

---

# **openbasiskaart Documentation**

*Release 1.0.0*

**OpenGeoGroep U.A.**

**Nov 30, 2018**



---

## Contents

---

<b>1</b>	<b>Algemeen</b>	<b>3</b>
1.1	Waarom OpenBasisKaart ? . . . . .	3
1.2	Specificaties . . . . .	4
1.3	Drie Toolchains . . . . .	4
1.4	And The Winner Is... . . . .	5
1.5	Data . . . . .	5
<b>2</b>	<b>Mapnik Toolchain</b>	<b>7</b>
2.1	Installaties . . . . .	7
2.2	Data . . . . .	10
2.3	Services . . . . .	11
2.4	Monitoring . . . . .	12
2.5	Demo . . . . .	14
2.6	Tiles in EPSG:28992 . . . . .	14
<b>3</b>	<b>Mapcache Toolchain</b>	<b>23</b>
3.1	Installaties . . . . .	23
3.2	Data . . . . .	25
3.3	Service . . . . .	25
3.4	Tiling . . . . .	27
3.5	Seeding . . . . .	29
3.6	Andere stylen . . . . .	30
3.7	Nog te doen . . . . .	30
<b>4</b>	<b>Ontwikkelomgeving</b>	<b>31</b>
4.1	Installaties . . . . .	31
4.2	Werken met Vagrant Boxes . . . . .	32
<b>5</b>	<b>Links</b>	<b>35</b>
<b>6</b>	<b>Indices and tables</b>	<b>37</b>
	<b>Bibliography</b>	<b>39</b>



Contents:



Hieronder staat algemene informatie over het hoe en waarom van de OpenGeoGroep (OGG) OpenBasiskaart Nederland (OBKN). Zie ook de website: <http://openbasiskaart.nl>. Deze documentatie wordt automatisch gegenereerd vanuit het bijbehorende GitHub project: <https://github.com/opengeogroep/openbasiskaart> en heeft de nadruk op de gebruikte techniek.

## 1.1 Waarom OpenBasisKaart ?

Vanuit zowel OGG-klanten als de OGG-leden is vaker de wens uitgesproken om OpenStreetMap in de Nederlandse (RD-)projectie beschikbaar te maken.

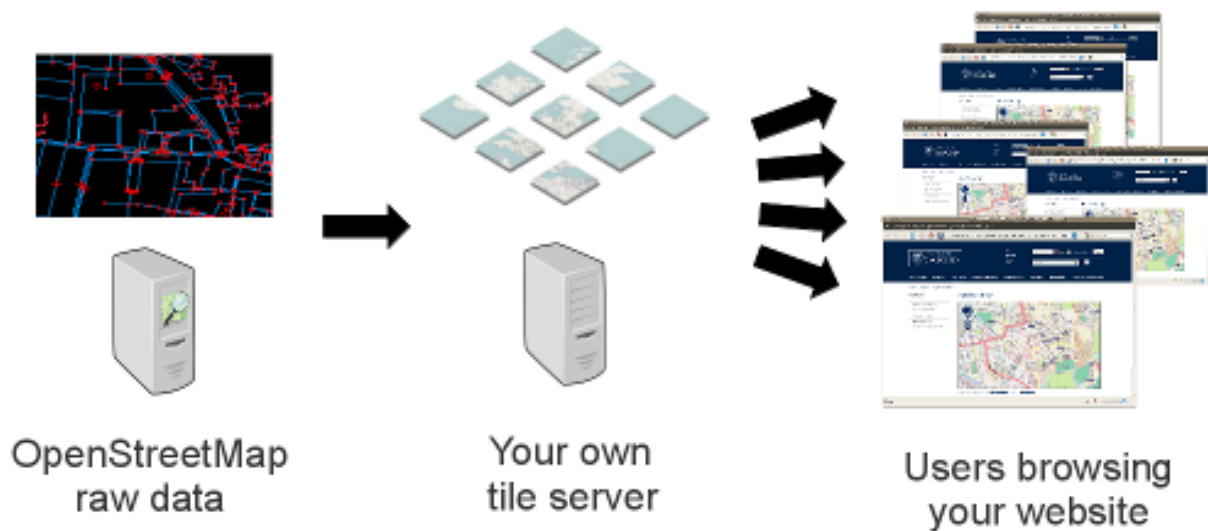


Fig. 1: *Figuur 1 - Overzicht (bron: <http://switch2osm.org/serving-tiles>)*

## 1.2 Specificaties

Services: zowel TMS als WMTS, evt WMS-C. WMS heeft niet primair de focus, wel wordt deze in de eerste en derde variant geboden.

Actualiteit: incrementele updates vanuit main OSM DB

Visualisaties: meerdere, gemakkelijk uitbreidbaar

URL: <http://openbasiskaart.nl>

Projecties: Voor de eerste twee varianten alleen RD, voor de derde variant wordt gewerkt aan ondersteuning van de de service standaarden voor WMS zoals genoemd op <http://www.geonovum.nl/geostandaarden/services/destandaarden>

Kaartlagen: voorlopig OSM, later eventueel ook Top10NL en waar mogelijk versterkt met de basisregistraties zoals BAG en BRT

## 1.3 Drie Toolchains

Er zijn meerdere mogelijkheden om vanuit een OSM Planet file uiteindelijk tot tiling en/of OGC webdiensten te komen. We hebben drie hoofdvarianten uitgezocht, bijgenaamd de “Mapcache Toolchain”, “Mapnik Toolchain” en “GeoServer Toolchain”.

In de eerste twee gevallen zal tiling via TMS en WMTS geleverd dienen te worden. In het derde geval zal tiling beschikbaar worden gemaakt via WMS-C, TMS en WMTS en zal ook WMS en KML beschikbaar komen. Via MapProxy (1e variant) is ook WMS beschikbaar, zij het met de tilecache als bron.

Ook willen we onderzoeken in hoeverre we in de eerste twee varianten MBTiles als opslag kunnen gebruiken.

### 1.3.1 Mapnik Toolchain

Deze gaat uit van:

- osm2pgsql
- Mapnik
- MBTiles (ipv file system)
- MapProxy (ipv mod\_tile)
- Osmosis (up to date houden tiles)

### 1.3.2 MapCache Toolchain

Deze gaat uit van:

- Imposm (importeren OSM Planet data)
- MapServer WMS
- MapCache tiling server
- Tiles op filesysteem
- Styling via MapServer Basemaps: <https://github.com/opengeogroep/basemaps>



### 1.3.3 GeoServer Toolchain

Deze gaat uit van:

- osm2pgsql
- osmosis
- GeoServer
- MapProxy

### 1.3.4 Optimalisaties

<http://www.remote.org/frederik/tmp/ramm-osm2pgsql-sotm-2012.pdf> (gebruikt ook Hetzner!)

## 1.4 And The Winner Is...

Uiteindelijk hebben we gekozen voor de MapCache toolchain. De redenen zijn deels rationeel, deels tijdgebrek om de andere chains volledig uit te werken. De GeoServer toolchain heeft als nadelen: Java (voor sommigen), beperking en verbosity SLDs. De Mapnik Toolchain was op zich aardig, er waren wat issues met MBTiles in andere projectie en bleek bovendien redelijk traag t.o.v. filestore, alhoewel natuurlijk een filestore uitstekend gebruikt kan worden. De MapCache toolchain was gewoon ook het beste uitgewerkt. Styling via MapServer bleek minder een issue dan gedacht vanwege de beschikbaarheid van MS basemap styles: <https://github.com/opengeogroep/basemaps>. Ook zijn een aantal componenten in deze MapCache toolchain goed te integreren/automatiseren, bijv. reseeding gedreven door OGR sources. De integratie met Apache is ook aantrekkelijk.

## 1.5 Data

De proeven met de eerste twee varianten worden uitgevoerd met een beperkt gebied; Amsterdam e.o. gedownload van: <http://metro.teczno.com/#amsterdam>

De derde variant bevat momenteel al een OpenStreetMap laag voor Nederland, België, Luxemburg en de gehele BeNeLux in de WGS84 en de “Spherical Mercator” of “Google” projectie



---

### Mapnik Toolchain

---

Hieronder staat de “toolchain” beschreven om OSM RD Tiles te genereren en te serveren volgens de “Mapnik Toolchain”. Dit is de meest standaard OpenStreetMap methode zoals ook gebruikt voor de tiles beschikbaar op <http://openstreetmap.org>, a.k.a. “The Slippy Map”.

