User:Ika-chan!/Fantasy maps with OSM software

< User:Ika-chan!

This article **is in the process of an expansion or major restructuring, and is not yet ready for use.**

I am currently developing instructions for creating a standalone OpenStreetMap (OSM) server because my roleplay world is not compatible with the world of OpenGeoFiction. I am also hopeful that this page will encourage more people to set up OSM-based projects instead of playing around with the live map (OpenPokéMap, anyone?).

I know it is complicated, but someone has to do it. The endgame is that the installation of an standalone OpenStreetMap clone will be as easy as installing MediaWiki, which is why the latter is so popular.

Contents

Preparation

If you prefer to copy and paste commands, you can get to this page by typing this short link:

minoa.li/geso (https://minoa.li/geso)

Part 1: Prepare the server

Download and install Ubuntu

This section has different instructions for physical servers and virtual servers. Click “Show” next to the type of server that you have.

Physical server

1. Download the official disk image (ISO) from the Ubuntu website (https://www.ubuntu.com/download/desktop).
2. Create a bootable USB stick by following the official tutorials: macOS (https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-macos) • Ubuntu (https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-ubuntu) • Windows (https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows)
3. Start up the server and boot into the newly-created USB stick.
4. When you reach the “Updates and other software” screen, select “Minimal installation”.

5. When you start Ubuntu for the first time, follow the introduction screens, because it will ask you if you wish to “help improve Ubuntu”.
6. When you reach the “Help improve Ubuntu” screen, you should select “No, don’t send system info”.
7. **Laptops only**: By default, Ubuntu goes into ‘hibernation’ when you close the lid of a laptop-based server: this prevents users from accessing the server when the lid is closed. Run the following command in **Terminal** to correct this:

```
sudo sed -i -e 's/#HandleLidSwitch=suspend
/HandleLidSwitch=ignore/g' /etc/systemd/logind.conf
```

Virtual server

1. Download the official disk image (ISO) from the Ubuntu website (https://www.ubuntu.com/download/desktop).
2. Start up the server and boot into ISO disk image.
3. When you reach the “Updates and other software” screen, select “Minimal installation”.
4. When you start Ubuntu for the first time, follow the introduction screens, because it will ask you if you wish to “help improve Ubuntu”.
5. When you reach the “Help improve Ubuntu” screen, you should select “No, don’t send system info”.
6. Install **Open VM Tools Desktop**, which will make it easier to copy and paste text between the host and guest. Run the following command in **Terminal**, and restart the computer:

```
sudo apt update && \
sudo apt install -y open-vm-tools-desktop && \
reboot
```

Change the download server

The download server should be reviewed, to reduce the time that Ubuntu takes to download packages and updates:

1. Open **Software & Updates**.
2. Change to the “Ubuntu Software” tab, and change the “Download from” drop-down menu to “Main server” or “Other” (the latter allows you to have Ubuntu detect the fastest mirror).

Improve usability and security

You should also make the following tweaks for better user experience and improved security. Open **Settings**, then:

1. **Disable Notifications**: Click **Notifications** on the sidebar, and set all options to “Off”.
2. **Disable searching for software**: Click **Search** on the sidebar, and set “Ubuntu Software” to ”Off”.
3. **Limit Usage & History**: Click **Privacy** on the sidebar, click **Usage & History**, and set a time limit as desired.

4. **Disable Problem Reporting**: Remain on the **Privacy** section, and set “Problem Reporting” to “Never”.
5. **Disable Autorun**: Click **Devices** on the sidebar, click **Removable Media** on the next sidebar, and check “Never prompt or start programs on media insertion”.

You may also need to set a fixed IP address. Click “Show” next to the type of internet connection that you have, to show the most appropriate instructions.

Ethernet (wired) connection

1. Open **Settings**, and click **Network** on the sidebar.
2. Click the cogwheel to the right of the active connection.
3. Uncheck “Make available to other users”, for security reasons.
4. Click the **IPv4** tab, and change the “IPv4 Method” to “Manual”.
5. Type in the connection details for your router, and click “Apply”.

