OSM The Eurasia Ubuntu 20.04



Тайл-сервер является основным инструментом для генерации и предоставления тайлов пользователям в мире OpenStreetMap. Это набор программ и библиотек, позволяющий эффективно обрабатывать географические данные и создавать картографические тайлы, представляющие собой небольшие изображения карты для отображения на веб-сайтах или мобильных приложениях. Важно понимать, что существует несколько различных конфигураций тайл-серверов, каждая из которых имеет свои плюсы и минусы.
Руководство по запуску стандартного тайл-сервера, аналогичного тому, используемому на серверах OpenStreetMap.org, может быть полезным для тех, кто только начинает знакомиться с этой областью. Обычно такие инструкции включают в себя установку необходимого программного обеспечения, настройку базы данных геоданных, настройку веб-сервера и самого тайл-сервера, а также запуск и тестирование процесса генерации тайлов.

В результате выполнения всех необходимых шагов по установке и настройке тайл-сервера, пользователь сможет получить доступ к генерируемым тайлам через веб-приложения или мобильные устройства. Это отличное решение для тех, кто хочет создать собственный проект на основе картографических данных OpenStreetMap и обеспечить пользователям доступ к персонализированным картам.

Введение

В данном руководстве установка программного обеспечения производиться на оборудование следующей конфигурации:

- **Материнская плата Supermicro MBD-H11SSL-I**
- **Процессор AMD EPYC™ 7551P OEM**
- **Raid5 массив 1tb SSD M.2 Samsung 970 EVO Plus**
- **256GB DDR4 3200MHz DIMM ECC Reg Micron**
- **ОС Ubuntu Server 20.04 LTS**

Конфигурация вашего оборудавания будет соответственно отличаться, но это так-же будет работать, заняв больше времени при установке и рендере карт на устройстве с минимальными параметрами:

- **CPU > 4 ядер**

- **RAM > 12Gb**
- **HDD > 1Tb**

Все ниже написанное относиться к Ubuntu Server 20.04 (Focal Fossa) и протестировано в августе 2024 года.

📖 **Примечание: В этом руководстве многое загружается с альтернативных, а не с официальных зеркал, по причине скорости загрузки и неизменности файлов в репозитории для данного руководства.**

Общая теория

Tile Server OSM представляет собой набор программ и библиотек, которые работают вместе для создания тайлового (плиточного) сервера. Он состоит из 5 основных компонентов:

- **mod_tile**
- **renderd**
- **mapnik**
- **osm2pgsql**
- **postgresql/postgis**

Postgresql/postgis - базы данных. **Mod_tile** — это модуль **http сервера apache**, который обслуживает кэшированные тайлы и решает, какие тайлы нуждаются в повторном рендеринге — либо потому, что они еще не кэшированы, либо потому, что они устарели. **Mapnik** — это программная библиотека, которая осуществляет реальный рендеринг с использованием открытого рендеринга.

** 📖 Предполагается,** что вы запускаете все от пользователя **без регистрации root** через **«sudo»**. Имя пользователя без **root**, повторяемое по умолчанию, — **«eva»**. Если вы указываете иного пользователя, то вам необходимо будет заменить во всех командах на свое имя и добавить его в группу пользователей, которые могут использовать sudo для получения root-прав. Из вашей обычной учетной записи пользователя без регистрации root.

📖 **Общая последовательнось действий:**

- **Устанавливаем и обновляем Ubuntu Server**
- **Устанавливаем основное и дополнительное ПО**
- **Загружаем 6 файлов карт**
- **Объеденяем 3 файла и загружаем в базу**
- **Настраиваем веб-сервер**
- **Производим первый рендер**
- **Догружаем в базу дополнительный файл карты**
- **Проверяем работоспособность сервера**
- **Удаляем все лишнее, объединяем и загружаем в базу данных файл Евразии**

Обновление программного обеспечения

- Устанавливаем и обновляем Ubuntu

```
login as: eva
eva@192.168.1.132's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-192-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu 15 Aug 2024 11:01:00 AM UTC

  System load:  0.45              Processes:             252
  Usage of /:   0.6% of 1.71TB    Users logged in:       0
  Memory usage: 0%                IPv4 address for ens3: 192.168.1.132
  Swap usage:   0%

116 updates can be applied immediately.
40 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

- Всегда полезно обновлять программное обеспечение сервера перед выполнением какой-либо серьезной работы на сервере. Войдем на сервер через SSH и выполните следующую команду.

```
sudo apt --yes update; sudo apt --yes upgrade -y
```

```
eva@word:~$ sudo apt --yes update; sudo apt --yes upgrade -y
[sudo] password for eva:
Get:1 file:/cdrom focal InRelease
Ign:1 file:/cdrom focal InRelease
Get:2 file:/cdrom focal Release
Err:2 file:/cdrom focal Release
  File not found - /cdrom/dists/focal/Release (2: No such file or directory)
Hit:3 http://ru.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://ru.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://ru.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://ru.archive.ubuntu.com/ubuntu focal-security InRelease
Get:7 http://ru.archive.ubuntu.com/ubuntu focal/main Translation-en [506 kB]
0% [7 Translation-en 395 kB/506 kB 78%]
```

Установка необходимых библиотек

- Итак, приступим:

```
sudo apt --yes install screen libboost-all-dev git tar unzip wget bzip2
build-essential autoconf libtool libxml2-dev libgeos-dev libgeos++-dev
libpq-dev libbz2-dev libproj-dev munin-node munin protobuf-c-compiler
libfreetype6-dev libtiff5-dev libicu-dev libgdal-dev libcairo2-dev
libcairomm-1.0-dev apache2 apache2-dev libagg-dev liblua5.2-dev ttf-unifont
lua5.1 liblua5.1-0-dev
```

```
eva@word:~$ sudo apt --yes install screen libboost-all-dev git tar unzip wget bzip2 build-essential autoconf libtool
libxml2-dev libgeos-dev libgeos++-dev libpq-dev libbz2-dev libproj-dev munin-node munin protobuf-c-compiler libfreety
pe6-dev libtiff5-dev libicu-dev libgdal-dev libcairo2-dev libcairomm-1.0-dev apache2 apache2-dev libagg-dev liblua5.2
-dev ttf-unifont lua5.1 liblua5.1-0-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
bzip2 is already the newest version (1.0.8-2).
bzip2 set to manually installed.
git is already the newest version (1:2.25.1-1ubuntu3.13).
git set to manually installed.
screen is already the newest version (4.8.0-1ubuntu0.1).
screen set to manually installed.
tar is already the newest version (1.30+dfsg-7ubuntu0.20.04.4).
tar set to manually installed.
wget is already the newest version (1.20.3-1ubuntu2.1).
wget set to manually installed.
```

Установка postgresql/postgis

- В Ubuntu есть предварительно упакованные версии как **postgis**, так и **postgresql**, поэтому их можно просто установить через диспетчер пакетов Ubuntu.

```
sudo apt --yes install postgresql postgresql-contrib postgis postgresql-12-postgis-3 postgresql-12-postgis-3-scripts
```

```
eva@word:~$ sudo apt --yes install postgresql postgresql-contrib postgis postgresql-12-postgis-3 postgresql-12-postgi
s-3-scripts
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libgmpxx4ldbl libllvm10 libprotobuf-c1 libsensors-config libsensors5 libsfcgal1 postgis-doc postgresql-12
  postgresql-client-12 postgresql-client-common postgresql-common sysstat
Suggested packages:
  lm-sensors postgis-gui postgresql-doc postgresql-doc-12 libjson-perl isag
The following NEW packages will be installed:
  libgmpxx4ldbl libllvm10 libprotobuf-c1 libsensors-config libsensors5 libsfcgal1 postgis postgis-doc postgresql
  postgresql-12 postgresql-12-postgis-3 postgresql-12-postgis-3-scripts postgresql-client-12
  postgresql-client-common postgresql-common postgresql-contrib sysstat
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 39.7 MB of archives.
After this operation, 179 MB of additional disk space will be used.
0% [Working]
```

- Здесь «**postgresql**» — это база данных, в которой мы собираемся хранить данные карты, а «postgis» добавляет к ней дополнительную графическую поддержку.
- Перезапускаем **postgresql**

```
sudo service postgresql restart
```

```
eva@word:~$ sudo service postgresql restart
eva@word:~$
```

- Проверяем работу **postgresql**

```
pg_isready
```

```
postgres@word:~$ pg_isready
/var/run/postgresql:5432 - accepting connections
postgres@word:~$
```

- Создаем базу данных **postgis**. По умолчанию в различных программах предполагается, что база данных называется **gis**, и тут мы будем использовать то же соглашение, хотя это и не обязательно. Укажите/замените **«свое имя»** пользователя вместо пользователя **«eva»**, где это используется ниже. Это должно быть имя пользователя, которое будет отображать карты с помощью **Mapnik**.

```
sudo -u postgres -i
```

```
eva@word:~$ sudo -u postgres -i
postgres@word:~$
```

- Создадим пользователя (**«eva»**-пример имени пользователя для данного руководства)

```
createuser eva
```

```
postgres@word:~$ createuser eva
postgres@word:~$
```

- Создадим базу данных **gis** с кодировкой **UTF8**

```
createdb -E UTF8 -O eva gis
```

```
postgres@word:~$ createdb -E UTF8 -O eva gis
postgres@word:~$
```

- Продолжая работать как пользователь **«postgres»**, настройте PostGIS в PostgreSQL (опять же, меняя имя пользователя на **«eva»** на **«свое»** имя пользователя ниже):

```
psql
```

```
postgres@word:~$ psql
psql (12.19 (Ubuntu 12.19-0ubuntu0.20.04.1))
Type "help" for help.

postgres=#
```

- (ответ **«postgres=#»**)

```
\c gis
```

```
postgres=# \c gis
You are now connected to database "gis" as user "postgres".
gis=#
```

- (ответ: **“You are now connected to database ‘gis’ as user ‘postgres’”**.)

```
CREATE EXTENSION postgis;
CREATE EXTENSION hstore;
ALTER TABLE geometry_columns OWNER TO eva;
ALTER TABLE spatial_ref_sys OWNER TO eva;
```

```
gis=# CREATE EXTENSION postgis;
CREATE EXTENSION
gis=# CREATE EXTENSION hstore;
CREATE EXTENSION
gis=# ALTER TABLE geometry_columns OWNER TO eva;
ALTER TABLE
gis=# ALTER TABLE spatial_ref_sys OWNER TO eva;
ALTER TABLE
gis=#
```

(ответ CREATE EXTENSION)
(ответ CREATE EXTENSION)
(ответ ALTER TABLE)
(ответ ALTER TABLE)

```
\q
```

```
gis=# \q
postgres@word:~$
```

- (выход из psql и приглашение к Linux)

```
exit
```

```
postgres@word:~$ exit
logout
eva@word:~$
```

- (это вернет нас к пользователю, которым мы были до того, как выполнили **«sudo -u postgres -i»** выше)
- Если вы еще не создали пользователя, создайте пользователя Unix и для этого пользователя, выбрав пароль при появлении запроса:

```
sudo useradd -m eva
```

```
eva@word:~$ sudo useradd -m eva
[sudo] password for eva:
```

```
sudo passwd eva
```

```
eva@word:~$ sudo passwd eva
New password:
Retype new password:
passwd: password updated successfully
eva@word:~$
```

- Опять же, выше замените **“eva”** на имя пользователя без регистрации **root**, которого вы выбрали.

Установка osm2pgsql

- Далее мы установим **osm2pgsql**:

```
sudo apt install osm2pgsql
```

```
eva@word:~$ sudo apt install osm2pgsql
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- **Ошибок обычно быть не должно**