Om in RD-stelsel te werken moet nog iets speciaals gedaan worden. Zie ook: <http://justobjects.org/blog/2010/openstreetmap-tiles-for-dutch-projection-epsg28992> ;-)

Normaal gesproken wordt “mod-tile” met `renderd` als tileserver/generator gebruikt. Echter dan zijn we beperkt tot TMS. Als variant op de toolchain proberen we `MapProxy` via `MBTiles` opslag.

We gaan in de volgende stappen te werk:

1. standaard installatie PostGIS/Mapnik/renderd/mod\_tile met tiles in EPSG:900913
2. vervang renderd/mod\_tile door MapProxy+MBTiles opslag
3. als 2. maar met tiles in EPSG:28992
4. onderzoek: meerdere styles (via TileMill)
5. onderzoek: automatisch updaten
6. heel NL tilen

*NB deze toolchain wordt niet gebruikt in de OpenBasisKaart maar is handig ter referentie.*

## 2.1 Installaties

Hieronder de stappen voor installatie van de verschillende tools.

### 2.1.1 Ubuntu

We gaan uit van Ubuntu 12.10 64-bits. Deze moet altijd eerst uptodate gebracht worden.

```
sudo apt-get update
sudo apt-get upgrade
```

## 2.1.2 Repositories

Ubuntu bevat vaak niet laatste versies benodigde packages. Door repositories aan “Apt” toe te voegen kan wel via standaard packages recente versies geïnstalleerd worden. Allereerst evt tool om repo’s toe te voegen (hoeft niet op Ubuntu 12.10).

```
# install the command add-apt-repository if the command can't be found.
sudo apt-get install software-properties-common
```

Dan Kai Krueger’s repo (<https://launchpad.net/~kkrueger/+archive/openstreetmap> Osm2pgsql, Imposm, Osmosis, Mapnik styles etc).

```
# to add the PPA and update your packaging system.
sudo add-apt-repository ppa:kkrueger/openstreetmap
sudo apt-get update
```

Mapnik repo t.b.v. Mapnik 2.1. Zie <https://launchpad.net/~mapnik/+archive/v2.1.0/+packages>.

```
sudo add-apt-repository ppa:mapnik/v2.1.0
sudo apt-get update
```

Altijd UbuntuGIS toevoegen <https://wiki.ubuntu.com/UbuntuGIS> !

```
# to add the UbuntuGIS PPA and update your packaging system.
sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
```

Check of de repo’s goed zijn toegevoegd

```
ls /etc/apt/sources.list.d
# geeft
kkrueger-openstreetmap-quantal.list
kkrueger-openstreetmap-quantal.list.save
mapnik-v2_1_0-quantal.list
ubuntugis-ubuntugis-unstable-quantal.list
ubuntugis-ubuntugis-unstable-quantal.list.save
```

## 2.1.3 Afhankelijkheden

Eerst afhankelijkheden installeren.

```
sudo apt-get install libgdal-dev gdal-bin apache2 apt-show-versions
```

Versies support libs:

- Proj: 4.8.0-3~quantal1
- GDAL: 1.9.2-2~quantal6
- Geos: 3.3.3-1.1

## 2.1.4 Postgresql/PostGIS

Belangrijk is om package “postgis” te installeren. Dan komt alles “mee”, bijv. Postgres 9.1 etc.

```
sudo apt-get install postgis postgresql-contrib postgresql-server-dev-9.1
```

Check of PostGIS v2 is installed.

```
apt-show-versions | grep postgis
# moet geven
postgis/quantal uptodate 2.0.1-2~quantal3
postgresql-9.1-postgis/quantal uptodate 2.0.1-2~quantal3
```

Template database aanmaken. Nieuwe manier voor PostGIS 2.0 met EXTENSIONS (ipv PostGIS sql laden) zie [http://postgis.net/docs/manual-2.0/postgis\\_installation.html#create\\_new\\_db\\_extensions](http://postgis.net/docs/manual-2.0/postgis_installation.html#create_new_db_extensions)

```
sudo -u postgres -i
# aanmaken user "osm" met zelfde password.
# answer yes for superuser (although this isn't strictly necessary)
createuser osm
psql -c "ALTER USER osm WITH PASSWORD 'osm';"
createdb -E UTF8 -O osm postgis2_template
psql -d postgis2_template -c "CREATE EXTENSION postgis;"

# legacy.sql compat layer om problemen met Mapnik 2.0 (niet bestaande functies op_
↳ te lossen)
psql -d postgis2_template -f /usr/share/postgresql/9.1/contrib/postgis-2.0/legacy.sql

# Aanmaken DB met PostGIS template
createdb -E UTF8 -O osm gis -T postgis2_template
```

Inloggen enablen.

```
# Edit the file /etc/postgresql/9.1/main/pg_hba.conf and replace ident by either md5_
↳ or trust,
# depending on whether you want it to ask for a password on your own computer or not.
# Then reload the configuration file with:

/etc/init.d/postgresql reload
```

Handig is phppgadmin. Zie ook <http://sql-info.de/postgresql/notes/installing-phppgadmin.html>

```
sudo apt-get install phppgadmin

# Toelaten inloggen
sudo emacs /usr/share/phppgadmin/conf/config.inc.php
$config['extra_login_security'] = false;

# dan via localhost /phppgadmin benaderen
```

## 2.1.5 OSM2PGSQL

OSM2pgsql wordt gebruikt voor inlezen OSM Planet dump in Postgres. Zie ook <http://wiki.openstreetmap.org/wiki/Osm2pgsql>

```
# install the osm2pgsql package.
sudo apt-get install osm2pgsql
```

Installeert: osm2pgsql (0.81.0-1~quantal3). NB Dit is de juiste versie voor 64-bit ID ondersteuning. Zie <http://web.archiveorange.com/archive/v/wQWib2eq6T9IKbr4XkWx>.

## 2.1.6 Mapnik

Mapnik is voor generatie van tiles. Via eigen repo te installeren. Zelf compileren is verleden tijd! Zie ook <https://github.com/mapnik/mapnik/wiki/UbuntuInstallation> en de packages: <https://launchpad.net/~mapnik/+archive/v2.1.0/+packages> (zie boven)

```
sudo apt-get install libmapnik mapnik-utils python-mapnik
```

Check installatie (libmapnik\_2.1.0-ubuntu1~quantal2\_amd64.deb)

```
python
Python 2.7.3 (default, Sep 26 2012, 21:51:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import mapnik
>>>
```

## 2.1.7 mod\_tile+renderd

Vanuit repo install. Dit installeert/activeert mod\_tile en renderd.

```
sudo apt-get install libapache2-mod-tile
```

Dit download ook automatisch /usr/share/mapnik-osm-data/world\_boundaries-spherical.tgz (50MB) en /usr/share/mapnik-osm-data/processed\_p.tar.bz2 (500MB) en /usr/share/mapnik-osm-data/shoreline\_300.tar.bz2 (40MB).

Notes:

- Output van apt-get bewaard, zie `_static/output-apt-get-mod-tile.txt`
- Toch even checken want hier wordt ook Mapnik (2.0) installed, terwijl we 2.1 hadden via apt-get !

NB bovenstaande wordt dus MBTiles+MapProxy!!

## 2.2 Data

Het laden van de data. Gebied Amsterdam. Zie <http://metro.teczno.com/#amsterdam>

Data ophalen.

```
mkdir /opt/openbasiskaart/data

# PBF download (53 MB)
wget http://osm-metro-extracts.s3.amazonaws.com/amsterdam.osm.pbf

# Coastline A'dam area download (is leeg, dus niet gebruiken!!)
# wget http://osm-metro-extracts.s3.amazonaws.com/amsterdam.coastline.zip
```



Fig. 1: *Figuur MT-1 - Amsterdam Extent* (bron: <http://metro.teczno.com/#amsterdam>)

Data laden in PostgreSQL.

```
cd /opt/openbasiskaart/data

# Op locale VirtualBox VM met weinig geheugen
# met "--cache-strategy sparse"
osm2pgsql -W -U osm -d gis --slim --cache-strategy sparse amsterdam.osm.pbf

# duurt plm 900 sec op VM
```

## 2.3 Services

Mapnik en mod\_tile/renderd met eigen configuratie.

