

La messa in “produzione” per i test sul server fornito da Wikimedia Italia ha seguito la procedura classica con l’installazione di PostgreSQL e PostGIS e tutti i software necessari, l’importazione dei dati di OpenStreetMap utilizzando `osm2pgsql` e scaricando i dati necessari per i due stili. Per questa parte le istruzioni sono disponibili all’interno dei rispettivi stili<sup>1 2</sup>.

Il server utilizza `mod_tile` e `renderd`, due software sviluppati proprio per renderizzare le tile raster sul sito [www.openstreetmap.org](http://www.openstreetmap.org).

I software scelti sono facilmente installabili e configurabili seguendo la guida messa a disposizione dagli sviluppatori<sup>3</sup> (utilizzando un sistema operativo Debian/Ubuntu recente è possibile installarli direttamente dal software per l’installazione dei pacchetti). Le uniche differenze apportate sono i permessi delle directory `/run/renderd` e `/var/cache/renderd/` che sono stati assegnati all’utente (attualmente il mio) che esegue `renderd` (che non può essere root ma l’utente che ha esportato gli stili per alcuni problemi di permessi sui raster utilizzati nello stile hiking non del tutto chiariti)

Il file importante che è stato modificato è `/etc/renderd.conf` che serve per configurare `renderd` e indicare dove si trovano i file xml di Mapnik con le istruzioni del vestimento dei dati.ù

Attualmente `renderd` viene lanciato direttamente dall’utente in background, è inoltre possibile utilizzare un sistema per il controllo dei servizi basato su `init.d` basandosi su questo script<sup>4</sup>.

Una guida completa può essere visualizzata sul sito `switch2osm`<sup>5</sup> e la procedura seguita può essere visualizzata nella pagina che segue.

Esistono altri strumenti per il rendering per esempio `tirex`<sup>6</sup> e `tessera`<sup>7</sup>, questi possono essere testati prima di andare in un sistema di produzione finale per capire se vi sono migliori prestazioni o capacità di gestire meglio lo spazio disco.

---

1 <https://github.com/osmItalia/openstreetmap-carto/blob/master/INSTALL.md>

2 <https://github.com/osmItalia/openstreetmap-hiking/blob/master/INSTALL.md>

3 [https://github.com/openstreetmap/mod\\_tile/blob/master/docs/build/building\\_on\\_ubuntu\\_20\\_04.md](https://github.com/openstreetmap/mod_tile/blob/master/docs/build/building_on_ubuntu_20_04.md)

4 [https://github.com/SomeoneElseOSM/mod\\_tile/blob/master/debian/renderd.init](https://github.com/SomeoneElseOSM/mod_tile/blob/master/debian/renderd.init)

5 <https://switch2osm.github.io/serving-tiles/>

6 <https://github.com/openstreetmap/tirex>

7 <https://github.com/mojodna/tessera>

```

### install needed packages
sudo apt install apache2 postgres postgresql python3-psycopg2 python3-yaml python3-requests gdal-
bin screen postgresql-11-postgis-2.5 postgresql-11-postgis-2.5-scripts npm
sudo apt install build-essential autoconf apache2-dev libcairo2-dev libcurl4-gnutls-dev libglib2.0-
dev libiniparser-dev libmapnik-dev libmemcached-dev librados-dev
sudo apt install fonts-noto-cjk fonts-noto-hinted fonts-noto-unhinted fonts-hanazono ttf-unifont
fonts-dejavu-core
sudo apt-get install make cmake g++ libboost-dev libboost-system-dev cmake-curses-gui libboost-
filesystem-dev libexpat1-dev zlib1g-dev libbz2-dev libpq-dev libproj-dev lua5.3 liblua5.3-dev
pandoc libosmium2-dev osmium libprotozero-dev
# install carto from node
sudo npm install -g carto

### compile osm2pgsql since newer version has better support for flex output
git clone https://github.com/openstreetmap/osm2pgsql
cd osm2pgsql/
mkdir build
cd build/
ccmake ..
make -j4
sudo make install

### create the database
sudo su postgres
createdb gis
psql -c "create extension postgis" gis
psql -c "create extension hstore" gis

### clone openstreetmap styles and compile xml file
git clone https://github.com/osmItalia/openstreetmap-carto.git
cd openstreetmap-carto/
git checkout --track origin/name_it
carto project.mml > osmitalia.xml
cd ..
git clone https://github.com/osmItalia/openstreetmap-hiking.git
cd openstreetmap-hiking/
carto project.mml > hiking.xml
cd ..

### download and import data
wget -c https://osmit-estratti.wmcloud.org/dati/poly/regioni/pbf/04_Trentino-Alto%20Adige-S
%C3%BCdtirol_poly.osm.pbf
cd openstreetmap-carto/
osm2pgsql -G --hstore --style openstreetmap-carto.style --tag-transform-script openstreetmap-
carto.lua -d gis ~/04_Trentino-Alto\ Adige-Südtirol_poly.osm.pbf
psql -d gis -f indexes.sql
python scripts/get-external-data.py
cd ../openstreetmap-hiking/
psql -f scripts/db_function.sql gis
osm2pgsql -d gis --output=flex -S scripts/route.lua ~/04_Trentino-Alto\ Adige-
Südtirol_poly.osm.pbf
# this should be executed all the times data are imported
psql -f scripts/indexes.sql gis

```

```
# copy raster data
cp ~/*.tif layers
# import contour
shp2pgsql -g wkb_geometry ~/contour.shp contour | psql gis

### compile, install and configure mod_tile since it is not yet packages in
debian 10
git clone https://github.com/openstreetmap/mod_tile.git
cd mod_tile/
./autogen.sh
./configure
make -j4
sudo mkdir --parents /run/renderd /var/cache/renderd/tiles
sudo chown -R lucadelu:wikidev /run/renderd /var/cache/renderd
sudo cp "etc/renderd/renderd.conf.examples" /etc/renderd.conf
sudo cp "etc/apache2/renderd.conf" /etc/apache2/conf.d/renderd.conf
sudo cp utils/example-map/index.html /var/www/html/osmitalia.html
echo "LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so" | sudo tee --append
/etc/apache2/mods-enabled/mod_tile.load
sudo make install
sudo make install-mod_tile
# add the two styles pointing to the xml file created with carto
sudo vim /etc/renderd.conf
sudo systemctl --now enable apache2
sudo systemctl restart apache2
# execute renderd from the user
renderd -f

### prepare the two pages to view the map
cd /var/www/html/
sudo wget -c http://cdn.leafletjs.com/leaflet/v1.7.1/leaflet.zip
sudo mkdir leaflet
cd leaflet/
sudo unzip ../leaflet.zip
sudo vim osmitalia.html
sudo cp osmitalia.html osmhiking.html
vim osmhiking.html
```