Mapnik

- Далее мы установим Mapnik, используя версию по умолчанию в Ubuntu 20.04:

```
sudo apt --yes install autoconf apache2-dev libtool libxml2-dev libbz2-dev libgeos-dev libgeos++-dev libproj-dev gdal-bin libmapnik-dev mapnik-utils python3-mapnik python3-psycopg2 python3-yaml
```

```
eva@word:~$ sudo apt --yes install autoconf apache2-dev libtool libxml2-dev libbz2-dev libgeos-dev libgeos++-dev libp
roj-dev gdal-bin libmapnik-dev mapnik-utils python3-mapnik python3-psycopg2 python3-yaml
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version (2.69-11.1).
libbz2-dev is already the newest version (1.0.8-2).
libtool is already the newest version (2.4.6-14).
libgeos++-dev is already the newest version (3.8.0-1build1).
libgeos-dev is already the newest version (3.8.0-1build1).
libproj-dev is already the newest version (6.3.1-1).
apache2-dev is already the newest version (2.4.41-4ubuntu3.21).
libxml2-dev is already the newest version (2.9.10+dfsg-5ubuntu0.20.04.7).
python3-yaml is already the newest version (5.3.1-1ubuntu0.1).
python3-yaml set to manually installed.
```

- Проверим правильность установки Mapnik:

```
python3
```

```
eva@word:~$ python3
Python 3.8.10 (default, Jul 29 2024, 17:02:10)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
import mapnik
```

```
>>> import mapnik
>>>
```

- (Ответ: >>> и без ошибок, то библиотека Mapnik была найдена Python.)

```
quit()
```

```
>>> import mapnik
>>> quit()
eva@word:~$
```

- выход

Установка mod_tile и render

- Далее мы установим mod_tile и renderd. «mod_tile» — это модуль Apache, который обрабатывает запросы на тайлы; «render» — это демон, который фактически отображает тайлы, когда «mod_tile» их запрашивает. Мы будем использовать ветку «switch2osm»
- https://github.com/SomeoneElseOSM/mod_tile, которая сама является ответвлением
- https://github.com/openstreetmap/mod_tile, но модифицирована для поддержки Ubuntu 20.04 и с пара других изменений для работы на стандартном сервере Ubuntu, а не на одном из серверов рендеринга OSM.

Скомпилируем исходный код mod_tile

- Создадим в домашнем каталоге папку **src** и перейдем в этот каталог

```
mkdir ~/src
cd ~/src
```

```
eva@word:~$ mkdir ~/src
eva@word:~$ cd ~/src
eva@word:~/src$
```

- Клонируем репозиторий **mod_tile**

Примечание: В этом руководстве mod_tile загружается с альтернативных, а не с официальных зеркал (https://github.com/SomeoneElseOSM/mod_tile.git), по причине неизменности файлов в репозитории для данного руководства.

```
git clone -b switch2osm https://git.wwoss.ru/root/mod_tile.git
```

```
eva@word:~/src$ git clone -b switch2osm https://git.wwoss.ru/root/mod_tile.git
Cloning into 'mod_tile'...
remote: Enumerating objects: 2959, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 2959 (delta 10), reused 3 (delta 3), pack-reused 2945 (from 1)
Receiving objects: 100% (2959/2959), 2.77 MiB | 31.47 MiB/s, done.
Resolving deltas: 100% (1869/1869), done.
eva@word:~/src$
```

- перейдя в каталог **mod_tile**

```
cd mod_tile
```

```
eva@word:~/src$ cd mod_tile
eva@word:~/src/mod_tile$
```

- выполним скрипт **autogen.sh**

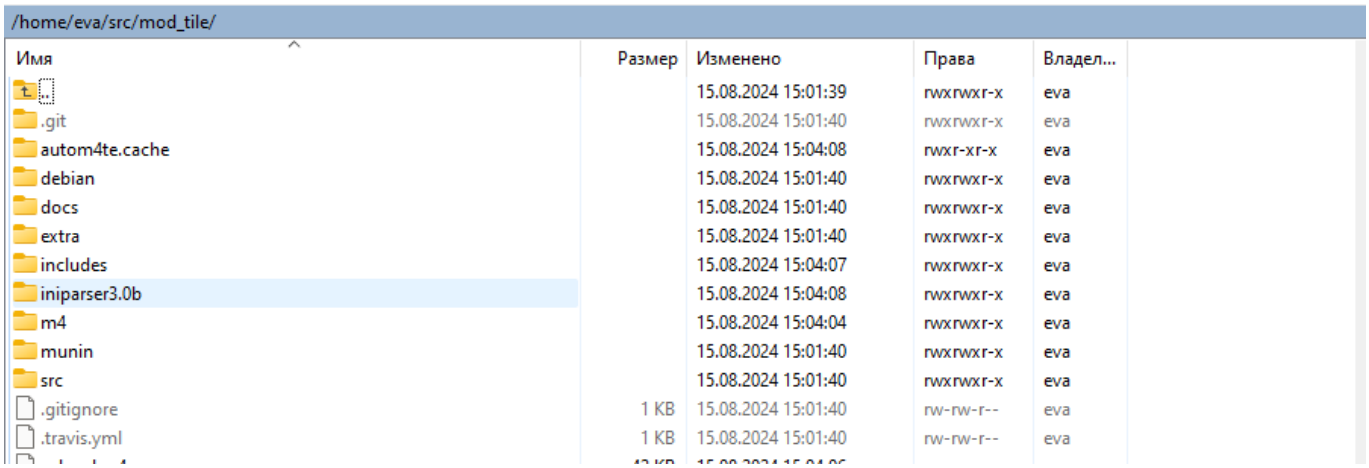
```
./autogen.sh
```

```
eva@word:~/src/mod_tile$ ./autogen.sh
autoreconf: Entering directory `.'
autoreconf: configure.ac: not using Gettext
autoreconf: running: aclocal --force -I m4
```

- (это должно закончится **“autoreconf: Leaving directory ‘.’”.**)

```
autoreconf: running: automake --add-missing --copy --force-missing
configure.ac:8: installing './compile'
configure.ac:8: installing './config.guess'
configure.ac:8: installing './config.sub'
configure.ac:7: installing './install-sh'
configure.ac:7: installing './missing'
Makefile.am: installing './depcomp'
autoreconf: Leaving directory `.'
eva@word:~/src/mod_tile$
```

- (в предыдущих установках мне приходилось проделывать это 2 раза для создания папки **mod_tile** в директории **src**)



- Сконфигурируем

```
./configure
```

```
eva@word:~/src/mod_tile$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking how to print strings... printf
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g...
```

- (это должно закончится **“config.status: executing libtool commands”**)

```
configure: creating ./config.status
config.status: creating iniparser3.0b/Makefile
config.status: creating Makefile
config.status: creating includes/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
eva@word:~/src/mod_tile$
```

```
make
```

```
eva@word:~/src/mod_tile$ make
Making all in iniparser3.0b
make[1]: Entering directory '/home/eva/src/mod_tile/iniparser3.0b'
depbase=`echo src/iniparser.lo | sed 's|[^/]*$|.deps/&|;s|\.lo$||'`;\
/bin/bash ../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I../includes     -g -O2 -MT src/iniparser.lo
 -MD -MP -MF $depbase.Tpo -c -o src/iniparser.lo src/iniparser.c &&\
mv -f $depbase.Tpo $depbase.Plo
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../includes -g -O2 -MT src/iniparser.lo -MD -MP -MF src/.deps/iniparser.
Tpo -c src/iniparser.c  -fPIC -DPIC -o src/.libs/iniparser.o
src/iniparser.c: In function 'iniparser_load':
src/iniparser.c:600:30: warning: '%s' directive writing up to 1024 bytes into a region of size between 0 and 1024 [-W
format-overflow=]
  600 |             sprintf(tmp, "%s:%s", section, key);
      |                              ^~           ~~~
In file included from /usr/include/stdio.h:867,
                 from src/iniparser.h:24,
                 from src/iniparser.c:18:
/usr/include/x86_64-linux-gnu/bits/stdio2.h:36:10: note: '__builtin___sprintf_chk' output between 2 and 2050 bytes in
to a destination of size 1025
   36 |   return __builtin___sprintf_chk (__s, __USE_FORTIFY_LEVEL - 1,
      |          ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   37 |        __bos (__s), __fmt, __va_arg_pack ());
      |        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../includes -g -O2 -MT src/iniparser.lo -MD -MP -MF src/.deps/iniparser.
Tpo -c src/iniparser.c -o src/iniparser.o >/dev/null 2>&1
```

- Обратите внимание, что здесь некоторые «тревожные» сообщения будут прокручиваться вверх по экрану. Однако это должно заканчиваться на **«make[1]: Leaving directory '/home/eva/src/mod_tile'»**

```
ame -Wl,mod_tile.so -o ./src/.libs/mod_tile.so
libtool: link: ( cd "./src/.libs" && rm -f "mod_tile.la" && ln -s "../mod_tile.la" "mod_tile.la" )
make[1]: Leaving directory '/home/eva/src/mod_tile'
eva@word:~/src/mod_tile$
```

```
sudo make install
```

```
eva@word:~/src/mod_tile$ sudo make install
[sudo] password for eva:
Making install in iniparser3.0b
make[1]: Entering directory '/home/eva/src/mod_tile/iniparser3.0b'
make[2]: Entering directory '/home/eva/src/mod_tile/iniparser3.0b'
 /usr/bin/mkdir -p '/usr/local/lib'
 /bin/bash ../libtool   --mode=install /usr/bin/install -c   libiniparser.la '/usr/local/lib'
libtool: install: /usr/bin/install -c .libs/libiniparser.so.3.0.0 /usr/local/lib/libiniparser.so.3.0.0
libtool: install: (cd /usr/local/lib && { ln -s -f libiniparser.so.3.0.0 libiniparser.so.3 || { rm -f libiniparser.so
.3 && ln -s libiniparser.so.3.0.0 libiniparser.so.3; }; })
libtool: install: (cd /usr/local/lib && { ln -s -f libiniparser.so.3.0.0 libiniparser.so || { rm -f libiniparser.so &
& ln -s libiniparser.so.3.0.0 libiniparser.so; }; })
libtool: install: /usr/bin/install -c .libs/libiniparser.lai /usr/local/lib/libiniparser.la
libtool: install: /usr/bin/install -c .libs/libiniparser.a /usr/local/lib/libiniparser.a
libtool: install: chmod 644 /usr/local/lib/libiniparser.a
libtool: install: ranlib /usr/local/lib/libiniparser.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/sbin" ldconfig -n /usr
/local/lib
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/local/lib
```

- (это должно закончиться **«make[1]: выход из каталога должно быть '/home/eva/src/mod_tile'»**)

```
libtool: install: /usr/bin/install -c .libs/renderd /usr/local/bin/renderd
libtool: install: /usr/bin/install -c render_expired /usr/local/bin/render_expired
libtool: install: /usr/bin/install -c render_list /usr/local/bin/render_list
libtool: install: /usr/bin/install -c render_speedtest /usr/local/bin/render_speedtest
libtool: install: /usr/bin/install -c render_old /usr/local/bin/render_old
 /usr/bin/mkdir -p '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 docs/render_expired.1 docs/render_list.1 docs/render_old.1 docs/render_speedtest.1 '/usr/
local/share/man/man1'
 /usr/bin/mkdir -p '/usr/local/share/man/man8'
 /usr/bin/install -c -m 644 docs/renderd.8 '/usr/local/share/man/man8'
 /usr/bin/mkdir -p '/usr/local/etc'
 /usr/bin/install -c -m 644 renderd.conf '/usr/local/etc'
make[2]: Leaving directory '/home/eva/src/mod_tile'
make[1]: Leaving directory '/home/eva/src/mod_tile'
eva@word:~/src/mod_tile$
```

```
sudo make install-mod_tile
```

```
eva@word:~/src/mod_tile$ sudo make install-mod_tile
mkdir -p `apxs -q LIBEXECDIR`
apxs -S LIBEXECDIR=`apxs -q LIBEXECDIR` -c -i   -I./includes    -lcurl ./src/mod_tile.c ./src/sys_utils.c ./src/store
.c ./src/store_file.c ./src/store_file_utils.c ./src/store_memcached.c ./src/store_rados.c ./src/store_ro_http_proxy.
c ./src/store_ro_composite.c ./src/store_null.c
/usr/share/apr-1.0/build/libtool  --mode=compile --tag=disable-static x86_64-linux-gnu-gcc -prefer-pic -pipe -g -O2 -
fstack-protector-strong -Wformat -Werror=format-security  -Wdate-time -D_FORTIFY_SOURCE=2   -DLINUX -D_REENTRANT -D_G
NU_SOURCE  -pthread  -I/usr/include/apache2  -I/usr/include/apr-1.0   -I/usr/include/apr-1.0 -I/usr/include -I./inclu
des  -c -o ./src/mod_tile.lo ./src/mod_tile.c && touch ./src/mod_tile.slo
libtool: compile:  x86_64-linux-gnu-gcc -pipe -g -O2 -fstack-protector-strong -Wformat -Werror=format-security -Wdate
-time -D_FORTIFY_SOURCE=2 -DLINUX -D_REENTRANT -D_GNU_SOURCE -pthread -I/usr/include/apache2 -I/usr/include/apr-1.0 -
I/usr/include/apr-1.0 -I/usr/include -I./includes -c ./src/mod_tile.c  -fPIC -DPIC -o ./src/.libs/mod_tile.o
```