Wi-Fi connection

1. Open **Settings**, and click **Wi-Fi** on the sidebar.
2. Select the name of your Wi-Fi network, and type in the password.
3. Click the cogwheel to the right of the active connection.
4. Uncheck “Make available to other users”, for security reasons.
5. Click the **IPv4** tab, and change the “IPv4 Method” to “Manual”.
6. Type in the connection details for your router, and click “Apply”.

Add Secure Shell (SSH) support

Ubuntu does not come with Secure Shell (SSH) support by default. If you plan to access your server from another computer, run the following command in **Terminal** to install OpenSSH:

```
sudo apt install -y openssh-client openssh-server
```

Enable Samba (optional)

Replace `[WORKGROUP]` with your workgroup name, and `[COMPUTER]` with your computer name:

```
sudo apt install -y samba && \
sudo sed -i -e 's/workgroup = WORKGROUP/workgroup =
[WORKGROUP]/g' /etc/samba/smb.conf && \
sudo sed -i '$a\\n[COMPUTER]\n   path = \/\n   read only = no\n
browsable = yes\n   guest ok = no' /etc/samba/smb.conf && \
sudo smbpasswd -a $USER
```

Terminal will automatically replace `$USER` with your user name.

Set Rails Port mode and create base directories

Run the following commands in **Terminal** to permanently set the Rails Port installation mode to "production",[1] before creating the directories that will host server and planet files, including the tiles (Terminal will automatically replace $USER with your user name):

```
sudo sed -i "\$aRAILS_ENV=\"production\"" /etc/environment && \
sudo mkdir -p /srv/planet /srv/styles /srv/tools /srv/www/api
/srv/www/tiles /srv/www/rails /srv/www/search && \
sudo chown -R $USER /srv
```

The above command will create this starter folder structure:

```
srv
├─planet   # Planet dumps (empty by default)
├─styles   # Stylesheets
├─tools    # Utilities, such as Osmosis
└─www
  ├─api    # CGImap cache (empty by default)
  ├─tiles  # Tile cache (empty by default)
  ├─rails  # Rails Port
  └─search # Nominatim
```

Add OSM repository, remove telemetry and update Ubuntu

Finally, run the following commands in **Terminal** to add the OpenStreetMap (https://launchpad.net/~osmadmins/+archive/ubuntu/ppa) repository, remove the Popularity Contest package, and install all pending Ubuntu updates:

```
sudo add-apt-repository -y ppa:osmadmins/ppa && \
sudo apt update && \
sudo apt remove -y popularity-contest && \
sudo apt upgrade -y && \
sudo apt autoremove -y && \
reboot
```

Part 2: Build the Website

This part covers the website side of OpenStreetMap, including CGImap, Nominatim, Osmosis, Rails Port, and Phusion Passenger.

Install the website's dependencies

Run the following commands in **Terminal** to install the following dependencies for the Rails Port:[2][3][4][5]

```
sudo apt install -y apache2 apache2-dev build-essential bundler
firefox-geckodriver git-core imagemagick libarchive-dev libbz2-dev
libffi-dev libgd-dev libmagickwand-dev libpq-dev libruby2.5
libsasl2-dev libxml2-dev libxslt1-dev nodejs postgresql
postgresql-contrib ruby2.5 ruby2.5-dev npm apt-transport-https ca-
certificates dirmngr gnupg libboost-dev libboost-filesystem-dev
libboost-locale-dev libboost-program-options-dev libboost-system-
dev libcrypto++-dev libfcgi-dev libmemcached-dev libpqxx-dev
libyajl-dev cmake g++ libapache2-mod-php libexpat1-dev libproj-dev
php php-intl php-pgsql postgresql-10-postgis-2.4 postgresql-10-
postgis-scripts postgresql-contrib-10 postgresql-server-dev-10
python3-setuptools python3-dev python3-pip python3-psycopg2
python3-tidylib zlib1g-dev osmctools osmosis && \
sudo gem2.5 install bundler && \
sudo npm install -g yarn
```

Next, install **Phusion Passenger**:[3]