De config van renderd in `/etc/renderd.conf`, is voorlopig Mapnik 2.0, maar mogelijk later proberen met Mapnik 2.1

```
[renderd]
stats_file=/var/run/renderd/renderd.stats
socketname=/var/run/renderd/renderd.sock
num_threads=4
tile_dir=/var/lib/mod_tile

[mapnik]
plugins_dir=/usr/lib/mapnik/2.0/input
font_dir=/usr/share/fonts/truetype/ttf-dejavu
font_dir_recurse=false

[default]
URI=/osm/
XML=/opt/openbasiskaart/mapnik/default/osm.xml
DESCRIPTION=This is the standard osm mapnik style
;ATTRIBUTEION=&copy;;<a href="\http://www.openstreetmap.org/">OpenStreetMap</a> and <a
↪ href="\http://wiki.openstreetmap.org/w\
iki/Contributors\">contributors</a>, <a href="\http://creativecommons.org/licenses/by-
↪ sa/2.0/">CC-BY-SA</a>
```

(continues on next page)

(continued from previous page)

```
;HOST=tile.openstreetmap.org
;SERVER_ALIAS=http://a.tile.openstreetmap.org
;SERVER_ALIAS=http://b.tile.openstreetmap.org
;HTCPHOST=proxy.openstreetmap.org
```

### Configureren Renderd/Mapnik/mod\_tile.

```
# Maak kopie default mapnik config
mkdir /opt/openbasiskaart/mapnik
cp -r /etc/mapnik-osm-data /opt/openbasiskaart/mapnik/default
cd /opt/openbasiskaart/mapnik/default

# zet user/password naar osm/osm in
e inc/datasource-settings.xml.inc

<Parameter name="type">postgis</Parameter>
<Parameter name="password">osm</Parameter>
<Parameter name="host">localhost</Parameter>
<Parameter name="user">osm</Parameter>
<Parameter name="dbname">gis</Parameter>
<!-- this should be 'false' if you are manually providing the 'extent' -->
<Parameter name="estimate_extent">>false</Parameter>
<!-- manually provided extent in epsg 900913 for whole globe -->
<!-- providing this speeds up Mapnik database queries -->
<!-- <Parameter name="extent">4.88,52.36,4.90,52.38</Parameter> -->
<Parameter name="extent">543239.115,6865481.657,545465.505,6869128.129</Parameter>

# herstarten en log volgen renderd
tail -f /var/log/syslog |grep renderd &
/etc/init.d/renderd restart
```

#### Notes:

- **Mapnik 2.0 met PosGIS 2.0: legacy.sql laden in PostGIS DB**
  - `psql -d gis -f /usr/share/postgresql/9.1/contrib/postgis-2.0/legacy.sql`
- **extent**
  - moet in EPSG:900913
  - extent gezet op klein stukje A'dam C voor testen
- **files verwijderen/opschonen**
  - `rm -rf /var/lib/mod_tile/default`
  - `touch /var/lib/mod_tile/planet-import-complete`
- herstarten renderd: `/etc/init.d/renderd restart`

## 2.4 Monitoring

Munin is een flexibele monitoring tool, zie: <http://munin-monitoring.org>.

Installeren.



```
sudo apt-get install munin-node munin
```

Enablen voor andere hosts via `/etc/apache2/conf.d/munin`.

Via browser raadplegen, zie:

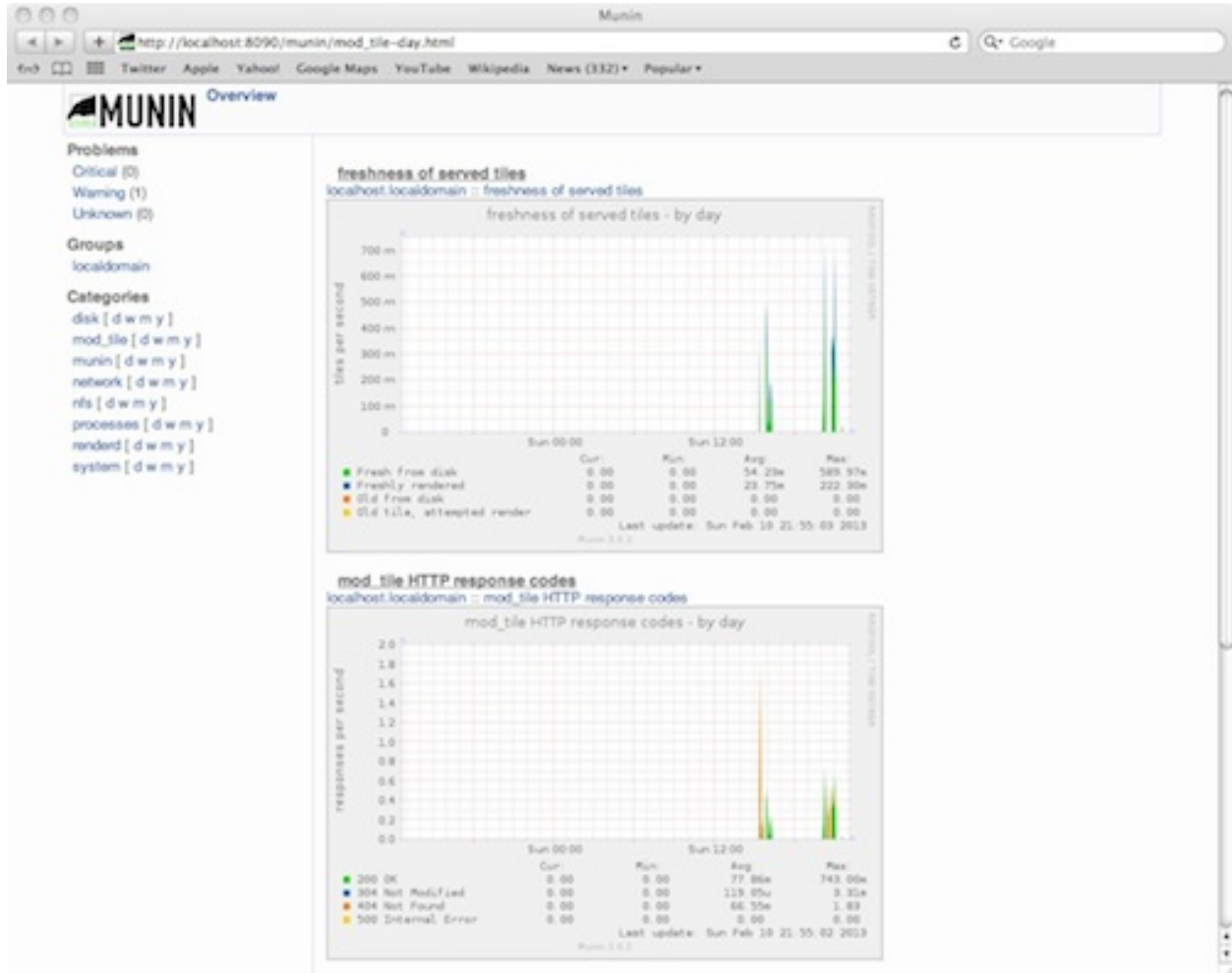


Fig. 2: *Figuur MT-2 - Munin in actie*

Verder loggen/volgen:

- PostgreSQL debug output zetten: `/etc/postgresql/9.1/main/postgresql.conf`, zet `client_min_messages = log`
- volgen renderd logfile: `tail -f /var/log/syslog |grep renderd &`
- volgen postgresql log: `tail -f /var/log/postgresql/postgresql-9.1-main.log &`
- losse tile: <http://localhost:8090/osm/17/67318/43072.png>

## 2.5 Demo

Een demo app staat onder `/var/www/osm/slippymap.html`. Hier HTML aanpassen om centrum op Amsterdam te zetten. Evt port zetten indien port forwarding naar local VM (8090 bijv). Dan zetten.

```
var newLayer = new OpenLayers.Layer.OSM("Local Tiles",
    "http://localhost:8090/osm/{z}/{x}/{y}.png", {numZoomLevels: 19});
```

Het resultaat met wat logging info hieronder.