- (это должно закончиться **«chmod 644 /usr/lib/apache2/modules/mod_tile.so»**)

```
/lib/apache2/modules
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/lib/apache2/modules

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the 'LD_RUN_PATH' environment variable
     during linking
   - use the '-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
chmod 644 /usr/lib/apache2/modules/mod_tile.so
eva@word:~/src/mod_tile$
```

```
sudo ldconfig
```

```
eva@word:~/src/mod_tile$ sudo ldconfig
eva@word:~/src/mod_tile$
```

- **(без ответа)**

Конфигурация таблицы стилей

- Теперь, когда все необходимое программное обеспечение установлено, вам нужно загрузить и настроить таблицу стилей.
- Здесь мы будем использовать тот же стиль, что и на «стандартной» карте на веб-сайте openstreetmap.org.
- Переходим в домашний каталог.

```
cd ~/src
```

```
eva@word:~/src/mod_tile$ cd ~/src
eva@word:~/src$
```

- клонируем репозитарий

Примечание: В этом руководстве openstreetmap-carto загружается с альтернативных, а не с официальных зеркал (https://github.com/gravitystorm/openstreetmap-carto), по причине неизменности файлов openstreetmap-carto в репозитории для данного руководства.

```
git clone https://git.wwoss.ru/root/openstreetmap-carto.git
```

- клонируем репозитарий

```
eva@word:~/src$ git clone https://git.wwoss.ru/root/openstreetmap-carto.git
Cloning into 'openstreetmap-carto'...
remote: Enumerating objects: 18194, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 18194 (delta 7), reused 4 (delta 4), pack-reused 18185 (from 1)
Receiving objects: 100% (18194/18194), 17.27 MiB | 37.00 MiB/s, done.
Resolving deltas: 100% (12571/12571), done.
eva@word:~/src$
```

- переходим в каталог openstreetmap-carto

```
cd openstreetmap-carto
```

```
eva@word:~/src$ cd openstreetmap-carto
eva@word:~/src/openstreetmap-carto$
```

- установим подходящую версию компилятора «carto».

```
sudo apt --yes install npm
```

```
eva@word:~/src/openstreetmap-carto$ sudo apt --yes install npm
[sudo] password for eva:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- размер в 600mb, т.ч. перекур

```
sudo npm install -g carto
```

```
eva@word:~/src/openstreetmap-carto$ sudo npm install -g carto
/usr/local/bin/carto -> /usr/local/lib/node_modules/carto/bin/carto
+ carto@1.2.0
added 64 packages from 29 contributors in 9.568s
eva@word:~/src/openstreetmap-carto$
```

- Проверим версию программы

```
carto -v
```

- Это должно ответить номером версии, которая должна быть не меньше: **1.2.0**

```
eva@word:~/src/openstreetmap-carto$ carto -v
1.2.0
eva@word:~/src/openstreetmap-carto$
```

- преобразуем проект carto во что-то, понятное Mapnik:

```
carto project.mml > mapnik.xml
```

```
eva@word:~/src/openstreetmap-carto$ carto project.mml > mapnik.xml
Warning: style/landcover.mss:628:4 line-offset is unstable. It may change in the future.
Warning: style/landcover.mss:631:6 line-offset is unstable. It may change in the future.
Warning: style/water-features.mss:112:6 line-offset is unstable. It may change in the future.
Warning: style/water-features.mss:114:6 line-offset is unstable. It may change in the future.
Warning: style/water-features.mss:99:4 line-offset is unstable. It may change in the future.
Warning: style/water-features.mss:105:4 line-offset is unstable. It may change in the future.
Warning: style/landcover.mss:841:6 line-offset is unstable. It may change in the future.
Warning: style/landcover.mss:838:4 line-offset is unstable. It may change in the future.
```

- Теперь у вас есть таблица стиля Mapnik XML в /home/eva/src/openstreetmap-carto/mapnik.xml.

📖 **Примечание:** все предупреждения
Warning: style/admin.mss:22:18 Styles do not match layer selector #admin-low-zoom.
Warning: style/admin.mss:18:6 Styles do not match layer selector #admin-low-zoom.
Игнорируем

/home/eva/src/openstreetmap-carto/

Имя	Размер	Изменено	Права	Владел...
CODE_OF_CONDUCT.md	6 KB	15.08.2024 15:42:03	rw-rw-r--	eva
CONTRIBUTING.md	9 KB	15.08.2024 15:42:03	rw-rw-r--	eva
DOCKER.md	11 KB	15.08.2024 15:42:03	rw-rw-r--	eva
docker-compose.yml	1 KB	15.08.2024 15:42:03	rw-rw-r--	eva
Dockerfile	2 KB	15.08.2024 15:42:03	rw-rw-r--	eva
Dockerfile.db	1 KB	15.08.2024 15:42:03	rw-rw-r--	eva
Dockerfile.import	1 KB	15.08.2024 15:42:03	rw-rw-r--	eva
external-data.yml	4 KB	15.08.2024 15:42:03	rw-rw-r--	eva
indexes.sql	3 KB	15.08.2024 15:42:03	rw-rw-r--	eva
indexes.yml	2 KB	15.08.2024 15:42:03	rw-rw-r--	eva
INSTALL.md	6 KB	15.08.2024 15:42:03	rw-rw-r--	eva
LICENSE.txt	1 KB	15.08.2024 15:42:03	rw-rw-r--	eva
mapnik.xml	3 216 KB	15.08.2024 15:53:04	rw-rw-r--	eva
openstreetmap-carto.lua	13 KB	15.08.2024 15:42:03	rw-rw-r--	eva
openstreetmap-carto.style	3 KB	15.08.2024 15:42:03	rw-rw-r--	eva
preview.png	361 KB	15.08.2024 15:42:03	rw-rw-r--	eva

Загрузка данных

- в домашнем каталоге создаем папку data и переходим в эту папку

```
mkdir ~/data
cd ~/data
```

```
eva@word:~/src/openstreetmap-carto$ mkdir ~/data
eva@word:~/src/openstreetmap-carto$ cd ~/data
eva@word:~/data$
```

- на странице зеркала загрузки https://wwoss.ru/openstreetmap/ выбираем нужную нам карту для и копируем ее ссылку

📖 **Примечание: В этом руководстве файлы «.osm.pbf» загружается с альтернативных, а не с официальных зеркал (https://download.geofabrik.de/), по причине скорости**

загрузки с данного репозитория для данного руководства.

Последовательность: Сейчас мы загрузим 3 небольшие карты (georgia, moldova и azerbaijan), произведем их слияние (для понимания и наглядного примера слияния) и с ними продолжим установку и настройку сервера. Так-же загрузим небольшую карту (andorra) для примера дозагрузки карт в базу и конечно карты Europe (28.9 GB) и Asia (13.1 GB). Europe и Asia поставим на слияние и загрузим в базу в конце статьи, на уже подготовленный нами и проверенный (рабочий) сервер.

```
wget
https://wwoss.ru/openstreetmap/europe/europe-sub-region/georgia-latest.osm.p
bf
wget
https://wwoss.ru/openstreetmap/europe/europe-sub-region/moldova-latest.osm.p
bf
wget
https://wwoss.ru/openstreetmap/asia/asia-sub-region/azerbaijan-latest.osm.pb
f

wget
https://wwoss.ru/openstreetmap/europe/europe-sub-region/andorra-latest.osm.p
bf

wget https://wwoss.ru/openstreetmap/europe/europe-latest.osm.pbf
wget https://wwoss.ru/openstreetmap/asia/asia-latest.osm.pbf
```

```
azerbaijan-latest.osm.pbf     100%[=============================================>]  37.46M  73.7MB/s    in 0.5s

2024-08-18 16:26:49 (73.7 MB/s) - 'azerbaijan-latest.osm.pbf' saved [39275141/39275141]

eva@word:~/data$
eva@word:~/data$ wget https://wwoss.ru/openstreetmap/europe/europe-sub-region/andorra-latest.osm.pbf
--2024-08-18 16:26:49--  https://wwoss.ru/openstreetmap/europe/europe-sub-region/andorra-latest.osm.pbf
Resolving wwoss.ru (wwoss.ru)... 78.36.10.192
Connecting to wwoss.ru (wwoss.ru)|78.36.10.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2679565 (2.6M) [application/vnd.openstreetmap.data+xml]
Saving to: 'andorra-latest.osm.pbf'

andorra-latest.osm.pbf        100%[=============================================>]   2.55M  --.-KB/s    in 0.04s

2024-08-18 16:26:49 (59.0 MB/s) - 'andorra-latest.osm.pbf' saved [2679565/2679565]

eva@word:~/data$
eva@word:~/data$ wget https://wwoss.ru/openstreetmap/europe/europe-latest.osm.pbf
--2024-08-18 16:26:49--  https://wwoss.ru/openstreetmap/europe/europe-latest.osm.pbf
Resolving wwoss.ru (wwoss.ru)... 78.36.10.192
Connecting to wwoss.ru (wwoss.ru)|78.36.10.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30961314293 (29G) [application/vnd.openstreetmap.data+xml]
Saving to: 'europe-latest.osm.pbf'

europe-latest.osm.pbf          6%[==>                                        ]   1.90G  71.7MB/s    eta 6m 37s
```

- карты занимают место размером примерно 40Gb, процесс не быстрый (минут 20..), идем гулять с собачкой...

/home/eva/data/

Имя	Размер	Изменено	Права	Владел...
..		18.08.2024 19:06:36	rwxr-xr-x	eva
andorra-latest.osm.pbf	2 617 KB	27.07.2024 4:09:09	rw-rw-r--	eva
asia-latest.osm.pbf	13 676 996 KB	27.07.2024 2:04:55	rw-rw-r--	eva
azerbaijan-latest.osm.pbf	38 355 KB	27.07.2024 5:17:53	rw-rw-r--	eva
europe-latest.osm.pbf	30 235 659 KB	27.07.2024 2:04:55	rw-rw-r--	eva
georgia-latest.osm.pbf	82 573 KB	27.07.2024 5:01:30	rw-rw-r--	eva
moldova-latest.osm.pbf	71 741 KB	27.07.2024 4:43:52	rw-rw-r--	eva

- Произведем слияние карт (georgia, moldova и azerbaijan) для наглядного примера и понимани, как это работает.
- Установим **osmctools**

```
sudo apt install osmctools
```

```
eva@word:~/data$ sudo apt install osmctools
[sudo] password for eva:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  osmctools
0 upgraded, 1 newly installed, 0 to remove and 173 not upgraded.
Need to get 151 kB of archives.
After this operation, 431 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu focal/universe amd64 osmctools amd64 0.9-2 [151 kB]
Fetched 151 kB in 0s (567 kB/s)
```

- Используем **Osmconvert** и объединим georgia с moldova в один pbf файл с названием для примера «**new-latest**»:

```
cd ~/data
osmconvert georgia-latest.osm.pbf --out-o5m | osmconvert - moldova-
latest.osm.pbf -o=new-latest.osm.pbf
```