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-
keys 561F9B9CAC40B2F7 && \
sudo sh -c 'echo deb https://oss-binaries.phusionpassenger.com
/apt/passenger bionic main > /etc/apt/sources.list.d
/passenger.list' && \
sudo apt update && \
sudo apt install -y libapache2-mod-passenger && \
sudo a2enmod passenger && \
sudo apache2ctl restart
```

Test Phusion Passenger (recommended)

You can run the following command in **Terminal** (one at a time), and follow the on-screen instructions, to check if Phusion Passenger is running:

```
sudo /usr/bin/passenger-config validate-install
sudo /usr/sbin/passenger-memory-stats
```

Finally, install the latest version of **Osmosis**:[6]

```
cd /srv/tools && \
wget -O osmosis.tgz https://github.com/openstreetmap/osmosis
/releases/download/0.48.2/osmosis-0.48.2.tgz && \
mkdir osmosis && tar -xpzf osmosis.tgz -C osmosis && \
echo "alias osmosis='/srv/tools/osmosis/bin/osmosis'" >> ~/.bashrc
&& \
source ~/.bashrc
```

Clone and configure the Rails Port

You can run the following command in **Terminal** to download the OpenStreetMap website from GitHub (you do not need the entire history),[2] create a writeable temporary directory,[1] and then remove the event banners (which are relevant only to the live map):

```
git clone -b live --depth=1 https://github.com/openstreetmap
/openstreetmap-website.git /srv/www/rails && \
cd /srv/www/rails && \
mkdir tmp && \
chmod -R 777 tmp && \
rm app/assets/images/banners/*.png && \
echo -e 'blank:\n id: blank' > config/banners.yml
```

Disable warnings about optional missing packages (optional)

Run the following command to prevent the server from complaining about the optional missing packages (advpng, gifsicle, jhead, jpegoptim, jpegtran, optipng, pngcrush, pngquant, and svgo):

```
sed -i -e 's/servers)/servers)\n    config.assets.image_optim =
false/g' config/application.rb
```

Next, copy two configuration files, and then open `settings.yml` in **nano**:

```
cp config/example.database.yml config/database.yml && \
cp config/example.storage.yml  config/storage.yml  && \
nano config/settings.yml
```

Replace the following fields (`server_url`, `nominatim_url` and `overpass_url`) with the domains or IP addresses that you want to use. For example:

```
server_url: "maps.minoa"
nominatim_url: "http://maps.minoa/nominatim/"
overpass_url: "http://maps.minoa/api/interpreter"
```

Next, use Bundler to install the Ruby gems (entering the password when requested):

```
rm Gemfile.lock && \
bundle install && \
bundle exec rake yarn:install && \
touch config/settings.local.yml
```

Setup the website's database

Osmosis requires the user to have a PostgreSQL password to create the planet file.

Type the following commands to set up your PostgreSQL account and build the databases for the Rails Port: Terminal will replace `$USER` with your user name, but replace `[PASSWORD]` with one of your own, and remember it:[2]

```
sudo -u postgres createuser -s $USER && \
bundle exec rake db:create && \
psql -d osm -c "ALTER ROLE $USER WITH LOGIN PASSWORD '[PASSWORD]'
VALID UNTIL 'infinity'" && \
psql -d osm -c "CREATE EXTENSION btree_gist" && \
psql -d osm -f db/functions/functions.sql && \
bundle exec rake db:migrate
```

The `bundle exec rake test:db` and `bundle exec rails server` commands, that appear in the instructions for the Rails Port, do not appear to apply to production-stage installations, and can be ignored.

Setup CGImap

Next, Install and build CGImap from source:[4]

```
git clone -b v0.8.3 --depth=1 https://github.com/zerebubuth
/openstreetmap-cgimap.git /srv/tools/CGImap && \
cd /srv/tools/CGImap && \
./autogen.sh && \
./configure --enable-yajl && \
make
```

Create a new CGImap configuration file, called `cgimap-wrapper`, in **nano**:

```
nano /srv/tools/CGImap/scripts/cgimap-wrapper
```

Paste the following syntax into the editing window, replacing `[USER]` and `[PASSWORD]` with your PostgreSQL credentials, then save the file:

Contents of `cgimap-wrapper`

