nomAlias> /var/www/<nomDossierRacineSource>/app/webroot
<Directory "/var/www/<nomDossierRacineSource>/app/webroot">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

où *<nomAlias>* est le nom qui sera utilisé dans l'URL à entrer dans le navigateur pour accéder à GeCol.

2.4.4 MISE EN PLACE DU FICHIER .HTACCESS

Ce fichier, en plus de la configuration précédente, va permettre notamment au Framework Cake PHP, utilisé par l'application, de fonctionner correctement et plus précisément de réécrire les URLs correctement.

Tout d'abord, plaçons nous sur le répertoire `/var/www/<nomDossierRacineSource>/webroot` via la commande :

```
#cd /var/www/<nomDossierRacineSource>/webroot
```

Pour créer le fichier ;

```
#gedit .htaccess &
```

Y coller :

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /<nomAlias>
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]
</IfModule>
```

où *<nomAlias>* est le nom d'alias créé dans la section 2.4.3

2.4.5 CONFIGURATION DES FICHIERS DE CONFIGURATION DE CAKE PHP.

Copiez le fichier `/var/www/<nomDossierRacineSource>/config/database.php.default` et le coller dans le même répertoire en le renommant **database.php** (donc sans le suffixe **.php**)

```
cd /var/www/<nomDossierEacinesource>/config/  
cp database.php.default database.php
```

Configurer la variable Javascript `$default` du fichier `/var/www/<nomDossierRacineSource>/config/database.php` comme ce qui suit :

```
var $default = array(html/html/  
    'driver' => 'postgres',  
    'persistent' => false,  
    'port' => 5432,  
    'host' => '<ip du serveur de BD ou localhost>',  
    'login' => '<user BD>',  
    'password' => '<password de l\'user de la BD>',  
    'database' => '<nom de la BD>s',  
    'prefix' => '',
```

où le nom de la base de données à utiliser est `<nom de la BD>`, ou `<ip du serveur de BD ou localhost>` est soit l'adresse IP du serveur PostgreSQL à distance ou soit localhost si le serveur est local, `<user BD>` le propriétaire ou le super utilisateur (**postgres**) de la base de données, `<password de l'user de la BD>` est **postgres** si le propriétaire de la base est postgres sinon le mot de passe relatif à l'utilisateur désigné dans le champ `<user BD>`.

Copiez le fichier `/var/www/<nomDossierRacineSource>/config/bootstrap.php.default` et le coller dans le même répertoire en le renommant **bootstrap.php** (donc sans le suffixe **.php**)

```
cp bootstrap.php.default bootstrap.php
```

Deux possibilités d'affichage des menus sont possibles.

Pour utiliser tous les menus : Spécimens – Echantillons – Prélèvements – Collectes – Conservations :

Dans le fichier `/var/www/<nomDossierRacineSource>/config/bootstrap.php`, décommenter **`configure::write('umr','cbgp')`**

et commenter **`configure::write('umr','mivegec');`** comme ce qui suit :

```
configure::write('umr','cbgp');  
//configure::write('umr','mivegec');
```

Pour utiliser les menus : Spécimens – Catalogue Taxonomique, on configure de la manière suivante :

```
//configure::write('umr','cbgp');  
configure::write('umr','mivegec');
```

2.4.6 INSTALLER LES POLICES ARIAL DANS XUBUNTU

Pour se faire, rien de plus simple :

Récupérer les fichiers de polices **arial.ttf** et **ariabd.ttf** dans le répertoire **/var/www/<nom application>/data/fonts/** et en tant que **root**, Copiez ceux-ci dans le répertoire **/usr/share/fonts/truetype/** :

```
#cd /var/www/<dossier racine>/data/fonts/  
sudo cp arial.ttf arialbd.ttf /usr/share/fonts/truetype
```

2.4.7 CHANGER LES DROITS ET PROPRIETAIRES

dans le dossier **/var/www/**