```
eva@word:~/data$ osmconvert georgia-latest.osm.pbf --out-o5m | osmconvert - moldova-latest.osm.pbf -o=new-latest.osm.
pbf
eva@word:~/data$
```

- Как мы можем видеть, у нас получился новый файл карты с названием **new-latest.osm.pbf** размером 147Mb.

/home/eva/data/

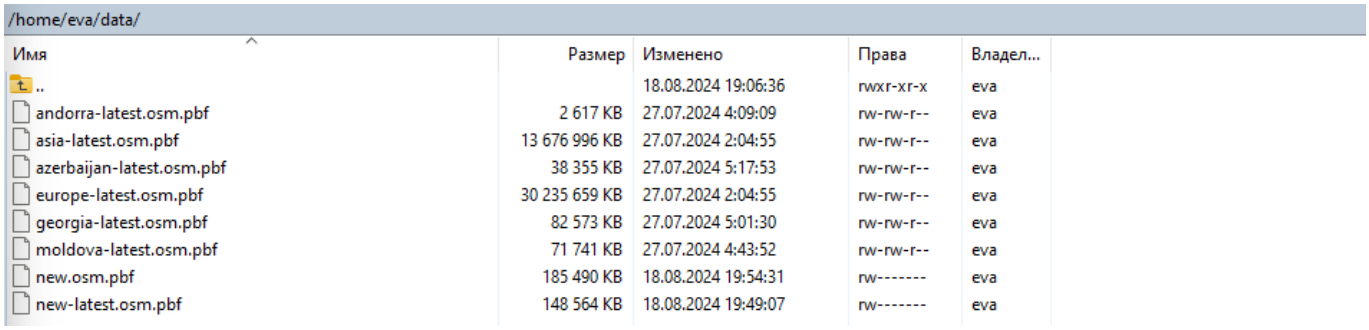
Имя	Размер	Изменено	Права	Владел...
..		18.08.2024 19:06:36	rwxr-xr-x	eva
andorra-latest.osm.pbf	2 617 KB	27.07.2024 4:09:09	rw-rw-r--	eva
asia-latest.osm.pbf	13 676 996 KB	27.07.2024 2:04:55	rw-rw-r--	eva
azerbaijan-latest.osm.pbf	38 355 KB	27.07.2024 5:17:53	rw-rw-r--	eva
europe-latest.osm.pbf	30 235 659 KB	27.07.2024 2:04:55	rw-rw-r--	eva
georgia-latest.osm.pbf	82 573 KB	27.07.2024 5:01:30	rw-rw-r--	eva
moldova-latest.osm.pbf	71 741 KB	27.07.2024 4:43:52	rw-rw-r--	eva
new-latest.osm.pbf	148 564 KB	18.08.2024 19:49:07	rw-------	eva

- А теперь файл карты azerbaijan объединим с нашим вновь созданным файлом **new-latest**

```
osmconvert azerbaijan-latest.osm.pbf --out-o5m | osmconvert - new-
latest.osm.pbf -o=new.osm.pbf
```

```
eva@word:~/data$ osmconvert azerbaijan-latest.osm.pbf --out-o5m | osmconvert - new-latest.osm.pbf -o=new.osm.pbf
eva@word:~/data$
```

- Как мы можем видеть, у нас получился новый файл карты с названием «new» (new.osm.pbf) размером 185Mb. Именно его мы на данном этапе и будем загружать в базу данных.


/home/eva/data/

Имя	Размер	Изменено	Права	Владел...
..		18.08.2024 19:06:36	rwxr-xr-x	eva
andorra-latest.osm.pbf	2 617 KB	27.07.2024 4:09:09	rw-rw-r--	eva
asia-latest.osm.pbf	13 676 996 KB	27.07.2024 2:04:55	rw-rw-r--	eva
azerbaijan-latest.osm.pbf	38 355 KB	27.07.2024 5:17:53	rw-rw-r--	eva
europe-latest.osm.pbf	30 235 659 KB	27.07.2024 2:04:55	rw-rw-r--	eva
georgia-latest.osm.pbf	82 573 KB	27.07.2024 5:01:30	rw-rw-r--	eva
moldova-latest.osm.pbf	71 741 KB	27.07.2024 4:43:52	rw-rw-r--	eva
new.osm.pbf	185 490 KB	18.08.2024 19:54:31	rw-------	eva
new-latest.osm.pbf	148 564 KB	18.08.2024 19:49:07	rw-------	eva

- Следующая команда вставит загруженные ранее данные **new.osm.pbf** в базу данных. Этот шаг очень интенсивно использует дисковый ввод-вывод; импорт нашего примера не займет много времени, в отличие от загрузки Europe и Asia, где будет использованно несколько дней или недель в зависимости от аппаратного обеспечения и это мы сделаем в конце этой статьи.

```
cd ~/data
osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 64000 --number-
processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style
~/data/new.osm.pbf
```

```
eva@word:~/data$ osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script ~/src/openstreetmap-carto/opens
treetmap-carto.lua -C 64000 --number-processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/new.os
m.pbf
osm2pgsql version 1.2.1 (64 bit id space)

Allocating memory for dense node cache
Allocating dense node cache in one big chunk
Allocating memory for sparse node cache
Sharing dense sparse
Node-cache: cache=64000MB, maxblocks=1024000*65536, allocation method=11
Mid: pgsql, cache=64000
Setting up table: planet_osm_nodes
Setting up table: planet_osm_ways
Setting up table: planet_osm_rels
Using lua based tag processing pipeline with script /home/eva/src/openstreetmap-carto/openstreetmap-carto.lua
Using projection SRS 3857 (Spherical Mercator)
Setting up table: planet_osm_point
Setting up table: planet_osm_line
Setting up table: planet_osm_polygon
Setting up table: planet_osm_roads

Reading in file: /home/eva/data/new.osm.pbf
Using PBF parser.
```

- **Обратите внимание на параметр -c (-create) при импорте через osm2pgsql. * С этим параметром удаляется вся существующая информация с таблиц.**

📖 Дополнение: Откроем второе окно программы Putty и поставим слияник карт Europe и Asia, чтобы к концу статьи у нас уже был объеденненный файл «eurasia.osm.pbf»

```
cd ~/data
osmconvert europe-latest.osm.pbf --out-o5m | osmconvert - asia-
```

```
latest.osm.pbf -o=eurasia.osm.pbf
```

```
eva@word:~$ cd ~/data
eva@word:~/data$ osmconvert europe-latest.osm.pbf --out-o5m | osmconvert - asia-latest.osm.pbf -o=eurasia.osm.pbf
```

Догрузка карт в базу данных

Внимание!: Этот шаг выполняеся только в случае необходимости дозагрузки в готовую базу данных дополнительных файлов, в нашем случае сейчас пропустим этот подраздел, перейдем в радел "Создание индексов" и вернемся сюда в конце статьи при уже настроенном сервере. (этот шаг указан в этом разделе т.к команды эдинтичные с разницей в способе загрузки (--create/--append)).

- Следующая команда загрузит файл карты «andorra» в уже существующую базу данных.

```
cd ~/data
osm2pgsql -d gis --append --slim  -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 64000 --number-
processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style
~/data/andorra-latest.osm.pbf
```

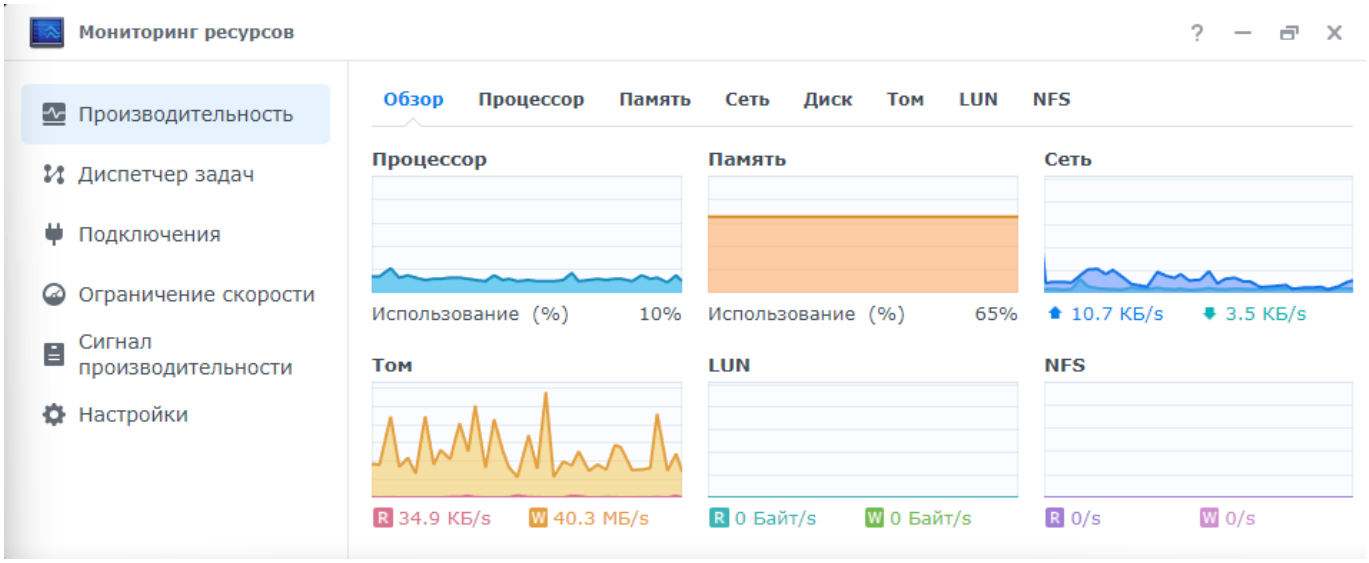
- **Если вам необходимо сделать импорт ещё нескольких стран, то вместо -c указывайте параметр -a (–append).**

Командные опции:

- -d gis
- База данных, с которой нужно работать (раньше gis была по умолчанию, теперь нужно указать).
- –create
- Загружает данные в пустую базу данных, а не пытается добавить их в существующую (иначе –append).
- –slim
- osm2pgsql может использовать разные макеты таблиц; «тонкие» таблицы работают для рендеринга.
- -G
- Определяет, как обрабатываются мультиполигоны.
- –hstore
- Позволяет тегам, для которых нет явных столбцов базы данных, использоваться для рендеринга.
- –tag-transform-script
- Определяет сценарий lua, используемый для обработки тегов. Это простой способ обработки тегов OSM до того, как их обработает сам стиль, что значительно упрощает логику стиля.
- -C 4000
- Выделено 4,0 Гб памяти osm2pgsql для процесса импорта. Если у вас меньше памяти, вы можете попробовать меньшее число, а если процесс импорта завершится из-за нехватки памяти, вам нужно будет попробовать меньшее число или меньший размер карты OSM.
- –number-processes 2
- Используйте 2 ЦП. Если у вас есть больше доступных ядер, вы можете использовать

больше.
- -S
- Создайте столбцы базы данных в этом файле (на самом деле они не изменились по сравнению с «openstreetmap-carto»)
- ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/planet-latest.osm.pbf
- Место сохранения и файл данных для загрузки.

- смотрим монитор ресурсов Synology NAS



- Команда загрузки объеденненных 3х файлов карт в единый файл new.osm.pbf

должна завершится чем-то вроде «Osm2pgsql в целом занял 312 секунд».