```
#!/bin/bash

CGIMAP_HOST=localhost; export CGIMAP_HOST
CGIMAP_DBNAME=osm; export CGIMAP_DBNAME
```

```
CGIMAP_USERNAME=[USER]; export CGIMAP_USERNAME
CGIMAP_PASSWORD=[PASSWORD]; export CGIMAP_PASSWORD

CGIMAP_PIDFILE=cgimap.pid; export CGIMAP_PIDFILE
CGIMAP_LOGFILE=cgimap.log; export CGIMAP_LOGFILE

CGIMAP_MEMCACHE=localhost; export CGIMAP_MEMCACHE
CGIMAP_RATELIMIT=102400; export CGIMAP_RATELIMIT
CGIMAP_MAXDEBT=250; export CGIMAP_MAXDEBT

exec /srv/tools/CGImap/map
```

Finally, make `cgimap-wrapper` executable:

```
chmod +x /srv/tools/CGImap/scripts/cgimap-wrapper
```

Install Nominatim

In **Terminal**, create users for the Nominatim database:[5]

```
sudo useradd -d /srv/www/search -s /bin/bash -m nominatim && \
chmod a+x /srv/www/search && \
sudo -u postgres createuser -s nominatim && \
sudo -u postgres createuser www-data
```

Next, Install and build Nominatim and its user interface from source:[7]

```
cd /srv/www/search && \
wget -O Nominatim.tar.bz2 https://nominatim.org/release/Nominatim-
3.5.1.tar.bz2 && \
tar xf Nominatim.tar.bz2 && mv Nominatim-3.5.1 source && \
mkdir build && cd build && \
cmake /srv/www/search/source && make && \
git clone -b 1.2.1 --depth=1 https://github.com/osm-search
/nominatim-ui /srv/www/search-ui
```

Open `local.php` in **nano**, and add the following to the configuration file:

```
nano /srv/www/search/build/settings/local.php
```

Contents of `/etc/local.php`

```
<?php
    @define('CONST_Website_BaseURL', '/nominatim/');
    @define('CONST_Osm2pgsql_Flatnode_File', '/srv/www/search
/flatnode.file');
```

Create `config.js` in **nano**, and add the following to the configuration file (replace `maps.minoa` with your domain or IP address):

```
nano /srv/www/search-ui/dist/config.js
```

Contents of /etc/config.js

```
var Nominatim_Config = [];

Nominatim_Config['Nominatim_API_Endpoint'] = 'http://maps.minoa
/nominatim/';
Nominatim_Config['Map_Tile_URL'] = 'http://maps.minoa/standard
/{z}/{x}/{y}.png';
```

Part 3: Install the Tile Server

This part covers the tile server side of OpenStreetMap, including Mapnik and mod_tile.

Install the Tile Server components and dependencies

In **Terminal**, install the following components and dependencies, including **Mapnik**, **mod_tile** and **osm2pgsql**, and then test Mapnik: [8]

```
sudo apt install -y autoconf libagg-dev libboost-all-dev libcairo-
dev libcairomm-1.0-dev libfreetype6-dev libgdal-dev libgeos-dev
libgeos++-dev libgeotiff-epsg libicu-dev liblua5.1-dev liblua5.2-
dev libprotobuf-c0-dev libtiff5-dev libtool lua5.1 munin munin-
node protobuf-c-compiler ttf-unifont postgis gdal-bin libmapnik-
dev mapnik-utils python-mapnik libapache2-mod-tile osm2pgsql && \
mapnik-config -v && \
mapnik-config --input-plugins && \
python -c "import mapnik;print mapnik.__file__"
```

The tests should return no errors.

Create the Tile Server database

Create the tile server database named *gis*: Terminal will replace `$USER` with your user name.