```
cd /var/www/  
chmod -Rf 775 <dossier racine>  
chown -Rf www-data.www-data <dossier racine>
```

2.4.8 FICHIERS DE LOGS

les fichiers de logs de PHP et Apache sont dans **/var/log/apache2**

les fichiers de logs de postgresql sont dans **/var/log/postgresql**

Pour prendre en compte toutes les modifications, réinitialisons le serveur Apache :

```
#sudo service apache2 restart
```

ou

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

A ce stade, vous pouvez tester l'application via un navigateur. Entrez dans la barre d'adresse :

<http://localhost/<nomAlias>> où <nomAlias> est le nom d'alias choisi précédemment.

3) INSTALLATION DU SERVEUR LOCALE DE CARTE

Nous allons aborder ici l'installation d'un serveur WMS implémenté par l'application CGI MapServer. Ce serveur va venir se greffer sur notre serveur web Apache afin de fournir des cartes peu importe si notre serveur CentOS accède ou n'accède pas à Internet. Il est à noter que WMS est un protocole reconnu à l'international qui fournit des spécifications pour l'implémentation d'un "service web de carte géographique". Le protocole WMS nous garantit donc l'interopérabilité avec OpenLayers qui, ce dernier, va permettre de les visualiser en plus des autres cartes auxquels on accède via une connexion Internet.

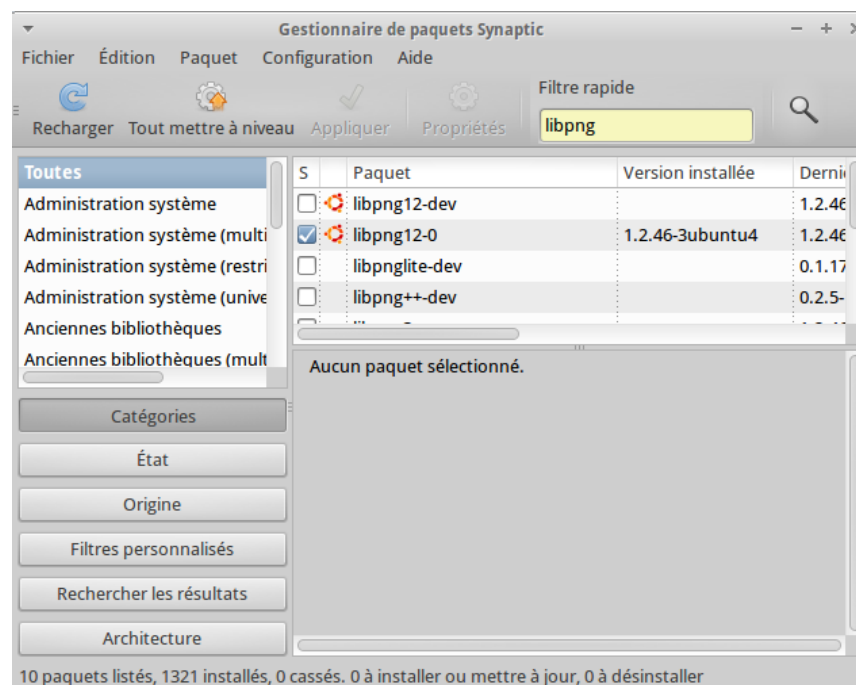
3.1 INSTALLATION DE TOUTES LES LIBRAIRIES NECESSAIRES A MAPSERVER