Fig. 3: *Figuur MT-3 - Amsterdam-C Extent met renderd+PostgreSQL logging*

## 2.6 Tiles in EPSG:28992

Dit betreft Stap 2. Totnuoet is een standaard Mapnik/mod\_tile toolchain opgezet. We moeten een aantal zaken wijzigen om hetzelfde voor EPSG:28992 tiles te realiseren. Dit is al eerder beschreven in <http://justobjects.org/blog/2010/openstreetmap-tiles-for-dutch-projection-epsg28992>. We proberen data in EPSG:28992 te laden.

### 2.6.1 Data

We nemen eerst een klein stukje planet-data (488kb) rond de Nieuwmarkt in Amsterdam (file: nieuwmarkt.osm).

## Stappen

```
# DB aanmaken
createdb -E UTF8 -O osm gis28992 -T postgis2_template

# Data laden
osm2pgsql -W -U osm -d gis28992 -E EPSG:28992 --slim --cache-strategy sparse ↵
↪amsterdam.osm.pbf
# DIT WERKT NIET: DE DATA WORDT GELADEN IN EPSG:4326

# data laden als EPSG:4326 (WGS84)
osm2pgsql -c -W -U osm -d gis28992 -E EPSG:4326 --slim --cache-strategy sparse ↵
↪nieuwmarkt.osm
```

## 2.6.2 MapProxy

Deze stappen voor basis Mapproxy install en de demo app via mod\_wsgi in Apache. Vervolgens MapProxy koppelen aan de bestaande Mapnik config.

### Basis Installatie

Deze stappen voor MapProxy 1.5.0

```
# MapProxy Install 1.5.0
# Python Pip
sudo apt-get install python-pip

# Deps
sudo apt-get install python-imaging python-yaml libproj0
sudo apt-get install libgeos-dev python-lxml libgdal-dev python-shapely
sudo apt-get install build-essential python-dev libjpeg-dev zlib1g-dev libfreetype6-
↪dev
sudo pip install https://bitbucket.org/olt/pil-2009-raclette/get/default.tar.gz
sudo apt-get install python-yaml

# MapProxy
sudo pip install MapProxy

# Check install
mapproxy-util --version
```

### mod\_wsgi Koppelen

mod\_wsgi is een van de vele manieren om MapProxy aan te roepen. Hier direct in Apache via mod\_wsgi. We draaien hier gelijk de standaard demo app van MapProxy.

```
# mod_wsgi install
apt-get install libapache2-mod-wsgi

# create basis wsgi config for demo app
mkdir /opt/openbasiskaart/mapproxy/demo
cd /opt/openbasiskaart/mapproxy/demo
```

(continues on next page)

(continued from previous page)

```

# create basis mapproxy config
# maakt mapproxy.yaml en seed.yaml aan
mapproxy-util create -t base-config ./

# maak WSGI Python webapp (config.py) voor deze config (mapproxy.yaml)
mapproxy-util create -t wsgi-app -f mapproxy.yaml config.py

# maak webserver config waarin mapproxy webapp gemapped:

# deze file aanmaken in /etc/apache2/sites-available/mapproxy
<VirtualHost *:80>
    WSGIScriptAlias /mpdemo /opt/openbasiskaart/mapproxy/demo/config.py/

    <Directory /opt/openbasiskaart/mapproxy/demo>
        Order deny,allow
        Allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/mapproxy-error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel debug

    CustomLog ${APACHE_LOG_DIR}/mapproxy-access.log combined
</VirtualHost>

# aanmaken site voor apache
a2site mapproxy
apache2ctl restart

# cache directory moet schrijfbaar zijn!!
# voorlopig zo
mkdir /opt/openbasiskaart/mapproxy/demo/cache_dir
chmod 777 /opt/openbasiskaart/mapproxy/demo/cache_dir

# met browser naar http://localhost/mpdemo OK

```

## Mapnik als Bron

### Problemen

```

sudo pip install nik2img
nik2img.py osm.xml mapasd.png -f png256 -b 4.897 52.370 4.898 52.371
# geeft goede map

# test tile
http://localhost:8090/mpdemo/tms/1.0.0/mapnik_default_layer_EPSG900913/15/33659/
↪43999.png

# hmm /usr/share/proj/epsg file toch niet op orde, deze toevoegen
<900913> +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.
↪0 +units=m +nadgrids=@null +no_defs +over<>

```

Default mapproxy.yaml met eigen OSM.

```

services:
  demo:
  kml:
  tms:
    # needs no arguments
  wmts:
  wms:
    # srs: ['EPSG:4326', 'EPSG:900913']
    # image_formats: ['image/jpeg', 'image/png']
  md:
    # metadata used in capabilities documents
    title: MapProxy WMS Proxy
    abstract: This is the fantastic MapProxy.
    online_resource: http://mapproxy.org/
    contact:
      person: Your Name Here
      position: Technical Director
      organization:
      address: Fakestreet 123
      city: Somewhere
      postcode: 12345
      country: Germany
      phone: +49(0)000-000000-0
      fax: +49(0)000-000000-0
      email: info@omniscale.de
    access_constraints:
      This service is intended for private and evaluation use only.
      The data is licensed as Creative Commons Attribution-Share Alike 2.0
      (http://creativecommons.org/licenses/by-sa/2.0/)
    fees: 'None'

layers:
  - name: osm
    title: Omniscale OSM WMS - osm.omniscale.net
    sources: [osm_cache]
  - name: mapnik_default_layer
    title: Mapnik Default
    sources: [mapnik_default_cache]

caches:
  osm_cache:
    grids: [GLOBAL_MERCATOR, global_geodetic_sqrt2]
    sources: [osm_wms]

  mapnik_default_cache:
    grids: [GLOBAL_MERCATOR]
    sources: [default_mapnik]

sources:
  osm_wms:
    type: wms
    req:
      url: http://osm.omniscale.net/proxy/service?
      layers: osm

  default_mapnik:
    type: mapnik

```

(continues on next page)

(continued from previous page)

```
mapfile: /opt/openbasiskaart/mapnik/default/osm.xml
use_mapnik2: true
coverage:
#   bbox: [4.88,52.36,4.90,52.38]
      bbox: [543239.115,6865481.657,545465.505,6869128.129]
      srs: 'EPSG:900913'

grids:
  global_geodetic_sqrt2:
    base: GLOBAL_GEODETTIC
    res_factor: 'sqrt2'

globals:
  # # cache options
  cache:
    # where to store the cached images
    base_dir: './cache_data'
    # where to store lockfiles
    lock_dir: './cache_data/locks'

  # image/transformation options
  image:
    resampling_method: nearest
```

## seed.yaml

```
seeds:
  myseed1:
    caches: [osm_cache]
    grids: [GLOBAL_MERCATOR]
    coverages: [austria]
    levels:
      to: 10
    refresh_before:
      time: 2010-10-21T12:35:00

  mapnik_default_seed:
    caches: [mapnik_default_cache]
    grids: [GLOBAL_MERCATOR]
    coverages: [mapnik_default_coverage]
    levels:
      to: 15
    refresh_before:
      time: 2010-10-21T12:35:00

cleanups:
  clean1:
    caches: [osm_cache]
    grids: [GLOBAL_MERCATOR]
    remove_before:
      days: 7
      hours: 3
    levels: [2,3,5,7]

coverages:
  austria:
    bbox: [9.36, 46.33, 17.28, 49.09]
```

(continues on next page)

(continued from previous page)

```

bbox_srs: "EPSG:4326"
mapnik_default_coverage:
  bbox: [543239.115, 6865481.657, 545465.505, 6869128.129]
  bbox_srs: "EPSG:900913"

```

MapProxy met Mapnik2 lijkt moeizaam vanuit TMS, wel als we eerst seeden.

```
mapproxy-seed -f mapproxy.yaml -c 4 seed.yaml --seed=mapnik_default_seed
```

Uiteindelijk resultaat.

