

# Installation d'un serveur de tuiles

## Création des dossiers pour les sources

- cd home/pi/
- mkdir src

## Installation de Nodejs

- sudo apt-get install g++ curl git-core curl build-essential openssl libssl-dev
- cd /home/pi/Downloads
- wget https://nodejs.org/dist/v6.2.0/node-v6.2.0.tar.gz
- cd /home/pi/src
- tar xzvf ../Downloads/node-v6.2.0.tar.gz
- cd node-v6.2.0
- ./configure
- make
- sudo make install

## Installation de osm2pgsql

- sudo apt-get install make cmake g++ libboost-dev libboost-system-dev \ libboost-filesystem-dev libexpat1-dev zlib1g-dev \ libbz2-dev libpq-dev libgeos-dev libgeos++-dev libproj-dev lua5.2 \ liblua5.2-dev
- cd home/pi/src/
- git clone https://github.com/openstreetmap/osm2pgsql.git
- cd osm2pgsql
- mkdir build && cd build
- cmake ..
- make
- sudo make install

## Installation de Postgresql 9.4 & Postgis 2.1 + Configuration

- sudo apt-get install postgresql postgis
- su - postgres
- createuser osm
- createdb -E UTF8 -O osm osm
- createlang plpgsql osm
- psql osm
  - ALTER USER postgres PASSWORD '<password>';
  - CREATE EXTENSION postgis; CREATE EXTENSION hstore;
  - ALTER TABLE geometry\_columns OWNER TO osm; ALTER TABLE spatial\_ref\_sys OWNER TO osm;
  - \q

## Télécharger les données OSM et les charger dans Postgresql

- `cd /home/pi/Download`
- `wget http://download.geofabrik.de/europe/france/languedoc-roussillon-latest.osm.pbf`
- Importer les données initiales :
  - Fichier de grande taille : `osm2pgsql --create --database osm --username osm --prefix osm -E 3857 --slim --style ../src/osm2pgsql/default.style --hstore --cache 512 languedoc-roussillon-latest.osm.pbf`
  - Fichier de petite taille : `osm2pgsql --create --database osm languedoc-roussillon-latest.osm.pbf`
- Ajouter de nouvelles données :
  - Fichier de grande taille : `osm2pgsql -l --append --database osm --username osm --prefix osm -E 3857 --slim --style ../src/osm2pgsql/default.style --hstore --cache 512 midi-pyrenees-latest.osm.pbf`

## Ajouter des index à la base osm

- `psql osm`
- Puis exécuter les requêtes suivantes :

```
CREATE INDEX osm_point_tags ON osm_point USING GIN(tags);
CREATE INDEX osm_line_tags ON osm_line USING GIN(tags);
CREATE INDEX osm_polygon_tags ON osm_polygon USING GIN(tags);
```

## Solution basée sur CartoCSS & Mapnik

Fonctionne ! Voir : [https://medspx.fr/kb/GIS/generate\\_osm/](https://medspx.fr/kb/GIS/generate_osm/)

- `sudo aptitude install gdal-bin mapnik-utils node-carto python-shapely`
- `cd ~/src`
- `git clone https://github.com/gravitystorm/openstreetmap-carto`
  - Alternative (non testé [2016-06-22]) correspondant au style osm-fr (<http://tile.openstreetmap.fr/>): `git clone https://github.com/cquest/osmfr-cartocss.git`
- `cd openstreetmap-carto`
  - `wget http://svn.openstreetmap.org/applications/rendering/mapnik/generate_tiles_multiprocess.py`
  - `chmod -x generate_tiles_multiprocess.py`
  - `./get-shapefiles.sh`
  - `sed -i '/"dbname": "gis"/c \\t"dbname": "osm"' project.mml`
  - `sed -i '/"dbname": "osm"/ i \\t"user": "osm", \\n\\t"host": "localhost",' project.mml`
- Installer *carto*, le parseur de CartoCSS :