Certaines librairies sont installées de base avec le système Xubuntu. Il sera souvent utile de vérifier si celles-ci sont présentes ou pas. Pour cela :

On utilisera le gestionnaire de paquets *Synaptic* et son interface graphique que l'on retrouve dans **Menu des applications>Système>Gestionnaire de paquets Synaptic**.

Si non installer, installer le avec menu->ubuntu software center

Cliquer Recharger en haut à gauche :



3.1.0 INSTALLATION DES COMPILATEURS C/C++

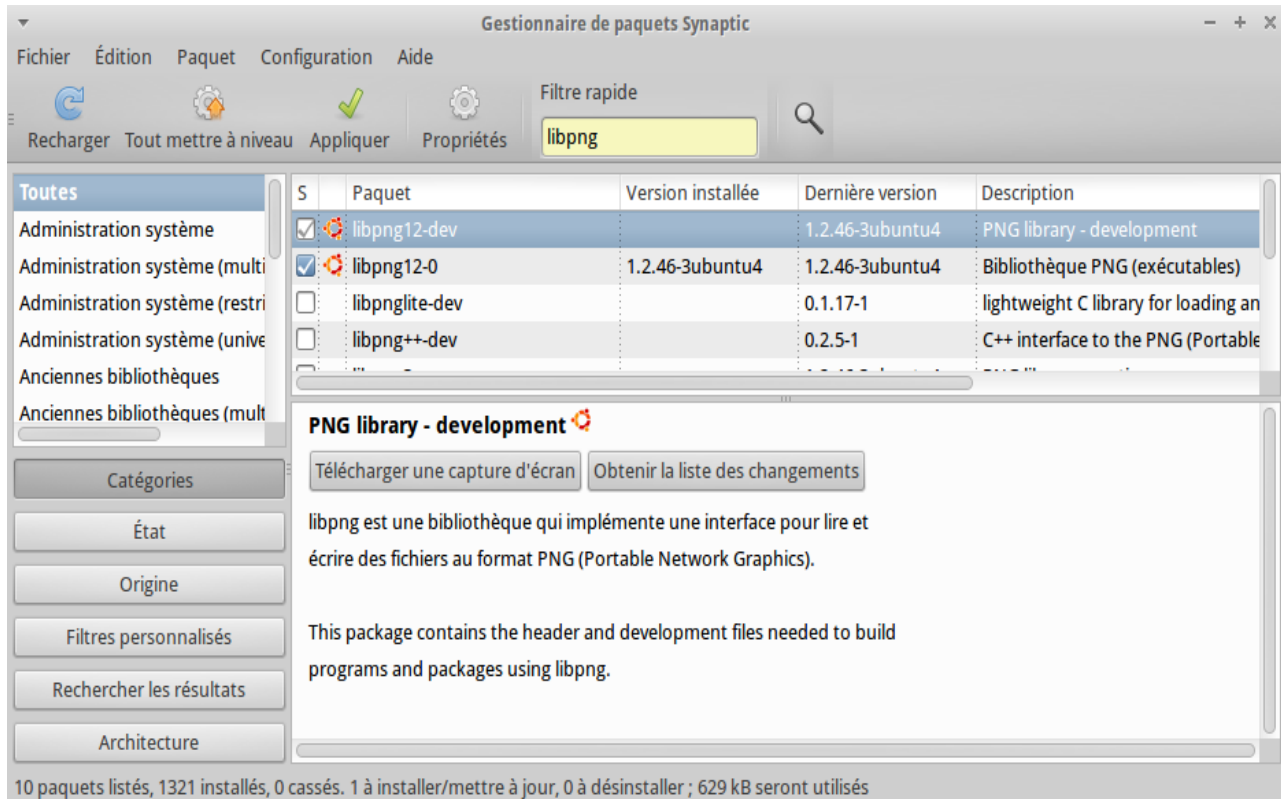
Dans le gestionnaire de paquets, entrez dans le filtre rapide gcc et installer le.