```
Osm2pgsql took 312s overall
node cache: stored: 26012186(100.00%), storage efficiency: 51.45% (dense blocks: 303, sparse nodes: 24036290), hit ra
te: 100.00%
eva@word:~/data$
```

- А если вы сразу грузите карту eurarasia размером 40gb (которая по факту заняла 07 часов 23 минуты), то перейдем к разделу Шрифты, открыв второе окно подключения к серверу.

Создание индексов

- Начиная с версии v5.3.0 некоторые дополнительные индексы теперь необходимо применять вручную .
- переходим в папку openstreetmap-carto

```
cd ~/src/openstreetmap-carto/
```

```
eva@word:~/data$ cd ~/src/openstreetmap-carto/
eva@word:~/src/openstreetmap-carto$
```

- запускаем команду создание индексов

```
scripts/get-external-data.py
```

```
eva@word:~/data$ cd ~/src/openstreetmap-carto/
eva@word:~/src/openstreetmap-carto$ psql -d gis -f indexes.sql
CREATE INDEX
```

- **(ответ: «CREATE INDEX» 16 раз.)**

```
eva@word:~/src/openstreetmap-carto$ psql -d gis -f indexes.sql
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
eva@word:~/src/openstreetmap-carto$
```

Загрузка Shape-файла

- Shape-файл — векторный формат для хранения объектов, описываемых геометрией и сопутствующими атрибутами.
- Несмотря на то, что большая часть данных для создания карты, берется из данных OpenStreetMap, загружается не все, некоторые шейп-файлы для таких вещей, как границы стран с малым масштабированием, по-прежнему необходимы. Чтобы загрузить и проиндексировать их:
- переходим в папку openstreetmap-carto

```
cd ~/src/openstreetmap-carto/
```

- и выполным команду

```
scripts/get-external-data.py
```

```
eva@word:~/src/openstreetmap-carto$ scripts/get-external-data.py
INFO:root:Starting load of external data into database
INFO:root:Checking table simplified_water_polygons
```

- Этот процесс включает в себя загрузку и требует некоторого времени — во время работы на экране мало что освещает. На самом деле он загружает таблица водных полигонов и прочее и заполняет каталог «data» ниже «openstreetmap-carto».
- это занято примерно 5-7 минут т.к. пока наша база данных не большая на это уйдет пара минут

```
eva@word:~/src/openstreetmap-carto$ scripts/get-external-data.py
INFO:root:Starting load of external data into database
INFO:root:Checking table simplified_water_polygons
INFO:root:  Download complete (23861973 bytes)
INFO:root:  Decompressing file
INFO:root:  Importing into database
INFO:root:  Import complete
INFO:root:Checking table water_polygons
INFO:root:  Download complete (864615701 bytes)
INFO:root:  Decompressing file
INFO:root:  Importing into database
INFO:root:  Import complete
INFO:root:Checking table icesheet_polygons
INFO:root:  Download complete (52420604 bytes)
INFO:root:  Decompressing file
INFO:root:  Importing into database
INFO:root:  Import complete
INFO:root:Checking table icesheet_outlines
INFO:root:  Download complete (53083621 bytes)
INFO:root:  Decompressing file
INFO:root:  Importing into database
INFO:root:  Import complete
INFO:root:Checking table ne_110m_admin_0_boundary_lines_land
INFO:root:  Download complete (57325 bytes)
INFO:root:  Decompressing file
INFO:root:  Importing into database
INFO:root:  Import complete
eva@word:~/src/openstreetmap-carto$
```

Шрифты

- Переходим в каталог **openstreetmap-carto** в каталоге **src**

```
cd ~/src/openstreetmap-carto/
```

```
eva@word:~$ cd ~/src/openstreetmap-carto/
eva@word:~/src/openstreetmap-carto$
```

- и собственно загружаем шрифты

```
scripts/get-fonts.sh
```

```
eva@word:~/src/openstreetmap-carto$ scripts/get-fonts.sh
Warning: Illegal date format for -z, --time-cond (and not a file name).
Warning: Disabling time condition. See curl_getdate(3) for valid date syntax.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  555k  100  555k    0     0   558k      0 --:--:-- --:--:-- --:--:-- 4006k
Warning: Illegal date format for -z, --time-cond (and not a file name).
Warning: Disabling time condition. See curl_getdate(3) for valid date syntax.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
```

- Процесс займет примерно 6 минут

```
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 16.2M  100 16.2M    0     0  4796k      0  0:00:03  0:00:03 --:--:-- 9842k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 21837    0 21837    0     0  85972      0 --:--:-- --:--:-- --:--:-- 85972
[/tmp/get-fonts.bM7uCidfs/Noto_Emoji.zip]
  End-of-central-directory signature not found.  Either this file is not
  a zipfile, or it constitutes one disk of a multi-part archive.  In the
  latter case the central directory and zipfile comment will be found on
  the last disk(s) of this archive.
unzip:  cannot find zipfile directory in one of /tmp/get-fonts.bM7uCidfs/Noto_Emoji.zip or
        /tmp/get-fonts.bM7uCidfs/Noto_Emoji.zip.zip, and cannot find /tmp/get-fonts.bM7uCidfs/Noto_Emoji.zip.ZIP, per
iod.
eva@word:~/src/openstreetmap-carto$
```

Настройка веб-сервера

визуализация

- Отрываем с помощью текстового редактора **nano** файл конфигурации для «render» — **«/usr/local/etc/renderd.conf»**

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

- редактируем строку XML=/home/renderaccount/src/openstreetmap-carto/mapnik.xml меняя renderaccount на имя своего пользователя (в нашем случае это - eva)

```
  GNU nano 4.8                          /usr/local/etc/renderd.conf                          Modified
[renderd]
num_threads=4
tile_dir=/var/lib/mod_tile
stats_file=/var/run/renderd/renderd.stats

[mapnik]
plugins_dir=/usr/lib/mapnik/3.0/input
font_dir=/usr/share/fonts/truetype
font_dir_recurse=1

[ajt]
URI=/hot/
TILEDIR=/var/lib/mod_tile
XML=/home/eva/src/openstreetmap-carto/mapnik.xml
HOST=localhost
TILESIZE=256
MAXZOOM=20
```

- нажимаем последовательно CTRL + O для сохранения файла
- нажимаем ENTER для подтверждения сохранения
- нажимаем последовательно CTRL + X для выхода из редактора

Настройка Apache

- создаем каталог mod_tile в директории /var/lib/ нашего сервера

```
sudo mkdir /var/lib/mod_tile
```

```
eva@word:~/src/openstreetmap-carto$ sudo mkdir /var/lib/mod_tile
[sudo] password for eva:
eva@word:~/src/openstreetmap-carto$
```

- задаем права пользователя для этого каталога

```
sudo chown eva /var/lib/mod_tile
```

```
eva@word:~/src/openstreetmap-carto$ sudo chown eva /var/lib/mod_tile
eva@word:~/src/openstreetmap-carto$
```

- создаем каталог renderd

```
sudo mkdir /var/run/renderd
```

```
eva@word:~/src/openstreetmap-carto$ sudo mkdir /var/run/renderd
eva@word:~/src/openstreetmap-carto$
```

- задаем права пользователя для этой папки

```
sudo chown eva /var/run/renderd
```

```
eva@word:~/src/openstreetmap-carto$ sudo chown eva /var/run/renderd
eva@word:~/src/openstreetmap-carto$
```

- сообщаем Apache о «mod_tile» редактируя файл mod_tile.conf добавляя следующую строку **LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so**

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

```
eva@word:~/src/openstreetmap-carto$ sudo nano /etc/apache2/conf-available/mod_tile.conf
```

- Вставляем нижеприведенную строку в тело файла (в нашем случае это пустой файл)

```
LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so
```

```
  GNU nano 4.8                    /etc/apache2/conf-available/mod_tile.conf                    Modified
LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo
```

- нажимаем последовательно CTRL + O для сохранения файла
- нажимаем ENTER для подтверждения сохранения
- нажимаем последовательно CTRL + X для выхода из редактора

```
eva@word:~/src/openstreetmap-carto$ sudo nano /etc/apache2/conf-available/mod_tile.conf
eva@word:~/src/openstreetmap-carto$
```

- выполняем команду

```
sudo a2enconf mod_tile
```

```
eva@word:~/src/openstreetmap-carto$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
apache2.service is not active, cannot reload.
eva@word:~/src/openstreetmap-carto$
```

- сообщение о необходимости перезапуска Apache «systemctl reload apache2», чтобы активировать новую конфигурацию; выполним запуск Apache.

```
systemctl start apache2
```

```
eva@word:~/src/openstreetmap-carto$ systemctl start apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'apache2.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
eva@word:~/src/openstreetmap-carto$
```

- и сразу же перезапустим Apache

```
systemctl reload apache2
```

```
eva@word:~/src/openstreetmap-carto$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
eva@word:~/src/openstreetmap-carto$
```

- сообщаем Apache о «renderd»
- открываем в редакторе nano файл 000-default.conf

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

```
eva@word:~/src/openstreetmap-carto$ sudo nano /etc/apache2/sites-available/000-default.conf
```

- вставляем следующие строки между строками «ServerAdmin» и «DocumentRoot»

```
LoadTileConfigFile /usr/local/etc/renderd.conf
ModTileRenderdSocketName /var/run/renderd/renderd.sock
# Timeout before giving up for a tile to be rendered
ModTileRequestTimeout 0
# Timeout before giving up for a tile to be rendered that is otherwise
missing
ModTileMissingRequestTimeout 30
```

```
  GNU nano 4.8                    /etc/apache2/sites-available/000-default.conf                    Modified
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        LoadTileConfigFile /usr/local/etc/renderd.conf
        ModTileRenderdSocketName /var/run/renderd/renderd.sock
        # Timeout before giving up for a tile to be rendered
        ModTileRequestTimeout 0
        # Timeout before giving up for a tile to be rendered that is otherwise missing
        ModTileMissingRequestTimeout 30
        DocumentRoot /var/www/html


^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line   M-E Redo
```

- нажимаем последовательно CTRL + O для сохранения файла
- нажимаем ENTER для подтверждения сохранения
- нажимаем последовательно CTRL + X для выхода из редактора

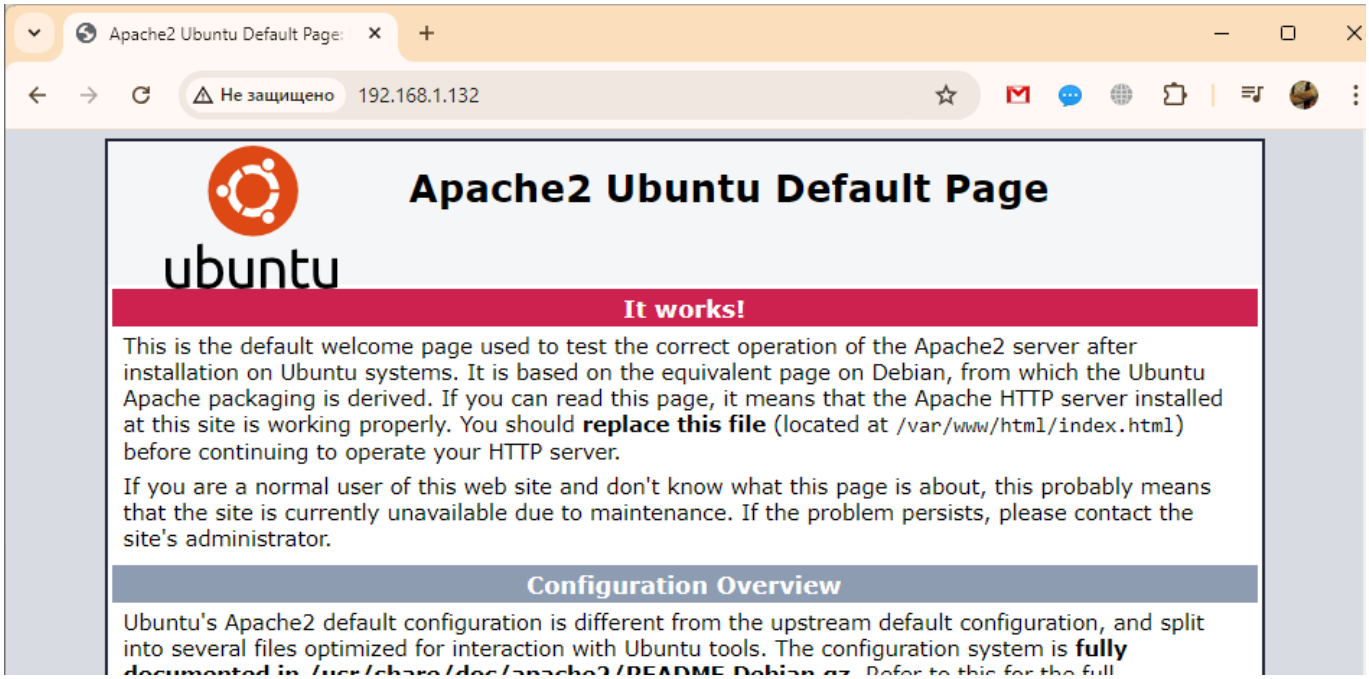
```
eva@word:~/src/openstreetmap-carto$ sudo nano /etc/apache2/sites-available/000-default.conf
eva@word:~/src/openstreetmap-carto$
```

- Снова перезагружаем apache:

```
service apache2 reload
```

```
eva@word:~/src/openstreetmap-carto$ service apache2 reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
eva@word:~/src/openstreetmap-carto$
```

- проверим наш Apache набрав в браузере адрес нашего виртуального сервера (в нашем случае: http://192.168.1.132/)



- **Дальше не продолжаем если не дождались загрузки данных в базу и не осуществили после этого Создание индексов**

Запуск рендера в первый раз

- Запустим renderd, чтобы попытаться отрендерить некоторые плитки. Сначала мы запустим его на переднем плане, чтобы видеть любые ошибки по мере их возникновения:

```
sudo -u 'eva' renderd -f -c /usr/local/etc/renderd.conf
```

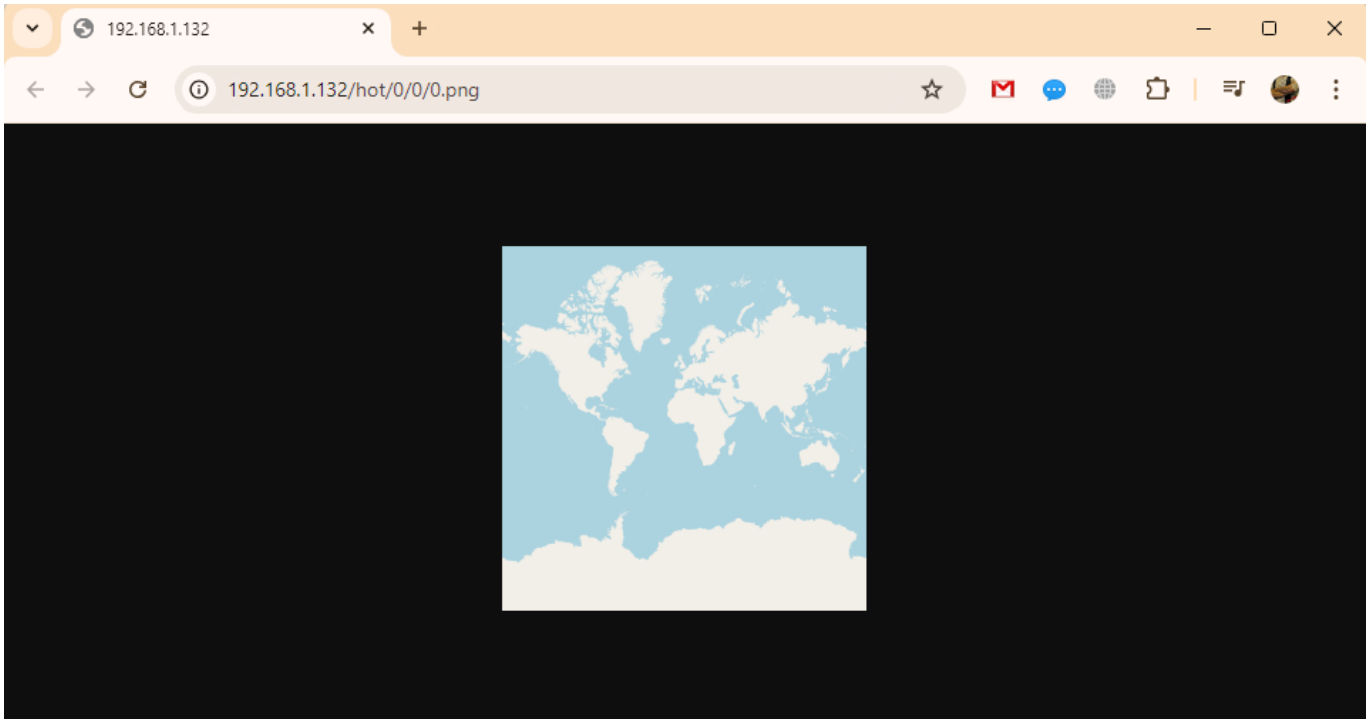
```
eva@word:~/src/openstreetmap-carto$ sudo -u 'eva' renderd -f -c /usr/local/etc/renderd.conf
renderd[50837]: Rendering daemon started
renderd[50837]: Initiating request_queue
renderd[50837]: Parsing section renderd
renderd[50837]: Parsing render section 0
renderd[50837]: Parsing section mapnik
renderd[50837]: Parsing section ajt
renderd[50837]: config renderd: unix socketname=/var/run/renderd/renderd.sock
renderd[50837]: config renderd: num_threads=4
renderd[50837]: config renderd: num_slaves=0
renderd[50837]: config renderd: tile_dir=/var/lib/mod_tile
renderd[50837]: config renderd: stats_file=/var/run/renderd/renderd.stats
renderd[50837]: config mapnik:  plugins_dir=/usr/lib/mapnik/3.0/input
renderd[50837]: config mapnik:  font_dir=/usr/share/fonts/truetype
renderd[50837]: config mapnik:  font_dir_recurse=1
renderd[50837]: config renderd(0): Active
```

- Эта команда отобразит рендер карты в окне консоли примерно следующее

```
renderd[50837]: Using web mercator projection settings
renderd[50837]: Using web mercator projection settings
renderd[50837]: Using web mercator projection settings
renderd[50837]: Using web mercator projection settings
renderd[50837]: DEBUG: Got incoming connection, fd 7, number 1
renderd[50837]: DEBUG: Got incoming request with protocol version 2
renderd[50837]: DEBUG: Got command RenderPrio fd(7) xml(ajt), z(0), x(0), y(0), mime(image/png), options()
renderd[50837]: DEBUG: START TILE ajt 0 0-0 0-0, new metatile
renderd[50837]: Rendering projected coordinates 0 0 0 -> -20037508.342800|-20037508.342800 20037508.342800|20037508.3
42800 to a 1 x 1 tile
renderd[50837]: DEBUG: DONE TILE ajt 0 0-0 0-0 in 0.267 seconds
debug: Creating and writing a metatile to /var/lib/mod_tile/ajt/0/0/0/0/0/0.meta

renderd[50837]: DEBUG: Sending render cmd(3 ajt 0/0/0) with protocol version 2 to fd 7
renderd[50837]: DEBUG: Connection 0, fd 7 closed, now 0 left
```

- Откройте веб-браузер по адресу: http://yourserveripaddress/hot/0/0/0.png (в нашем случае http://192.168.1.132/hot/0/0/0.png)



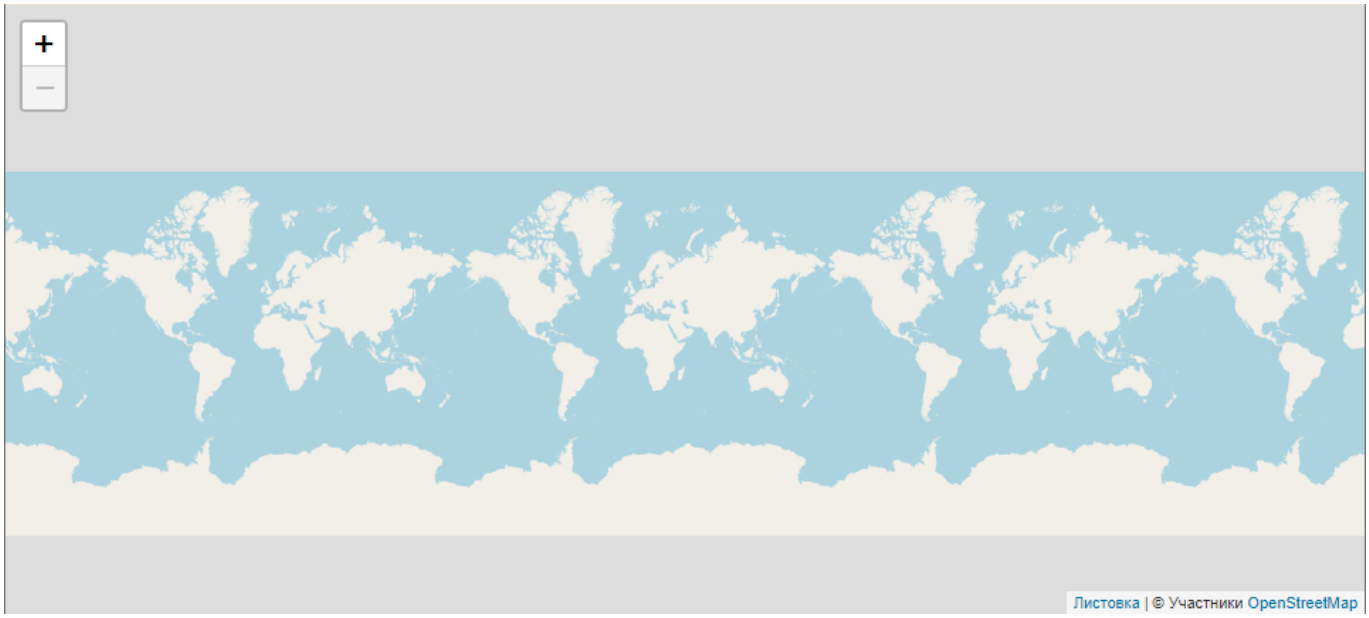
- Данное окно с плиткой говорит нам о том, что мы сделали все верно и наш сервер правильно работает. Загрузим файл просмотра плиток, разорхивируем и откроем его сначала в текстовом редакторе и изменим ip-адрес сервера на наш (в нашем примере 192.168.1.132).

```
<!DOCTYPE html>
<html style="height:100%;margin:0;padding:0;">
<title>Leaflet page with OSM render server selection</title>
<meta charset="utf-8">
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.3/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.3/dist/leaflet.js"></script>
<script src="https://unpkg.com/leaflet-hash@0.2.1/leaflet-hash.js"></script>
<style type="text/css">
.leaflet-tile-container { pointer-events: auto; }
</style>
</head>
<body style="height:100%;margin:0;padding:0;">
<div id="map" style="height:100%"></div>
<script>
//<![CDATA[
var map = L.map('map').setView([40.36629, 49.83335], 18);