- Utiliser node v4.4.x : `nvm use 4.4`
  - Installer `carto` : `npm install carto`
  - Compiler le projet : `./node_modules/.bin/carto project.mml > mapnik.xml`
  - Installer les Polices :
    - `sudo apt-get install ttf-dejavu fonts-droid ttf-unifont fonts-sipa-arundina fonts-sil-padauk fonts-khmeros fonts-taml-tscu fonts-indic fonts-knda`
    - S'il reste des polices non supportée, supprimer les lignes qui pausent problème dans le fichier `mapnik.xml`
  - Aller sur <http://www.openstreetmap.org/> :
    - Afficher la zone correspondant à vos données puis cliquer sur "Exporter", les coordonnées max de la zone (bbox) vont alors s'afficher.
1. Pour plus de facilité, copier le lien *Api Overpass*, qui contient la bbox (**-0.027,42.155,8.597,45.325**) :  
<http://overpass-api.de/api/map?bbox=-0.027,42.155,8.597,45.325>
  2. Extraire dans l'url le zoom minimum (**8**) :  
<http://www.openstreetmap.org/export#map=8/43.761/4.285>
    - Zoomer au maximum voulu puis extraire dans l'URL le zoom max (**17**) :  
<http://www.openstreetmap.org/export#map=17/44.34644/3.95358>
- Éditer le fichier `generate_tiles_multiprocess.py` : `vi generate_tiles_multiprocess.py`
    - Remplacer les dernières lignes par le contenu ci-dessous où l'on aura remplacer les valeurs `mapfile`, `tile_dir`, `minZoom`, `maxZoom` et `bbox` par celle relevées sur Openstreetmap.
    - Donner les droits d'exécution au script : `chmod +x generate_tiles_multiprocess.py`
  - Lancer le script : `./generate_tiles_multiprocess.py&`
    - Laisser tourner le script pour générer l'ensemble des tuiles. Cela peut prendre vraiment très longtemps, suivant l'étendue des de la zone à couvrir et le zoom maximum demandé.

### Dernière ligne du fichier `generate_tiles_multiprocess.py` :

```
if __name__ == "__main__":

    home = os.environ['HOME']
    try:
        mapfile = os.environ['MAPNIK_MAP_FILE']
    except KeyError:
        mapfile = home + "/src/openstreetmap-carto/mapnik.xml"
    try:
        tile_dir = os.environ['MAPNIK_TILE_DIR']
    except KeyError:
        tile_dir = home + "/web/tiles/bitmap/mapnik/"

    if not tile_dir.endswith('/'):
        tile_dir = tile_dir + '/'

    #-----
    --
    #
    # Change the following for different bounding boxes and zoom levels
```

```
minZoom = 8
maxZoom = 17
bbox = (-0.027, 42.155, 8.597, 45.325)
render_tiles(bbox, mapfile, tile_dir, minZoom, maxZoom)
```

## Solution basée sur MapCSS & node-tileserv & kothicjs

Non finalisé, problème avec node-tileserv...

### Installer le serveur de tuiles

- `sudo apt-get install libcairo2-dev libjpeg-dev libgif-dev`
- `cd /home/pi/`
- `mkdir workspace/ web/`
- `git clone https://github.com/rurseekatze/node-tileserv.git`
- `cd node-tileserv`
- `npm install`
- `cd /home/pi`
- `ln -s ~/workspace/node-tileserv ~/web/node-tileserv`

### Installer le parser MapCSS

- `cd /home/pi/workspace`
- `git clone https://github.com/Miroff/mapcss-parser.git`
- `cd mapcss-parser`
- `sudo pip install ply`
- `sudo python setup.py install`

### Installer les styles MapCSS

- `cd /home/pi/workspace`
- `git clone https://github.com/kothic/kothic-js-mapcss.git`
- `cd kothic-js-mapcss`
- `mkdir -p debug/styles/`
- `sudo apt-get install python-rsvg`
- `sudo pip2 install PIL`
- `sudo chmod 777 /usr/local/lib/python2.7/dist-packages/mapcss_parser/`
- `vi scripts/mapcss_converter.py`
  - Remplacer `import Image` par `from PIL import Image`
- `vi Makefile`
  - Remplacer `python` par `python2`
- Lancer la compilation des styles : `make all`
- `mkdir ../../web/styles`
- `cp debug/styles/*.* ../../web/styles`

## Gpsd, Nodejs et Leaflet

- Nodejs module faisant le pont avec Gpsd : <https://github.com/pdeschen/bancroft>

From:

<https://memos.clapas.org/> - **Memos**

Permanent link:

<https://memos.clapas.org/rpi/config/tile-server?rev=1582659633>

Last update: **2020/02/25 19:40**