Ensuite, entrez g++ et installer le. Il est fort possible que le compilateur C++ (g++) ne soit pas installer, il sera nécessaire pour l'installation de MapServer plus tard.

3.1.1 INSTALLATION DE LIBPNG

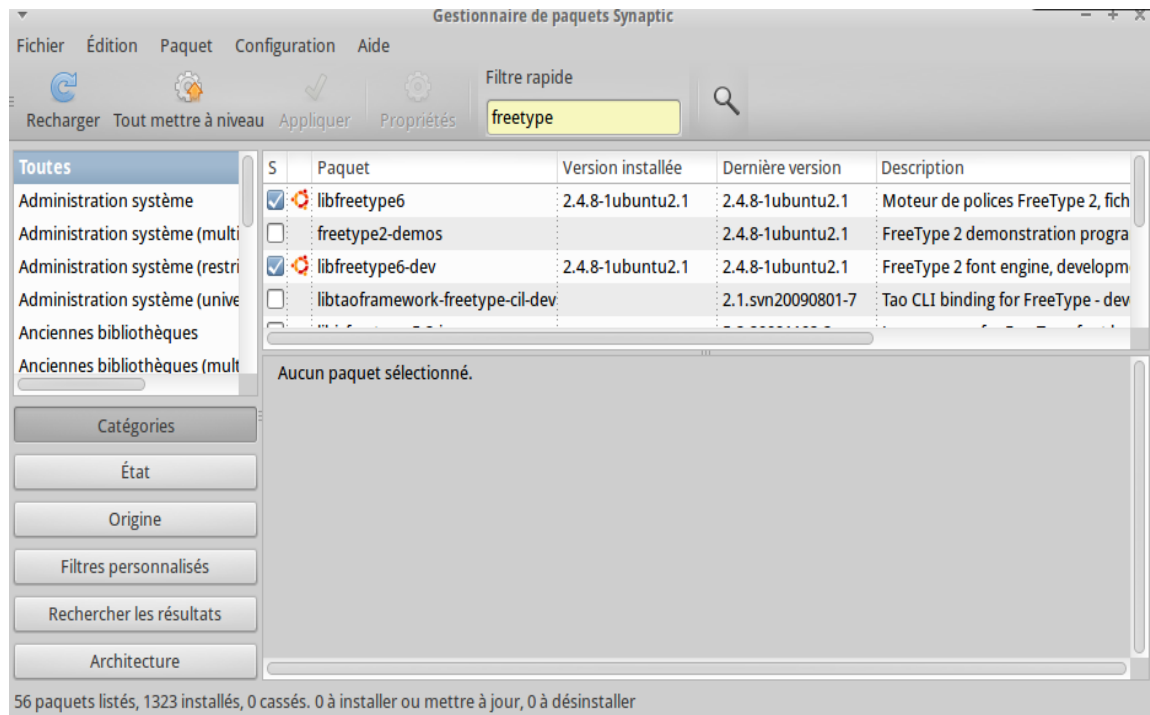
La librairie est déjà installée de base sur Ubuntu mais l'installer est toujours possible :

Tapez libpng dans le filtre rapide et sélectionner libpng12-0 et libpng12-dev et cliquer sur appliquer.



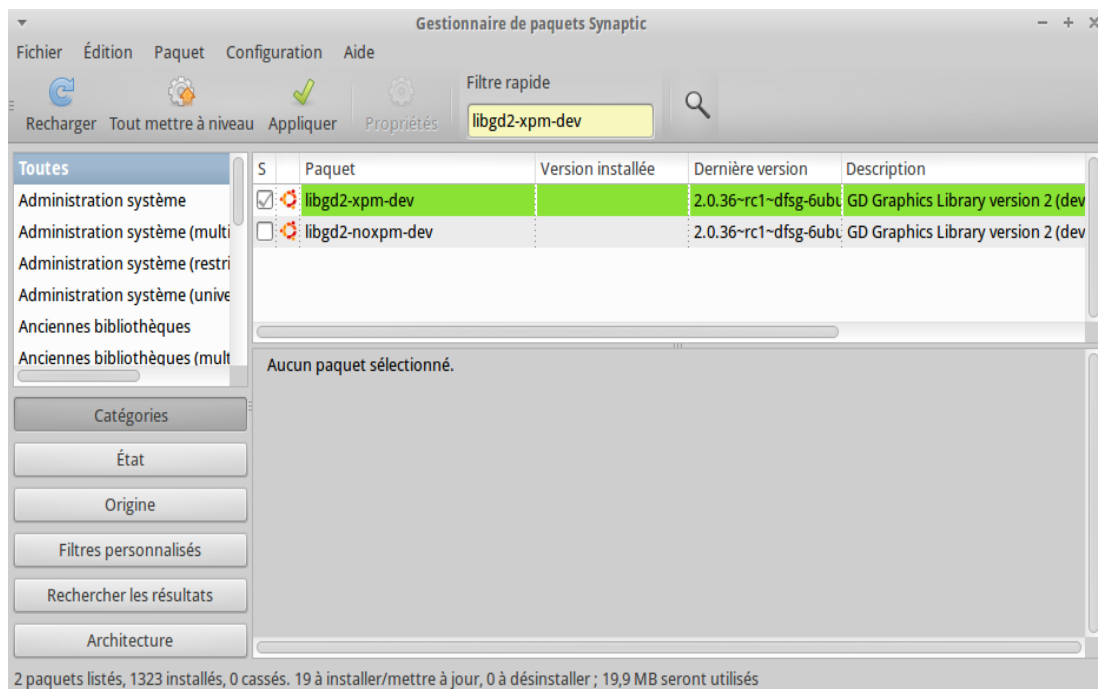
3.1.2 INSTALLATION DE FREETYPE

De même que *libpng*, **freetype** est déjà installée sur Ubuntu. Vérifiez que la version soit **supérieure à la version 2.0**. Pour l'installer, Tapez freetype et sélectionner en plus libfreetype6-dev :



3.1.4 INSTALLATION DE LIBGD