```
sudo -u postgres createdb -E UTF8 -O $USER gis && \
psql -d gis -c "CREATE EXTENSION postgis;" && \
psql -d gis -c "CREATE EXTENSION hstore;" && \
psql -d gis -c "ALTER TABLE geometry_columns OWNER TO $USER;" && \
psql -d gis -c "ALTER TABLE spatial_ref_sys OWNER TO $USER;"
```

Part 4: Install a stylesheet

The following instructions are for the standard style that appears on the OpenStreetMap website.

Other stylesheets may have different dependencies and instructions, but it will be easier for you if they all reside in their own folders at `/srv/styles`.

Install the stylesheet dependencies

First, install the dependencies for the stylesheet in **Terminal**:[8]

```
sudo apt install -y nodejs-dev node-gyp libssl1.0-dev && \
sudo apt install -y npm nodejs && \
sudo apt install -y fonts-hanazono fonts-noto-cjk fonts-noto-
hinted fonts-noto-unhinted ttf-unifont && \
sudo npm install -g carto
```

Configure and build the stylesheet

Download the latest version of OpenStreetMap-Carto from source, along with the standard coastlines:

```
git clone --depth=1 -b v5.2.0 https://github.com/gravitystorm
/openstreetmap-carto.git /srv/styles/standard && cd /srv/styles
/standard && \
scripts/get-shapefiles.py
```

At this stage, you can browse to `/srv/styles/standard` and edit the files to your requirements (such as changing the colours of the roads): remember to commit any changes when you are finished, by typing:

```
carto project.mml > style.xml
```

Part 5: Connect the Tile Server and Website to Apache

Configure Renderd

In **Terminal**, open `renderd.conf` in **nano**, and change the Renderd configuration file to the following, replacing `[HOST]` with your domain name or IP address:

```
sudo nano /etc/renderd.conf
```

Contents of `/etc/renderd.conf`

```
[renderd]
stats_file=/var/run/renderd/renderd.stats
socketname=/var/run/renderd/renderd.sock
num_threads=4
tile_dir=/srv/www/tiles

;[renderd01]
;stats_file=/var/run/renderd/renderd.stats
;num_threads=4
;tile_dir=/srv/www/tiles

[mapnik]
plugins_dir=/usr/lib/mapnik/3.0/input
font_dir=/usr/share/fonts
font_dir_recurse=true

[standard]
URI=/standard/
TILEDIR=/srv/www/tiles
XML=/srv/styles/standard/style.xml
DESCRIPTION=This is the standard OpenStreetMap Mapnik style
HOST=[HOST]
;TILESIZE=256
;MINZOOM=0
MAXZOOM=19
;SCALE=1.0

;[metro]
;URI=/metro/
;TILEDIR=/srv/www/tiles
;XML=/srv/styles/metro/style.xml
;DESCRIPTION=This style shows metro and subway systems.
;HOST=[HOST]
;TILESIZE=512
;MINZOOM=0
;MAXZOOM=19
;SCALE=2.0
```

Notes

- The name of a layer (i.e. `[standard]`) is also the name of the folder that will hold the layer's tiles (i.e. `/srv/www/tiles/standard/…`).
- Renderd's default maximum zoom level (`MAXZOOM`) is `18`: OpenStreetMap uses `19`, and the standard tile layer can go up to `22` (for zoom level 22).
- For high resolution (Retina) tiles, change `TILESIZE` to `512`, then uncomment and set `SCALE` to `2.0`;
- `num_threads` should match the nearest whole gigabyte of RAM (with the minimum value being 1). For example, `{{{1}}}` for up to 2 GB RAM, `{{{1}}}` for 2 to 2.9 GB, `{{{1}}}` for 3 to 3.9 GB, and so on;
- If you have a second stylesheet, uncomment both `[renderd01]` and `[metro]`, and configure accordingly.

Next, modify Renderd's initialisation file (at `/etc/init.d/renderd`), and then restart the service. Terminal will replace `$USER` with your user name, and the warning about renderd.service not being native does not require your attention:

```
sudo sed -i -e "s/RUNASUSER=www-data/RUNASUSER=$USER/g"
/etc/init.d/renderd && \
sudo /etc/init.d/renderd restart
```

Edit the list of available layers

Modify the Leaflet configuration file, `leaflet.osm.js`, in **nano**:

```
nano /srv/www/rails/vendor/assets/leaflet/leaflet.osm.js
```

For example (replacing `[HOST]` with your domain name or IP address):

Edited extract from `leaflet.osm.js`

```
...