L.tileLayer('http://192.168.1.132/hot/{z}/{x}/{y}.png', {
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
    maxZoom: 19,
    maxNativeZoom: 20
}).addTo(map);


var hash = L.hash(map)
//]]>
</script>
</body>
</html>
```

- Сохраним и откроем этот файл в веб-браузере



- Кнопками «+» и «-» приблизим район грузии, наблюдая в окне консоли как работает рендер



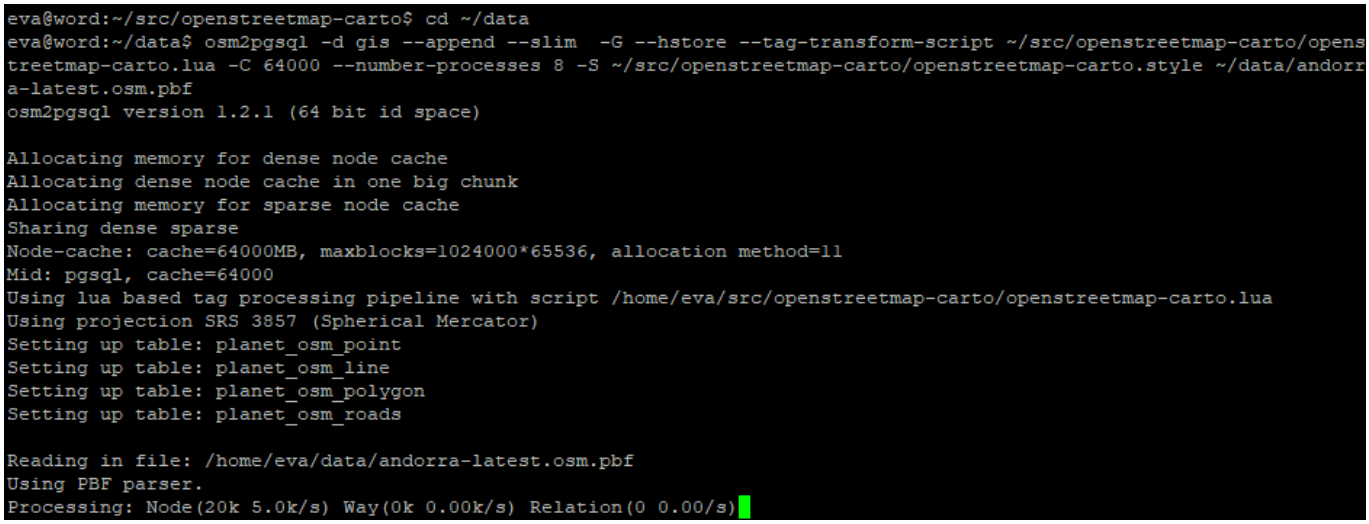
- Мы видим на карте отображение 3х государств (georgia, moldova и azerbaijan), которые мы объеденили в один файл в разделе Загрузка данных. Остановим ренден и загрузим небольшую карту «andorra», размером 2Mb в существующую карту используя команду (см. описание в разделе Догрузка карт в базу данных)
- Команда остановки работающего процесса **CTRL+C**, далее останавливаем рендер

```
systemctl stop renderd
```

```
eva@word:~/src/openstreetmap-carto$ systemctl stop renderd
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to stop 'renderd.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
Failed to stop renderd.service: Unit renderd.service not loaded.
eva@word:~/src/openstreetmap-carto$
```

- Переходим в каталог ~/data и запускаем загрузку файла карты «andorra»

```
cd ~/data
osm2pgsql -d gis --append --slim  -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 64000 --number-
processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style
~/data/andorra-latest.osm.pbf
```



- Процесс догрузки карт очень медленный, но это не важно когда запускаешь работу командой на сервере и он сам все делает, а пока за пару минут мы загрузили маленькую карту. Снова запустим рендер и проверим карту на наличие нового государства на карте.

```
sudo -u 'eva' renderd -f -c /usr/local/etc/renderd.conf
```

- В окне веб-браузера найдем наше новое государство. Т.к. оно очень маленькое, то искать его будем в строке браузера по координатам плиток «#7/39.428/1.725» (в меньших масштабах такие маленькие государства не отображаютя, а мы его взяли для примера догрузки)



Примечание:Иногда рендер идет с командой, т.е. без sudo -u 'eva'

```
renderd -f -c /usr/local/etc/renderd.conf
```

- Убидившись в работоспособности нашего сервера, научившись соединять файлы карт, загружать новые и дополнять в базе данных файлы карт, продолжим настройку сервера и подготовку к загрузке большого файла eurasia.osm.pbf (>40Gb), который к этому моменту уже должен подготовиться.

- Останавливаем работающий процесс **CTRL+C**, и далее останавливаем рендер

```
systemctl stop renderd
```

```
eva@word:~/data$ systemctl stop renderd
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to stop 'renderd.service'.
Authenticating as: eva
Password:
==== AUTHENTICATION COMPLETE ===
Failed to stop renderd.service: Unit renderd.service not loaded.
eva@word:~/data$
```

Запуск рендера в фоновом режиме

- Далее мы настроим «рендеринг» для работы в фоновом режиме (т.е. без нашего участи при запуске командой в консоли). Отредактируем файл **«~/src/mod_tile/debian/renderd.init»**, чтобы «RUNASUSER» был установлен для учетной записи без имени root, который мы использовали ранее, например «eva», и скопируем его в системный каталог. Откроем файл в редакторе nano

```
nano ~/src/mod_tile/debian/renderd.init
```

```
eva@word:~$ nano ~/src/mod_tile/debian/renderd.init
```

- Меняем в строке RUNASUSER renderaccount на ваше имя пользователя (в нашем случае это eva)

```
  GNU nano 4.8                    /home/eva/src/mod_tile/debian/renderd.init                Modified
#! /bin/sh
### BEGIN INIT INFO
# Provides:          renderd
# Required-Start:    $remote_fs
# Required-Stop:     $remote_fs
# Should-Start:      postgresql
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Mapnik rendering daemon
# Description:       Mapnik rendering daemon.
### END INIT INFO

# Do NOT "set -e"

# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin
DESC="Mapnik rendering daemon"
NAME=renderd
DAEMON=/usr/local/bin/$NAME
DAEMON_ARGS="-c /usr/local/etc/renderd.conf"
PIDSOCKDIR=/var/run/$NAME
PIDFILE=$PIDSOCKDIR/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
RUNASUSER=eva

# Exit if the package is not installed

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo
```

- нажимаем последовательно CTRL + O для сохранения файла
- нажимаем ENTER для подтверждения сохранения
- нажимаем последовательно CTRL + X для выхода из редактора

- Скопируем отредактированный файл это в системный каталог

```
sudo cp ~/src/mod_tile/debian/renderd.init /etc/init.d/renderd
```

```
eva@word:~$ sudo cp ~/src/mod_tile/debian/renderd.init /etc/init.d/renderd
[sudo] password for eva:
eva@word:~$
```

- Добавим право выполнения для владельца файла командой

```
sudo chmod u+x /etc/init.d/renderd
```

```
eva@word:~$ sudo chmod u+x /etc/init.d/renderd
eva@word:~$
```

- Скопирум сервис renderd в системный каталог

```
sudo cp ~/src/mod_tile/debian/renderd.service /lib/systemd/system/
```

```
eva@word:~$ sudo cp ~/src/mod_tile/debian/renderd.service /lib/systemd/system/
eva@word:~$
```

- Файл «render.service» является служебным файлом «systemd». Используемая здесь версия просто вызывает команды инициализации в старом стиле. Чтобы проверить, что команда запуска работает:

```
sudo /etc/init.d/renderd start
```

- (ответит «Starting renderd (через systemctl): renderd.service».)

```
eva@word:~$ sudo /etc/init.d/renderd start
Starting renderd (via systemctl): renderd.service.
eva@word:~$
```

- Чтобы это запускалось автоматически каждый раз:

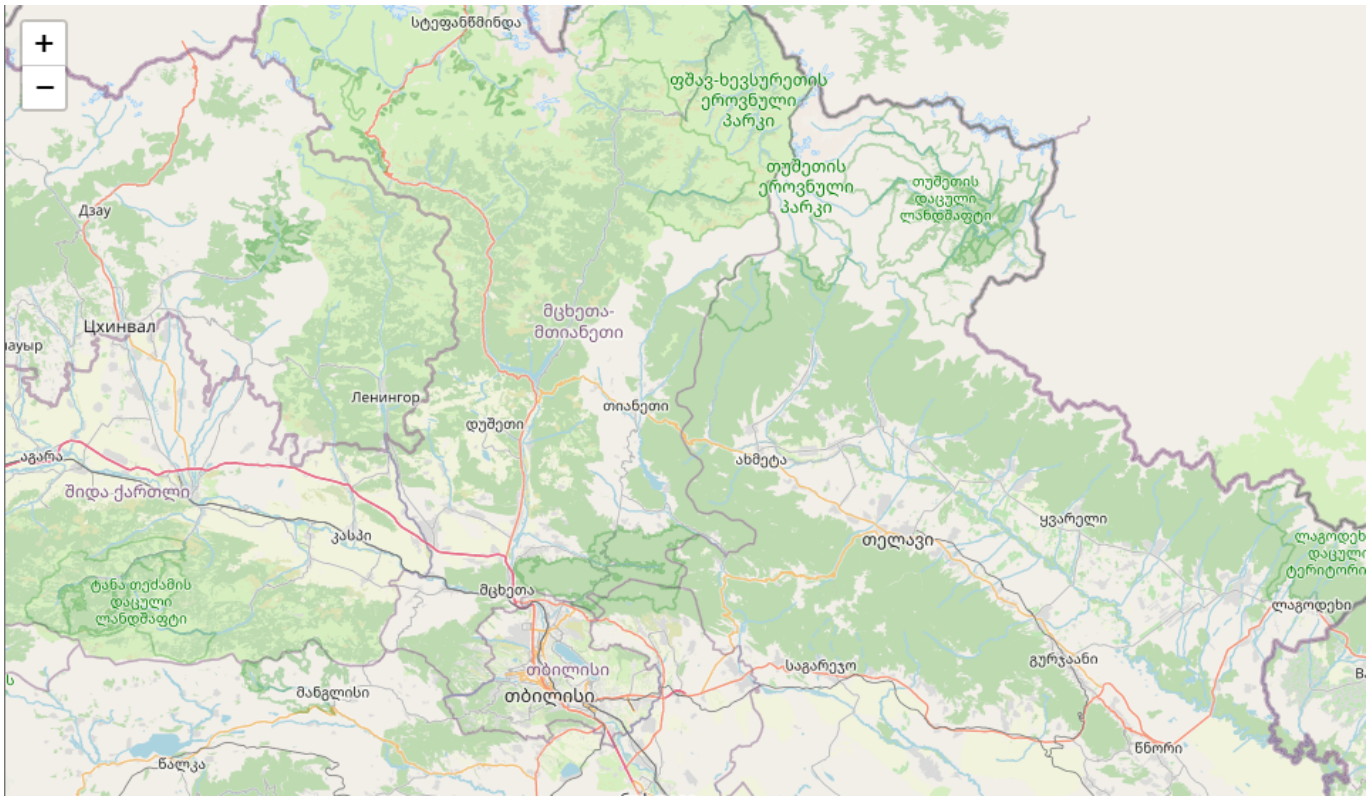
```
sudo systemctl enable renderd
```

```
eva@word:~$ sudo systemctl enable renderd
Synchronizing state of renderd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable renderd
Created symlink /etc/systemd/system/multi-user.target.wants/renderd.service → /lib/systemd/system/renderd.service.
eva@word:~$
```

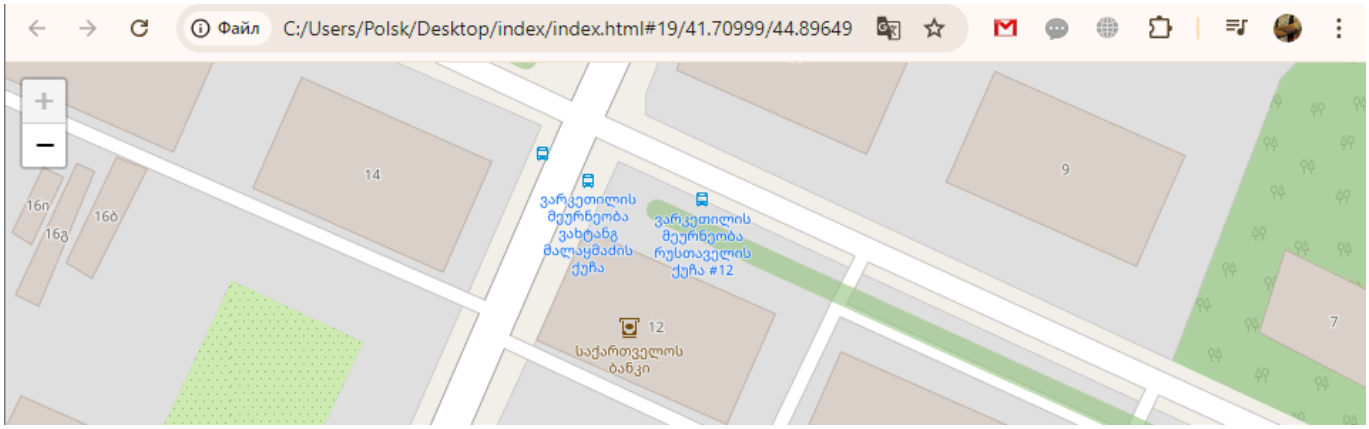
- В окне веб-браузера с нашей картой убедимся (приближаем/удаляем и двигаем нашу карту), что рендер работает автоматически, т.е. без нашего участия в консоли.



- Размытия и серые зоны говорят о незагруженности карты при рендере, последующие движения в этой области уже будет четкими.



- Загрузим максимально приближенный масштаб карты. Если у вам это получилось, значит сервер настроен правильно.



- Проверяем завершение слияние нашего файла **eurasia.osm.pbf** (во втором окне консоли должен быть завершенный процесс и ожидание новых команд)