La version **2.0.34** ou supérieure est requise car nous utiliserons la fonction d'anti aliasing sur les labels de la carte. Pour installer **libgd**, Tapez dans le filtre rapide libgd2-xpm-dev :



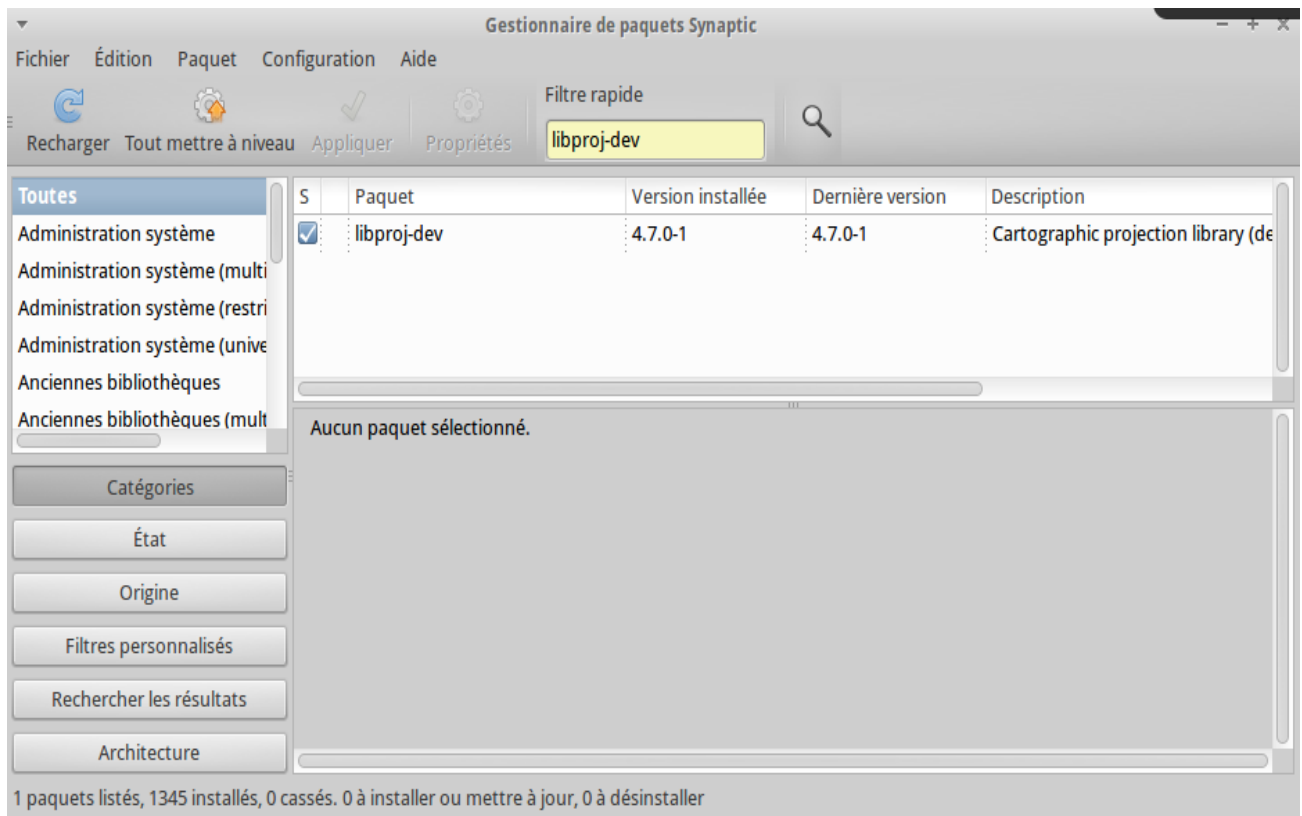
Après avoir cliqué sur appliquer, n'hésitez pas à ajouter à la sélection les paquetages proposés.

3.1.5 INSTALLATION DE ZLIB

zlib1g et zlib1g-dev est installé par défaut sur Ubuntu. Vous pouvez vérifier via le filtre rapide.

3.1.6 INSTALLATION DE LIBPROJ

Utilisez le filtre rapide et tapez libproj-dev. Sélectionnez ce dernier et n'hésitez pas à ajouter les paquetages proposés.

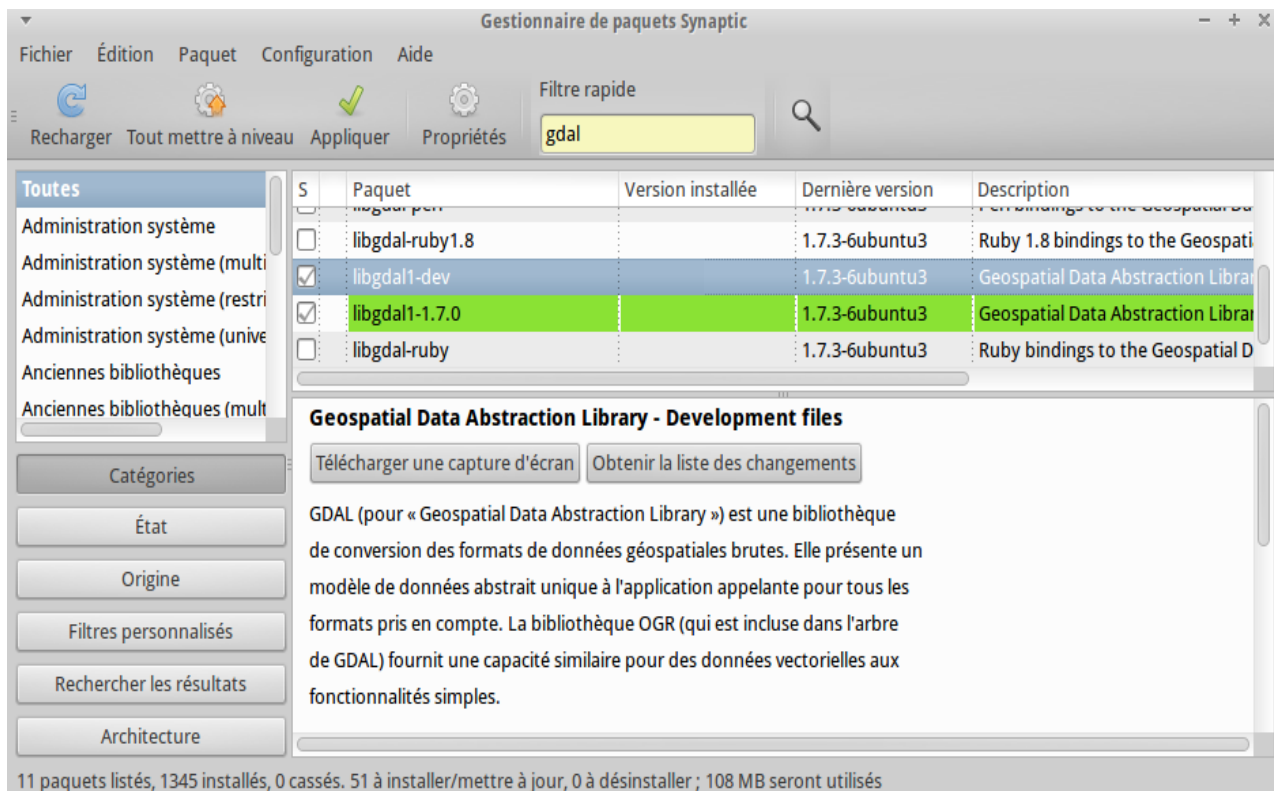


3.1.7 INSTALLATION DE LIBCURL