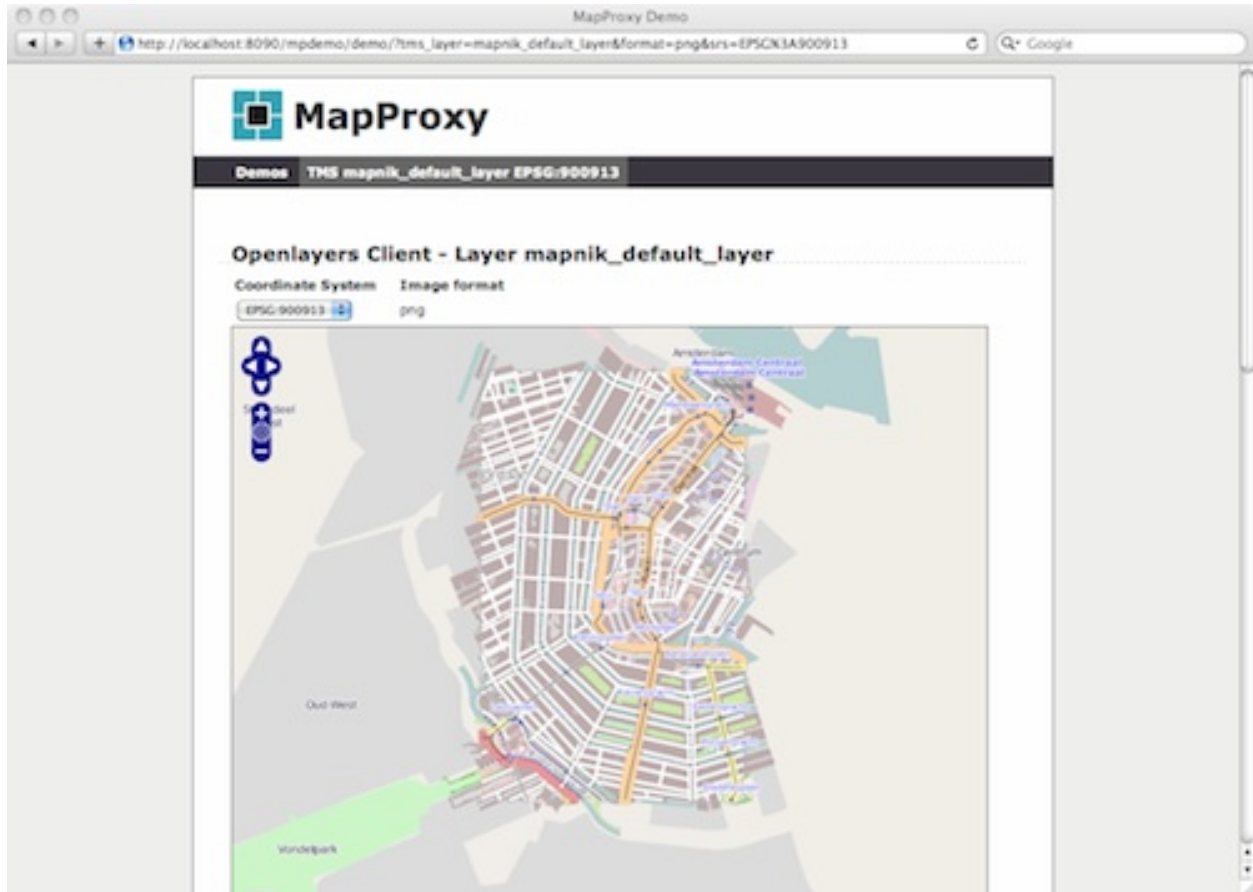


Fig. 4: *Figuur MT-4 - Eerste resultaat Mapnik met Mapproxy (900913+file cache)*

## MBTiles Cache

SQLite3 installatie. <http://www.sqlite.org>

```
sudo apt-get install sqlite3 libsqlite3-dev
```

Nieuwe Layer en Cache toevoegen in mapproxy.yaml

```

.
```

(continues on next page)

(continued from previous page)

```

layers:
  - name: osm
    title: Omniscale OSM WMS - osm.omniscale.net
    sources: [osm_cache]
  - name: mapnik_default_layer
    title: MapnikDefault
    sources: [mapnik_default_cache]
  - name: mapnik_mbtiles_default_layer
    title: MapnikMBTilesDefault
    sources: [mapnik_mbtiles_default_cache]
.
.
caches:
  osm_cache:
    grids: [GLOBAL_MERCATOR, global_geodetic_sqrt2]
    sources: [osm_wms]

  mapnik_default_cache:
    sources: [default_mapnik]
    grids: [GLOBAL_MERCATOR]

  mapnik_mbtiles_default_cache:
    sources: [default_mapnik]
    grids: [GLOBAL_MERCATOR]
    cache:
      type: mbtiles
      filename: mapnik_default.mbtiles

```

Seeden voor MBTiles cache.

```
sudo mapproxy-seed -f mapproxy.yaml -c 1 seed.yaml --seed=mapnik_mbtiles_default_seed
```

Notes:

- only one worker/thread -c 1.If larger than 1 gives error: "OperationalError: database is locked"
- only seeding works, not via TMS

Error wanneer expliciet tilen via TMS.

```

2013-02-15 16:33:07,061 - CRITICAL - mapproxy.wsgiapp - fatal error in tms for /tms/1.
↳0.0/mapnik_mbtiles_default_layer_EPSG900913/17/134637/175982.png
Traceback (most recent call last):
  File "/usr/local/lib/python2.7/dist-packages/mapproxy/wsgiapp.py", line 166, in __
↳call__
    resp = self.handlers[handler_name].handle(req)
  File "/usr/local/lib/python2.7/dist-packages/mapproxy/service/base.py", line 30, in _
↳handle
    return handler(parsed_req)
  File "/usr/local/lib/python2.7/dist-packages/mapproxy/service/tile.py", line 74, in _
↳map
    tile = layer.render(tile_request, use_profiles=tile_request.use_profiles, _
↳coverage=limit_to)
  File "/usr/local/lib/python2.7/dist-packages/mapproxy/service/tile.py", line 265, _
↳in render
    tile = self.tile_manager.load_tile_coord(tile_coord, with_metadata=True)
  File "/usr/local/lib/python2.7/dist-packages/mapproxy/cache/tile.py", line 105, in _
↳load_tile_coord

```

(continues on next page)



(continued from previous page)

```

        created_tiles = creator.create_tiles([tile])
    File "/usr/local/lib/python2.7/dist-packages/mapproxy/cache/tile.py", line 227, in _
↳create_tiles
        created_tiles = self._create_meta_tiles(meta_tiles)
    File "/usr/local/lib/python2.7/dist-packages/mapproxy/cache/tile.py", line 300, in _
↳create_meta_tiles
        created_tiles.extend(self._create_meta_tile(meta_tile))
    File "/usr/local/lib/python2.7/dist-packages/mapproxy/cache/tile.py", line 307, in _
↳create_meta_tile
        with self.tile_mgr.lock(main_tile):
    File "/usr/local/lib/python2.7/dist-packages/mapproxy/platform/cpython/lock.py", _
↳line 42, in __enter__
        self.lock()
    File "/usr/local/lib/python2.7/dist-packages/mapproxy/platform/cpython/lock.py", _
↳line 72, in lock
        raise LockTimeout('another process is still running with our lock')
LockTimeout: another process is still running with our lock

```

### Seeding en Cleanup (remove) voor default MBTiles cache. seed.yaml

```

seeds:
  .
  .
  mapnik_mbtiles_default_seed:
    caches: [mapnik_mbtiles_default_cache]
    grids: [GLOBAL_MERCATOR]
    coverages: [mapnik_default_coverage]
    levels:
      to: 17
  .
  .
cleanups:
  clean1:
    caches: [osm_cache]
    grids: [GLOBAL_MERCATOR]
    remove_before:
      days: 7
      hours: 3
      levels: [2,3,5,7]

  clean_mapnik_mbtiles_default:
    caches: [mapnik_mbtiles_default_cache]
    grids: [GLOBAL_MERCATOR]
    levels: [2,3,5,7]