L.OSM.TileLayer = L.TileLayer.extend({
  options: {
    url: 'http://[HOST]/standard/{z}/{x}/{y}.png',
    attribution: '© <a href="https://www.openstreetmap.org
/copyright" target="_blank">OpenStreetMap</a> contributors'
  },

  initialize: function (options) {
    options = L.Util.setOptions(this, options);
    L.TileLayer.prototype.initialize.call(this, options.url);
  }
});

L.OSM.Mapnik = L.OSM.TileLayer.extend({
```

```
  options: {
    url: 'http://[HOST]/standard/{z}/{x}/{y}.png',
    maxZoom: 19
  }
});

...
```

Notes

- Set `maxZoom` to `20` enable zoom level 20, and so on – the standard tile layer can go up to `22` (for zoom level 22).

Next, modify the OpenLayers configuration file, `OpenStreetMap.js`:

```
nano /srv/www/rails/vendor/assets/openlayers/OpenStreetMap.js
```

For example (replacing `[HOST]` with your domain name or IP address):

Edited extract from `OpenStreetMap.js`

```
...

OpenLayers.Layer.OSM.Mapnik =
OpenLayers.Class(OpenLayers.Layer.OSM, {
    initialize: function(name, options) {
        var url = [
            "http://[HOST]/standard/${z}/${x}/${y}.png"
        ];
        options = OpenLayers.Util.extend({
            numZoomLevels: 20,
            attribution: "&copy; <a
href='https://www.openstreetmap.org/copyright'>OpenStreetMap</a>
contributors",
            buffer: 0,
            transitionEffect: "resize"
        }, options);
        var newArguments = [name, url, options];
        OpenLayers.Layer.OSM.prototype.initialize.apply(this,
newArguments);
    },
    CLASS_NAME: "OpenLayers.Layer.OSM.Mapnik"
});

...
```

Notes

- Set `numZoomLevels` to `21` to enable zoom level 20 – the standard tile layer can go up to `23` (for zoom level 22).

Finally, precompile the production assets:[1]

```
cd /srv/www/rails && \
bundle exec rake i18n:js:export assets:precompile
```

Connect the Website and Tile Server to Apache

Create a virtual host file for our Rails Port (`mapserver.conf`), and open it with **nano**:

```
sudo mv /etc/apache2/sites-available/tileserver_site.conf
/etc/apache2/sites-available/mapserver.conf && \
sudo rm /etc/apache2/sites-enabled/tileserver_site.conf && \
sudo ln -s /etc/apache2/sites-available/mapserver.conf
/etc/apache2/sites-enabled && \
sudo nano /etc/apache2/sites-available/mapserver.conf
```

Paste the following syntax into the editing window, replacing `[HOST]` with your domain name or IP address, and then save the file:[4][9][7]

Contents of `mapserver.conf`

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    ServerName [HOST]
    DocumentRoot /srv/www/rails/public
    PassengerRuby /usr/bin/ruby

    LogLevel info

    LoadTileConfigFile /etc/renderd.conf
    ModTileRequestTimeout 3
    ModTileMissingRequestTimeout 10
    ModTileMaxLoadOld 2
    ModTileMaxLoadMissing 5
    ModTileRenderdSocketName /var/run/renderd/renderd.sock

    ModTileCacheDurationMax 604800
    ModTileCacheDurationDirty 900
    ModTileCacheDurationMinimum 10800
    ModTileCacheDurationMediumZoom 13 86400
    ModTileCacheDurationLowZoom 9 518400
    ModTileCacheLastModifiedFactor 0.20
```

```
    ModTileEnableTileThrottling Off
    ModTileEnableTileThrottlingXForward 0

    ModTileThrottlingTiles 10000 1
    ModTileThrottlingRenders 128 0.2

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /srv/www/rails/public>
        Options +Indexes +FollowSymLinks -MultiViews
        Allow from all
        Require all granted
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    <Directory "/srv/www/search-ui/dist">
        DirectoryIndex search.html
        Require all granted
    </Directory>
    Alias /nominatim/ui /srv/www/search-ui/dist

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>
</VirtualHost>
```