La version de base (libcurl3 à mettre à jour) suffira mais tenter néanmoins de la mettre à jour à partir du site : <http://curl.haxx.se/dlwiz/>

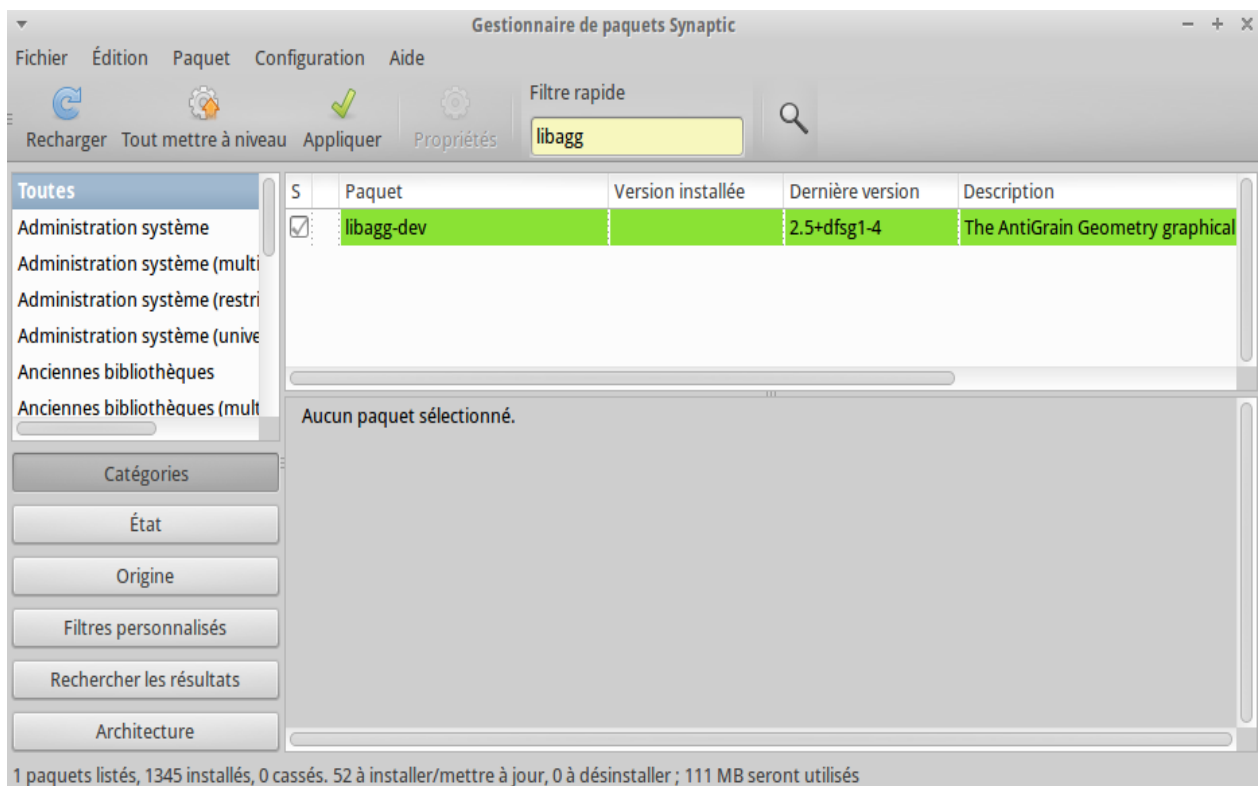
3.1.8 INSTALLATION DE OGR ET GDAL

Toujours dans le gestionnaire de paquets Synaptic, tapez gdal et sélectionnez libgdal1-dev. Ajoutez les paquetages proposés à la sélection.



3.1.9 INSTALLATION DE AGG

Tapez libagg dans le filtre rapide et sélectionnez-le :



3.1.10 INSTALLATION DE GEOS

Tapez libgeos-dev et installez-le avec les paquetages proposés.

3.2 COMPILATION ET INSTALLATION DE MAPSERVER

Téléchargez <http://download.osgeo.org/mapserver/mapserver-6.2.0.tar.gz>

Décompressez le fichier via :