```

### Commando voor cleanup.

```

sudo mapproxy-seed -f mapproxy.yaml -c 1 seed.yaml --cleanup=clean_mapnik_mbtiles_
↳default

```

### Util sqlite3 command line om database te beheren.

```

sudo sqlite3 cache_data/mapnik_default.mbtiles
SQLite version 3.7.13 2012-06-11 02:05:22
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select * from sqlite_master;

```

(continues on next page)

(continued from previous page)

```
table|tiles|tiles|2|CREATE TABLE tiles (  
                                zoom_level integer,  
                                tile_column integer,  
                                tile_row integer,  
                                tile_data blob)  
table|metadata|metadata|3|CREATE TABLE metadata (name text, value text)  
index|idx_tile|tiles|4|CREATE UNIQUE INDEX idx_tile on tiles  
                                (zoom_level, tile_column, tile_row)  
sqlite> select * from tiles;  
sqlite> select * from metadata;
```

---

## Mapcache Toolchain

---

Hieronder staat de “toolchain” beschreven om OSM RD Tiles te genereren en te serveren volgens de “MapCache Toolchain”. Dit is een toolchain op basis van Imposm, MapServer en MapCache.

Dit is de uiteindelijk gekozen toolchain voor OBKN, dus relevant.

### 3.1 Installaties

Hieronder de stappen voor installatie van de verschillende tools.

#### 3.1.1 Afhankelijkheden

Eerst afhankelijkheden installeren.

#### 3.1.2 UbuntuGIS PPA

<https://wiki.ubuntu.com/UbuntuGIS> !

```
# to add the UbuntuGIS PPA and update your packaging system.
sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
```

Installeer apache

```
sudo apt-get install apache2
```

Installeer git

```
sudo apt-get install git
```

### 3.1.3 Postgresql/PostGIS

On Ubuntu there are pre-packaged versions of both postgis and postgresql, so these can simply be installed via the ubuntu package manager.

```
sudo apt-get install postgres
sudo apt-get install pgadmin3

sudo -u postgres -i
# aanmaken user "osm" met zelfde password.
# answer yes for superuser (although this isn't strictly necessary)
createuser osm
psql -c "ALTER USER postgres WITH PASSWORD 'postgres';"
createdb -E UTF8 -O postgres postgis2_template
psql -d postgis2_template -c "CREATE EXTENSION postgis;"

# Aanmaken DB met PostGIS template
createdb -E UTF8 -O osm gis -T postgis2_template

# Zorg voor data schema, om postgis dingen te scheiden
psql gis
create schema data authorization osm;
alter database osm set search_path = data, public
grant all on spatial_ref_sys to public;

exit
```

### 3.1.4 Imposm

Imposm wordt gebruikt voor inlezen OSM Planet dump in Postgres. Zie ook <http://imposm.org/docs/imposm/latest/install.html> Install requirements

```
sudo apt-get install build-essential python-dev protobuf-compiler \
    libprotobuf-dev libtokyocabinet-dev python-psycopg2 \
    libgeos-cl
sudo apt-get install python-pip
sudo pip install imposm
```

### 3.1.5 Mapserver

De mapserver om de plaatjes te renderen

```
sudo apt-get install cgi-mapserver mapserver-bin
```

Test in browser of mapserver het doet

```
http://localhost/cgi-bin/mapserv
```

Als het goed is staat er nu iets als “No query information to decode. QUERY STRING is set, but empty.”

### 3.1.6 Mapcache

Installeer de tiling applicatie. Dit is een apache module

```
sudo apt-get install libmapcache mapcache-cgi mapcache-tools libapache2-mod-mapcache
mkdir ~/osm-demo/mapcache/cache
sudo chown www-data ~/osm-demo/mapcache/cache/
```

### 3.1.7 Sqlite3

Voor mbtiles nodig.

```
sudo apt-get install sqlite3
```

## 3.2 Data

### 3.2.1 Data downloaden

```
mkdir /opt/openbasiskaart/data

# PBF download (53 MB)
wget http://osm-metro-extracts.s3.amazonaws.com/amsterdam.osm.pbf
```

### 3.2.2 Data inladen

Lees de data (voorbewerking van imposm)

```
sudo imposm --proj=EPSG:28992 --read amsterdam.osm.pbf
```

Schrijf de data naar postgis

```
sudo imposm --write --database osm --proj=EPSG:28992 --host localhost --user osm --
↳port 5432
```

Check of de data goed is geschreven (in relatie tot de herprojectie)

```
select distinct(st_srid(geometry)) from osm_new aeroways;
```

Als het goed is komt hier alleen 28992 uit. Zo niet, dan moet je iets herstellen zodat dit wel het geval wordt!

## 3.3 Service

Maak de service in de mapfile

### 3.3.1 Mapserver utils

Zie ook <http://trac.osgeo.org/mapserver/wiki/RenderingOsmDataUbuntu#Installmapserver-utilsmapfilegenerator>  
Download mapserverutils

```
git clone https://github.com/mapserver/basemaps.git
cd basemaps
gedit osmbase.map
```

```
-----8<-----
    WEB
    ...
    IMAGEPATH "/tmp/ms_tmp/"
    IMAGEURL  "/ms_tmp/"
    END
    ...
----->8-----
```

```
vi Makefile
-----8<-----
OSM_SRID=28992
OSM_UNITS=meters
OSM_EXTENT=12000 304000 280000 620000
...
STYLE=default
...
OSM_WMS_SRS=EPSG:28992
----->8-----
```

```
mkdir /tmp/ms_tmp
chmod 777 /tmp/ms_tmp
```

Execute the mapserver-utils makefile to generate the mapfile. Note that the first ↵  
 ↵time you run 'make' several large files will be downloaded (country boundaries, etc.  
 ↵). This will happen only the first time.

```
cd mapserver-utils-svn
cd data
gedit Makefile
```

```
#Verander bij boundary lines de link naar >http://www.naturalearthdata.com/http//
↵www.naturalearthdata.com/download/10m/cultural/ne_10m_admin_0_boundary_lines_land.
↵zip<
# Verander bij unzip van boundary lines de parameter die aan unzip wordt ↵
↵meegegeven naar >ne_10m_admin_0_boundary_lines_land.zip<
make
cd ..
make
```

Comment de regel uit waarop staat

```
#CONFIG "PROJ_LIB" "/home/<USERNAME>/<path_to_mapserverutil>"
```

Verzeker je ervan dan bij de srs-en 28992 staat

```
"wms_srs" "EPSG:28992 EPSG:4326 EPSG:3857 EPSG:2154 EPSG:310642901 EPSG:4171 ↵
↵EPSG:310024802 EPSG:310915814 EPSG:310486805 EPSG:310702807 EPSG:310700806 ↵
↵EPSG:310547809 EPSG:310706808 EPSG:310642810 EPSG:310642801 EPSG:310642812 ↵
↵EPSG:310032811 EPSG:310642813 EPSG:2986 "
```

Maak verbinding naar de osm database en voer het volgende script uit

```

set session authorization osm;
-- DROP VIEW osm_new_waterways_gen0_view;

CREATE OR REPLACE VIEW osm_new_waterways_gen0_view AS
  SELECT osm_new_waterways.id, osm_new_waterways.osm_id, osm_new_waterways.name, osm_
↪new_waterways.type, st_simplifypreservetopology(osm_new_waterways.geometry, 2
↪00::double precision) AS geometry
  FROM osm_new_waterways;

ALTER TABLE osm_new_waterways_gen0_view
  OWNER TO osm;

-- View: osm_new_waterways_gen1_view

-- DROP VIEW osm_new_waterways_gen1_view;

CREATE OR REPLACE VIEW osm_new_waterways_gen1_view AS
  SELECT osm_new_waterways.id, osm_new_waterways.osm_id, osm_new_waterways.name, osm_
↪new_waterways.type, st_simplifypreservetopology(osm_new_waterways.geometry, 5
↪0::double precision) AS geometry
  FROM osm_new_waterways;

ALTER TABLE osm_new_waterways_gen1_view
  OWNER TO osm;

CREATE TABLE osm_new_waterways_gen1 AS
  SELECT * FROM osm_new_waterways_gen1_view;

CREATE TABLE osm_new_waterways_gen0 AS
  SELECT * FROM osm_new_waterways_gen0_view;

```

Test de mapfile door naar

```

http://yourserver.tld/cgi-bin/mapserv?map=/path/to/osm-demo/mapserv-utis-svn/osm-
↪outlined,google.map&mode=browse&template=openlayers&layers=all

```

Te gaan. Als er een pagina met openlayers en de kaart verschijnt, is het goed gegaan.