Next, create a configuration file for Nominatim (`nominatim.conf`), and open it with **nano**: [5][7]

```
sudo nano /etc/apache2/conf-available/nominatim.conf
```

Paste the following syntax into the editing window, and then save the file:

Contents of `nominatim.conf`

```
<Directory "/srv/www/search/build/website">
    Options FollowSymLinks MultiViews
    AddType text/html   .php
    Require all granted

    RewriteEngine On

    # This must correspond to the URL where nominatim can be
found.
    RewriteBase "/nominatim/"

    # If no endpoint is given, then use search.
    RewriteRule ^(/
```

Finally, activate Nominatim, reboot the computer, and test the website by visiting your domain name or IP address. Nominatim search results and the tile server will not work yet, because their associated databases are empty.

```
sudo a2enconf nominatim && \
sudo a2enmod rewrite && \
reboot
```

Part 6: Start mapping

Tuning PostgreSQL

The default settings for PostgreSQL are too conservative if you have modern hardware. See PostgreSQL § Tune the database for more details.

To open the configuration file, type:

```
sudo nano $(ls /etc/postgresql/*/main/postgresql.conf)
```

When you finish tuning, type:

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

Create an account

First, register a new account via the new website that you set up, and save your login details to a password manager.

If you do not prefer to use an email server, you can use the Rails console (on **Terminal**) to activate your account, and then grant your new account administrator and moderator

permissions:

```
cd /srv/www/rails && bundle exec rails console
```

Next, type the following (replacing `[USER]` with your user name):

```
user = User.find_by_display_name("[USER]")
user.status = "active"
user.roles.create(:role => "administrator", :granter_id =>
user.id)
user.roles.create(:role => "moderator", :granter_id => user.id)
user.save!
quit
```

Do not forget to uncomment the OAuth key fields, otherwise they will all error out. If you are only generating one OAuth key for all applications, then grant that key all the permissions.

Create OAuth consumer keys for Potlatch 2, iD, and the Notes functionality: click here (https://github.com/openstreetmap/openstreetmap-website/blob/master/CONFIGURE.md#oauth-consumer-keys) for instructions.

Key commands

More quick commands to come.

Databases

Back up the database

Do this often: the `osm` database is irreplaceable, in sharp contrast to the `gis` database and map tiles (which you can easily recreate).

```
pg_dump osm | gzip > "OpenStreetMap $(date +'%Y-%m-%d').gz"
```

Restore the database (destructive!)

Only use this command if you have to return to a last good snapshot.

```
dropdb osm && \
createdb osm && \
gunzip -c "OpenStreetMap (YYYY-MM-DD).gz" | psql osm
```

Editing

Install and configure JOSM

Never install the often-outdated version of JOSM from the Ubuntu Software: instead, install it from the official JOSM repository:

```
echo deb https://josm.openstreetmap.de/apt $(lsb_release -sc)
universe && \
sudo tee /etc/apt/sources.list.d/josm.list > /dev/null && \
wget -q https://josm.openstreetmap.de/josm-apt.key -O- && \
sudo apt-key add - && \
sudo apt-get update && \
sudo apt install -y josm
```

Tiles

Render from database

If you are the only user of the new server, you do not need to schedule tile updates. Run the following commands in **Terminal**, replacing `[PASSWORD]` with your PostgreSQL password:

```
osmosis -q --read-apidb database="osm" user=$USER password="
[PASSWORD]" --write-pbf file="/srv/planet/planet.osm.pbf" && \
osm2pgsql -s -G -k -S /srv/styles/standard/openstreetmap-
carto.style --tag-transform-script /srv/styles/standard
/openstreetmap-carto.lua -d gis "/srv/planet/planet.osm.pbf" && \
rm -rf /srv/www/tiles/standard/*
```

Test a stylesheet

Geofabrik data dumps are useful for testing customised stylesheets before the mapping begins.

```
cd ~ && wget 'https://download.geofabrik.de/europe/greece-
latest.osm.pbf' && \
osm2pgsql -s -G -k -S /srv/styles/standard/openstreetmap-
carto.style --tag-transform-script /srv/styles/standard
/openstreetmap-carto.lua -d gis ~/greece-latest.osm.pbf && \
rm -rf /srv/www/tiles/standard/*
```