```
#tar -xzf mapserver-6.2.0.tar.gz
```

Placez-vous sur le répertoire créé et entrer :

```
#!/configure --with-ogr=/usr/bin/gdal-config --with-gdal=/usr/bin/gdal-config  
--with-httpd=/usr/sbin/httpd --with-wmsclient --enable-debug --with-curl-  
config=/usr/bin/curl --with-proj=/usr/ --with-tiff --with-gd=/usr/bin/gdlib-  
config --with-jpeg --with-threads --with-wcs --with-freetype=/usr/ --with-  
zlib --with-libiconv=/usr/ --with-sos --with-geos=/usr/bin/geos-config
```

Si des erreurs apparaissent, c'est que certaines librairies manquent à l'installation. Dans ce cas, il est nécessaire de reproduire toutes les instructions de la section précédente en privilégiant cette fois ci les codes sources que l'on peut obtenir sur les sites officiels.

Ne passer surtout pas à la commande suivante si des erreurs persistent.

```
#make
```

Et continuer avec :

```
#sudo make install
```

Copiez l'exécutable pour le greffer à Apache comme ce qui suit :

```
#sudo cp /usr/local/bin/mapserv /usr/lib/cgi-bin/
```

Faire un test du serveur dans le navigateur en entrant dans la barre d'adresse :

<http://localhost/cgi-bin/mapserv>

Si tout se passe correctement, vous devez avoir le message suivant :

"No query information to decode. QUERY_STRING is set, but empty."

Si le serveur vous renvoie :

"403 Permission denied"

Faire :

```
#chmod o+x /usr/lib/cgi-bin/mapserv
```

Référence : <http://www.mapserver.org/installation/unix.html>

3.3 INSTALLATION DES CARTES HORS-LIGNE

Copiez le contenu du répertoire `/var/www/<nomDossierRacineSource>/data/linux/` dans le répertoire `/usr/lib/cgi-bin/`:

```
#sudo cp -r /var/www/<nomDossierRacineSource>/data/linux/* /usr/lib/cgi-bin/
```

Copiez le contenu du répertoire `/var/www/<nomDossierRacineSource>/data/map/` dans le répertoire `/var/www/cgi-bin/`

```
#sudo cp -r /var/www/<nomDossierRacineSource>/data/map/* /usr/lib/cgi-bin/
```

Pour n'afficher que les cartes de notre serveur local de carte, Ouvrez le fichier `/var/www/<nomDossierRacineSource>/config/bootstrap.php` et donnez à la variable de configuration **'internet'** la valeur 'non' :

```
configure::write('internet','non');
```

3.4 CHANGER LES DROITS du dossier `/usr/lib/cgi-bin/`

Dans le dossier `/usr/lib/cgi-bin/`

```
chmod -Rf 775 *
```