Kopieër de mapfile, fonts, font.lst en de datamap naar de gewenste plek: /opt/openbasiskaart/maps

## 3.4 Tiling

Maak het cache pad een geef www-data schrijfrechten

```

mkdir /opt/openbasiskaart/cache
sudo chown www-data /opt/openbasiskaart/cache/
cd /opt/openbasiskaart/cache

```

Maak mbtiles cache

```

sudo sqlite3 osmcache.mbtiles

```

Voer uit

```

create table if not exists images(
  tile_id text,
  tile_data blob,
  primary key(tile_id));
create table if not exists map (
  zoom_level integer,
  tile_column integer,
  tile_row integer,
  tile_id text,
  foreign key(tile_id) references images(tile_id),
  primary key(tile_row,tile_column,zoom_level));
create table if not exists metadata(
  name text,
  value text); -- not used or populated yet
create view if not exists tiles
  as select
    map.zoom_level as zoom_level,
    map.tile_column as tile_column,
    map.tile_row as tile_row,
    images.tile_data as tile_data
  from map
  join images on images.tile_id = map.tile_id;
.exit

```

Geef www-date rechten op de cache

```
sudo chown www-data osmcache.mbtiles
```

### 3.4.1 Setup mapcache

De configuratie voor mapcache

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- see the accompanying mapcache.xml.sample for a fully commented configuration_
↪file -->

<mapcache>
  <cache name="mbtiles" type="mbtiles">
    <dbfile>/opt/openbasiskaart/cache/osmcache.mbtiles</dbfile>
  </cache>

  <source name="osm" type="wms">
    <getmap>
      <params>
        <FORMAT>image/png</FORMAT>
        <LAYERS>default</LAYERS>
        <SRS>epsg:28992</SRS>
      </params>
    </getmap>

    <http>
      <url>http://localhost/cgi-bin/mapserv?map=/opt/openbasiskaart/maps/
↪osm-default.map</url>
    </http>
  </source>

```

(continues on next page)



(continued from previous page)

```

<grid name="rd">
  <metadata>
    <title>Rijksdriehoek-stelsel</title>
  </metadata>
  <!--extent>12000,304000,280000,620000</extent-->
  <extent>98827,469579,142183,502096</extent> <!-- Extent adam -->
  <srs>epsg:28992</srs>
  <resolutions>3440.64 1720.32 860.16 430.08 215.04 107.52 53.76 26.88
↪13.44 6.72 3.36 1.68 0.84 0.42 0.21</resolutions>
  <units>m</units>
  <size>256 256</size>
</grid>
<tileset name="osm">
  <metadata>
    <title>OSM MapServer served map</title>
    <abstract>see http://trac.osgeo.org/mapserver/wiki/
↪RenderingOsmDataUbuntu</abstract>
  </metadata>
  <source>osm</source>
  <cache>mbtiles</cache>
  <format>PNG</format>
  <grid>rd</grid>
</tileset>

<default_format>JPEG</default_format>

<service type="wms" enabled="true">
  <full_wms>assemble</full_wms>
  <resample_mode>bilinear</resample_mode>
  <format>JPEG</format>
  <maxsize>4096</maxsize>
</service>
<service type="wmts" enabled="true"/>
<service type="tms" enabled="true"/>
<service type="kml" enabled="true"/>
<service type="gmaps" enabled="true"/>
<service type="ve" enabled="true"/>
<service type="demo" enabled="true"/>
<errors>log</errors>
<lock_dir>/tmp</lock_dir>
</mapcache>

```

Sla dit bestand op en zet het in `/opt/openbasiskaart/cache`.

## 3.5 Seeding

seed de tiles

```
mapcache_seed -c mapcache-osm.xml -t osm -g rd -z 0,15 -n <numofthreads>
```

## 3.6 Andere stijlen

Andere stijlen die gemaakt zijn, zijn onder andere de google en de bing styles. Dit is makkelijk te genereren door naar je mapservr utils folder te gaan, de makefile aan te passen. Bij STYLE kan hier ipv default “google” of “bing” worden ingevuld. Run het command make en de mapfile wordt gegenereert (vergeet niet de PROJ variabele uit te commenten). Hierna de mbtiles caches te maken en daarna de mapcache-config.xml aanpassen aan de nieuwe mapfile en de mbtiles caches voor google/bing/etc.. Restart apache en seed de nieuwe tilesets.

## 3.7 Nog te doen

Invalidaten van tiles Osmosis??

---

## Ontwikkelomgeving

---

Voor het ontwikkelen en testen van toolchains is het handig om dit op je eigen systeem te doen. Echter, je wilt niet je systeem “vervuilen” met allerlei installaties. Ook is het dan lastig om een werkende toolchain over te dragen aan de werkelijke test/productie server, bijv als je op Mac OSX of Windows werkt of een oudere Ubuntu.

Daarom wordt steeds meer met **virtualisatie** gewerkt, ook op lokale systemen als bijv Mac OSX. Ik beschrijf hieronder mijn werkwijze om op Mac OSX met Ubuntu 12.10 te ontwikkelen en testen.

Er zijn hierbij twee onmisbare tools:

1. VirtualBox, zie <http://www.virtualbox.org>
2. Vagrant, zie <http://www.vagrantup.com>

Dit blijkt een gouden combinatie te zijn, ook als je niet met Puppet of Chef werkt. Dan is Vagrant helemaal handig, maar gebruik van Puppet of Chef hoeft dus niet in eerste instantie.

Hoewel VirtualBox natuurlijk op zichzelf is te gebruiken, voegt Vagrant juist extra handigheden toe, met name om lokaal gemakkelijk met je VMs, “boxes” geheten in Vagrant, te werken, bijv.

- snel VMs downloaden, in de lucht krijgen
- gemakkelijk port forwarden, m.n. voor SSH en je browser
- gemakkelijk lokaal inloggen
- evt “boxes” delen met elkaar
- files delen via shares, /vagrant is altijd aanwezig

### 4.1 Installaties

Zie <http://docs.vagrantup.com/v1/docs/getting-started/index.html>

### 4.1.1 VirtualBox

Laatste versie (in feb 2013: v4.2.6) downloaden en installeren. Zie <https://www.virtualbox.org/wiki/Downloads>. Ook de extension packs installeren.

### 4.1.2 Vagrant

Voor Mac OSX <http://downloads.vagrantup.com/tags/v1.0.6>. Maar kan op Linux OS-es simpeler, of via RubyGems, `gem install vagrant` (Vagrant is in Ruby ontwikkeld). Feb 2013: 1.0.6 installed.

## 4.2 Werken met Vagrant Boxes

### 4.2.1 Box Aanmaken

Belangrijkst is om een begin te maken met goede “box”, een VM image die door Vagrant geconfigureerd wordt.

Zie boxes o.a. hier: <http://www.vagrantbox.es>

We kiezen een Ubuntu 12.10 Server image: <http://cloud-images.ubuntu.com/quantal/current/quantal-server-cloudimg-vagrant-amd64-disk1.box>

Om bijv. een headless Ubuntu 12.10 server up en running in enkele minuten. Doe in lege directory.