Nominatim

Initial setup

```
cd /srv/www/search && ./build/utils/setup.php --osm-file
/srv/planet/planet.osm.pbf --all
```

Updates

Upgrading

Most packages are upgradable in the same way as updating Ubuntu.

```
sudo apt-get update && \
sudo apt-get upgrade -y && \
reboot
```

Supported packages include:

- Phusion Passenger[3]

References

1. Allan, Andy (1 November 2019). "Configuration (CONFIGURE.md) § Production Deployment" (https://web.archive.org/web/20200609173159/https://github.com/openstreetmap/openstreetmap-website/blob/master/CONFIGURE.md#production-deployment). GitHub. Archived from the original (https://github.com/openstreetmap/openstreetmap-website/blob/master/CONFIGURE.md#production-deployment) on 9 June 2020. Retrieved 9 June 2020.
2. Mvexel (8 December 2017). "Installation" (https://web.archive.org/web/20180101220042/https://github.com/openstreetmap/openstreetmap-website/blob/master/INSTALL.md). GitHub. Archived from the original (https://github.com/openstreetmap/openstreetmap-website/blob/master/INSTALL.md) on 1 January 2018. Retrieved 1 January 2018.
3. "Installing Passenger + Apache on Ubuntu 18.04 LTS (with APT)" (https://web.archive.org/web/20200609170208/https://www.phusionpassenger.com/library/install/apache/install/oss/bionic/). Phusion. Archived from the original (https://www.phusionpassenger.com/library/install/apache/install/oss/bionic/) on 9 June 2020. Retrieved 9 June 2020.
4. Zerebubuth (19 March 2017). "CGImap (Readme)" (https://web.archive.org/web/20180101221402/https://github.com/openstreetmap/cgimap/blob/master/README). GitHub. Archived from the original (https://github.com/openstreetmap/cgimap/blob/master/README) on 1 January 2018. Retrieved 1 January 2018. See also: Cgimap > Install.
5. "Installation on Ubuntu 18" (https://web.archive.org/web/20200824145111/https://nominatim.org/release-docs/latest/appendix/Install-on-Ubuntu-18/). Nominatim. Archived from the original (https://nominatim.org/release-docs/latest/appendix/Install-on-Ubuntu-18/) on 24 August 2020. Retrieved 24 August 2020.
6. Henderson, Brett (27 May 2020). "Osmosis (Readme)" (https://web.archive.org/web/20200615233810/https://github.com/openstreetmap/osmosis/blob/master/README.md). GitHub. Archived from the original (https://github.com/openstreetmap/osmosis/blob/master/README.md) on 16 June 2020. Retrieved 16 June 2020.
7. "Setting up the Nominatim UI" (https://web.archive.org/web/20200825025235/https://nominatim.org/release-docs/develop/admin/Setup-Nominatim-UI/). Nominatim.

Archived from the original (https://nominatim.org/release-docs/develop/admin/Setup-Nominatim-UI/) on 25 August 2020. Retrieved 25 August 2020.

8. Ircama (21 October 2017). "Installing an OpenStreetMap Tile Server on Ubuntu" (https://web.archive.org/web/20180101212455/https://ircama.github.io/osm-carto-tutorials/tile-server-ubuntu/). GitHub. Archived from the original (https://ircama.github.io/osm-carto-tutorials/tile-server-ubuntu/) on 1 January 2018. Retrieved 1 January 2018.
9. "Deploying a Ruby application" (https://web.archive.org/web/20180109142255/https://www.phusionpassenger.com/library/deploy/apache/deploy/ruby/). Phusion. Archived from the original (https://www.phusionpassenger.com/library/deploy/apache/deploy/ruby/) on 9 January 2018. Retrieved 9 January 2018.

OpenStreetMap for Roleplaying

Tutorials	Standalone server · Standard tile layer
Projects	Arhet (https://test.geofictician.net/) · Minoa · OpenGeofiction
Background	Pokémon Go (Guidance) · Vandalism