```
vagrant box add quantal64-server http://cloud-images.ubuntu.com/quantal/current/
↪quantal-server-cloudimg-vagrant-amd64-disk1.box
vagrant init
vagrant up
```

Dit voegt een nieuwe VM toe aan VirtualBox. Een file genaamd VagrantFile wordt aangemaakt in de directory waar je de box aanmaakt als boven. Deze file bevat config info over je Box, bijv hier de “quantal64-server”.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant::Config.run do |config|
  # All Vagrant configuration is done here. The most common configuration
  # options are documented and commented below. For a complete reference,
  # please see the online documentation at vagrantup.com.

  # Every Vagrant virtual environment requires a box to build off of.
  config.vm.box = "quantal64-server"

  # The url from where the 'config.vm.box' box will be fetched if it
  # doesn't already exist on the user's system.
  # config.vm.box_url = "http://domain.com/path/to/above.box"

  # Boot with a GUI so you can see the screen. (Default is headless)
  config.vm.boot_mode = :gui

  # Assign this VM to a host-only network IP, allowing you to access it
  # via the IP. Host-only networks can talk to the host machine as well as
  # any other machines on the same network, but cannot be accessed (through this
  # network interface) by any external networks.
  # config.vm.network :hostonly, "192.168.33.10"
```

(continues on next page)

(continued from previous page)

```

# Assign this VM to a bridged network, allowing you to connect directly to a
# network using the host's network device. This makes the VM appear as another
# physical device on your network.
config.vm.network :bridged

# Forward a port from the guest to the host, which allows for outside
# computers to access the VM, whereas host only networking does not.
config.vm.forward_port 80, 8090

# Share an additional folder to the guest VM. The first argument is
# an identifier, the second is the path on the guest to mount the
# folder, and the third is the path on the host to the actual folder.
# config.vm.share_folder "v-data", "/vagrant_data", "../data"

# Enable provisioning with Puppet stand alone. Puppet manifests
# are contained in a directory path relative to this Vagrantfile.
.
.
end

```

### Inloggen op je box

```
vagrant ssh
```

De box down brengen doe je met.

```
vagrant halt
```

Mocht je nu je box “verklodid” hebben dan kun je weer schoon starten via.

```
vagrant destroy
```

### Notes:

- als je bent ingelogd via `vagrant ssh` kun je via `sudo su - root` worden
- in settings binnen VB: zet op 64-bits en NW Adapter1 via en0
- set networking in VB to NAT to enable 2222->22 SSH forwarding

## 4.2.2 Werken met je Box

Je kunt de VM natuurlijk starten vanuit VirtualBox, maar heel handig is om je VM Box te zien als een remote headless server en deze vanuit SSH te benaderen. De directory waar je in staat is als `/vagrant` gemount binnen je box als je inlogt.

Inloggen op je box (vanuit dir waar VagrantFile staat)

```
vagrant up
vagrant ssh
```

Als je op je box bent ingelogd kun je verder werken als normaal.

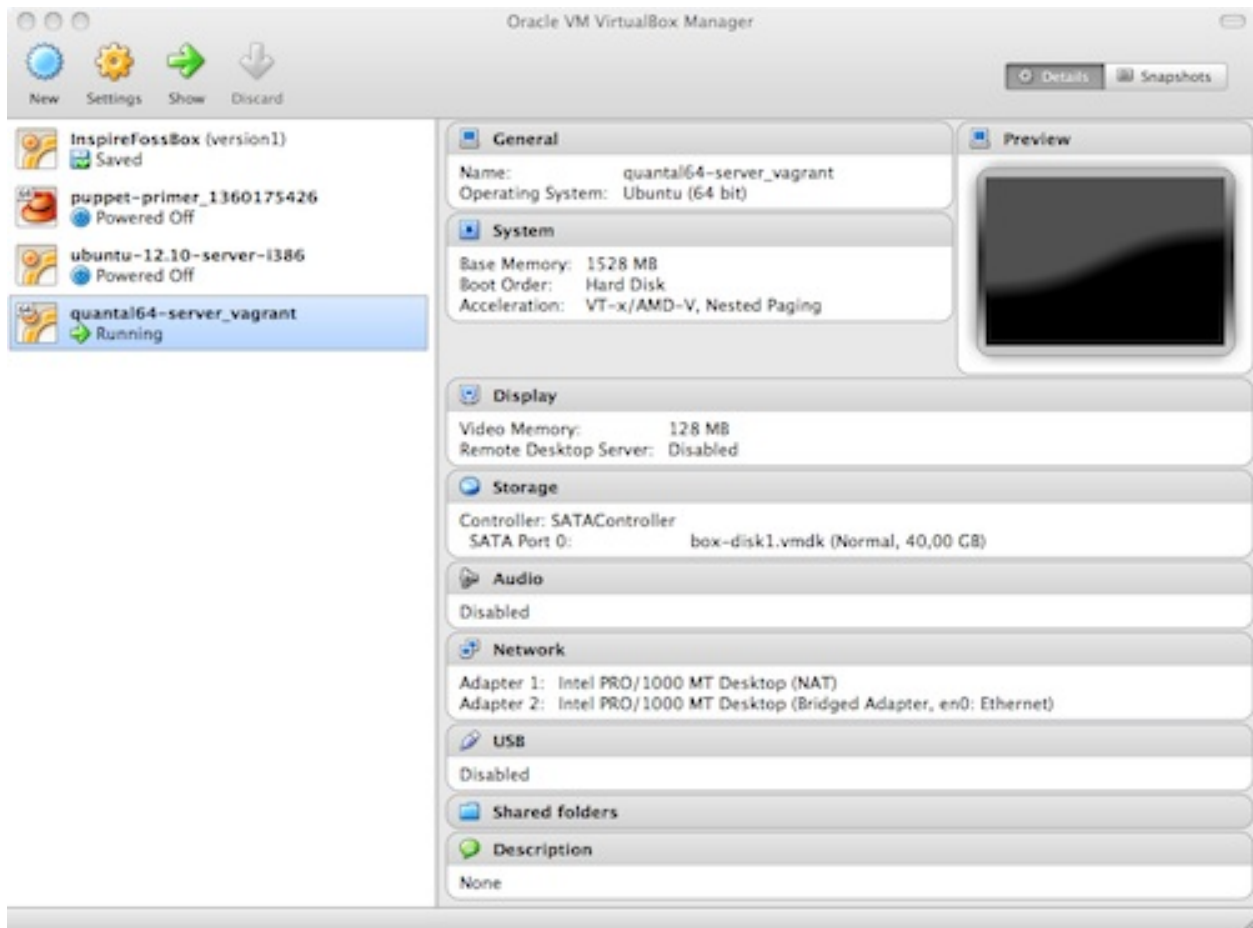


Fig. 1: *Figuur ONT-1 - VirtualBox Screenshot*

## CHAPTER 5

---

### Links

---

Hieronder links naar relevante zaken rond het project.





## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `search`



---

## Bibliography

---

- [GDALOGR] GDAL/OGR, <http://gdal.org>
- [PostGIS] PostGIS/PostgreSQL, <http://postgis.refrations.net>
- [Switch2OSMTiles] zelf tileserver bouwen met OSM tools, <http://switch2osm.org/serving-tiles/manually-building-a-tile-server-12-04/>
- [TileExpiry] Tile Expiry script, [https://github.com/openstreetmap/mod\\_tile/blob/master/openstreetmap-tiles-update-expire.sh](https://github.com/openstreetmap/mod_tile/blob/master/openstreetmap-tiles-update-expire.sh)
- [OSMDBUpdate] Martijn v Exel: hoe OSM DB up to date houden, <http://oegeo.wordpress.com/2012/03/06/a-self-updating-openstreetmap-database-of-us-bridges-a-step-by-step-guide/>
- [OSMPlanetCities] OSM Planet files voor steden, - OSM planet file single city, bijv Amsterdam, <http://metro.teczno.com/#amsterdam>
- [MBTiles] MBTiles spec, <http://mapbox.com/developers/mbtiles>
- [OSM2PSGSQL] <http://wiki.openstreetmap.org/wiki/Osm2pgsql>
- [Mapnik] <http://mapnik.org>
- [Imposm] <http://imposm.org>
- [Osmosis] <http://wiki.openstreetmap.org/wiki/Osmosis>
- [MapCache] <http://mapserver.org/trunk/mapcache>
- [GitHub] Basiskaart GitHub project, <https://github.com/opengeogroep/basiskaart>
- [Optimalisatie] Toolchain optimalisaties, Fred Ramm, <http://www.remote.org/frederik/tmp/ramm-osm2pgsql-sotm-2012.pdf>