```
eva@word:~/data$ osmconvert europe-latest.osm.pbf --out-o5m | osmconvert - asia-latest.osm.pbf -o=eurasia.osm.pbf
eva@word:~/data$
```

- Проверим размер файла eurasia.osm.pbf после слияния

```
cd ~/data
ls -n
```

```
eva@word:~/data$ ls -n
total 85989744
-rw-rw-r-- 1 1000 1000     2679565 Jul 27 01:09 andorra-latest.osm.pbf
-rw-rw-r-- 1 1000 1000 14005243451 Jul 26 23:04 asia-latest.osm.pbf
-rw-rw-r-- 1 1000 1000    39275141 Jul 27 02:17 azerbaijan-latest.osm.pbf
-rw------- 1 1000 1000 42544864248 Aug 18 19:23 eurasia.osm.pbf
-rw-rw-r-- 1 1000 1000 30961314293 Jul 26 23:04 europe-latest.osm.pbf
-rw-rw-r-- 1 1000 1000    84553767 Jul 27 02:01 georgia-latest.osm.pbf
-rw-rw-r-- 1 1000 1000    73462688 Jul 27 01:43 moldova-latest.osm.pbf
-rw------- 1 1000 1000   152128562 Aug 18 16:49 new-latest.osm.pbf
-rw------- 1 1000 1000   189941383 Aug 18 16:54 new.osm.pbf
eva@word:~/data$
```

- Удалим лишние файлы для освобождения места на сервере (оно нам понадобиться при загрузке eurasia.osm.pbf). На момент написания этой статьи вся карта мира (76Gb) не помещается на SSD диске размером 1,8Tb, учитывайте это, чтобы не получить сюрприз в виде нехватки места на диске.

```
sudo rm andorra-latest.osm.pbf
sudo rm asia-latest.osm.pbf
sudo rm azerbaijan-latest.osm.pbf
sudo rm europe-latest.osm.pbf
sudo rm georgia-latest.osm.pbf
sudo rm moldova-latest.osm.pbf
sudo rm new-latest.osm.pbf
sudo rm new.osm.pbf
```

```
eva@word:~/data$ sudo rm asia-latest.osm.pbf
eva@word:~/data$ sudo rm azerbaijan-latest.osm.pbf
eva@word:~/data$ sudo rm europe-latest.osm.pbf
eva@word:~/data$ sudo rm georgia-latest.osm.pbf
eva@word:~/data$ sudo rm moldova-latest.osm.pbf
eva@word:~/data$ sudo rm new-latest.osm.pbf
eva@word:~/data$ sudo rm new.osm.pbf
eva@word:~/data$
```

- Перепроверим, что в каталоге остался только файл eurasia.osm.pbf

```
ls -n
```

```
eva@word:~/data$ ls -n
total 41547728
-rw------- 1 1000 1000 42544864248 Aug 18 19:23 eurasia.osm.pbf
eva@word:~/data$
```

- Следующая команда с параметром **–create** сотрет все данные в нашей базе данных и загрузит файл карты eurasia. Процесс достаточно длительный и займет очень много времени (примерно >15-17 часов, в зависимости от комплектации сервера)

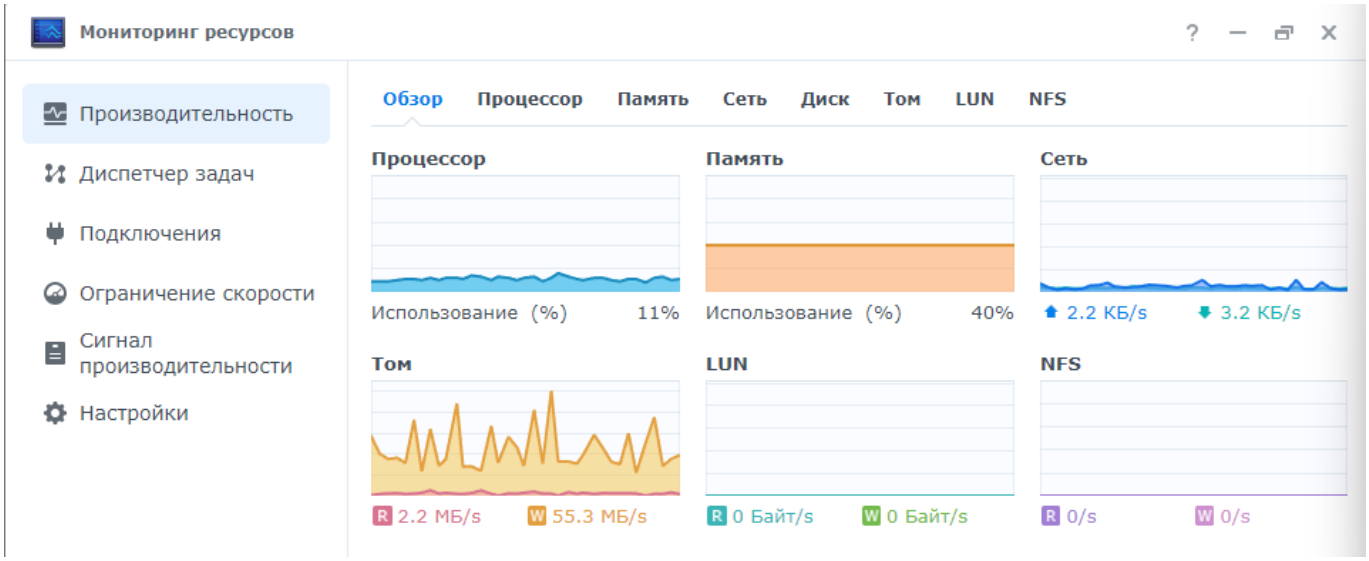
```
cd ~/data
osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script
~/src/openstreetmap-carto/openstreetmap-carto.lua -C 64000 --number-
processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style
~/data/eurasia.osm.pbf
```

```
eva@word:~/data$ osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script ~/src/openstreetmap-carto/opens
treetmap-carto.lua -C 64000 --number-processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/eurasi
a.osm.pbf
osm2pgsql version 1.2.1 (64 bit id space)

Allocating memory for dense node cache
Allocating dense node cache in one big chunk
Allocating memory for sparse node cache
Sharing dense sparse
Node-cache: cache=64000MB, maxblocks=1024000*65536, allocation method=11
Mid: pgsql, cache=64000
Setting up table: planet_osm_nodes
Setting up table: planet_osm_ways
Setting up table: planet_osm_rels
Using lua based tag processing pipeline with script /home/eva/src/openstreetmap-carto/openstreetmap-carto.lua
Using projection SRS 3857 (Spherical Mercator)
Setting up table: planet_osm_point
Setting up table: planet_osm_line
Setting up table: planet_osm_polygon
Setting up table: planet_osm_roads

Reading in file: /home/eva/data/eurasia.osm.pbf
Using PBF parser.
Processing: Node(13560k 347.7k/s) Way(0k 0.00k/s) Relation(0 0.00/s)
```

- По окончанию загрузки, мы получим полность рабочий тайловый сервер с картой Евразии. Останется только подключить сервер к сайту и провести предварительный рендер карты, чтобы посетители не ждали загрузку плиток.
- А пока (процесс Node загрузка >6млнk) взглянем на производительность нашего сервера



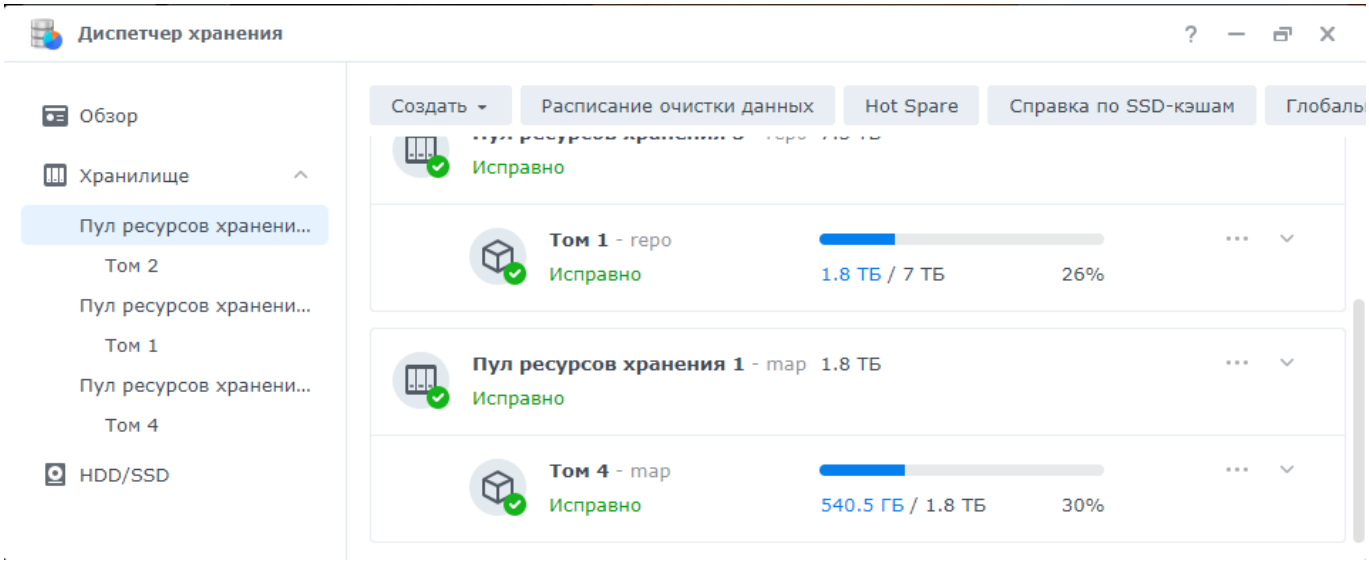
- После 7 часов преобразования данных (parser processing Node: 5287016k 279.3k/s)

```
eva@word:~/data$ sudo rm new-latest.osm.pbf
eva@word:~/data$ sudo rm new.osm.pbf
eva@word:~/data$ cd ~/data
eva@word:~/data$ ls -n
total 41547728
-rw------- 1 1000 1000 42544864248 Aug 18 19:23 eurasia.osm.pbf
eva@word:~/data$ osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script ~/src/openstreetmap-carto/opens
treetmap-carto.lua -C 64000 --number-processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/eurasi
a.osm.pbf
osm2pgsql version 1.2.1 (64 bit id space)

Allocating memory for dense node cache
Allocating dense node cache in one big chunk
Allocating memory for sparse node cache
Sharing dense sparse
Node-cache: cache=64000MB, maxblocks=1024000*65536, allocation method=11
Mid: pgsql, cache=64000
Setting up table: planet_osm_nodes
Setting up table: planet_osm_ways
Setting up table: planet_osm_rels
Using lua based tag processing pipeline with script /home/eva/src/openstreetmap-carto/openstreetmap-carto.lua
Using projection SRS 3857 (Spherical Mercator)
Setting up table: planet_osm_point
Setting up table: planet_osm_line
Setting up table: planet_osm_polygon
Setting up table: planet_osm_roads

Reading in file: /home/eva/data/eurasia.osm.pbf
Using PBF parser.
Processing: Node(5287016k 279.3k/s) Way(102971k 13.51k/s) Relation(0 0.00/s)
```

- Получаем загрузку диска в 30% (из каторых 20Gb- это собственно настроенная ОС Ubuntu Server + 41Gb наш файл eurasia.osm.pbf и 490Gb - преобразованные данные нашей карты)



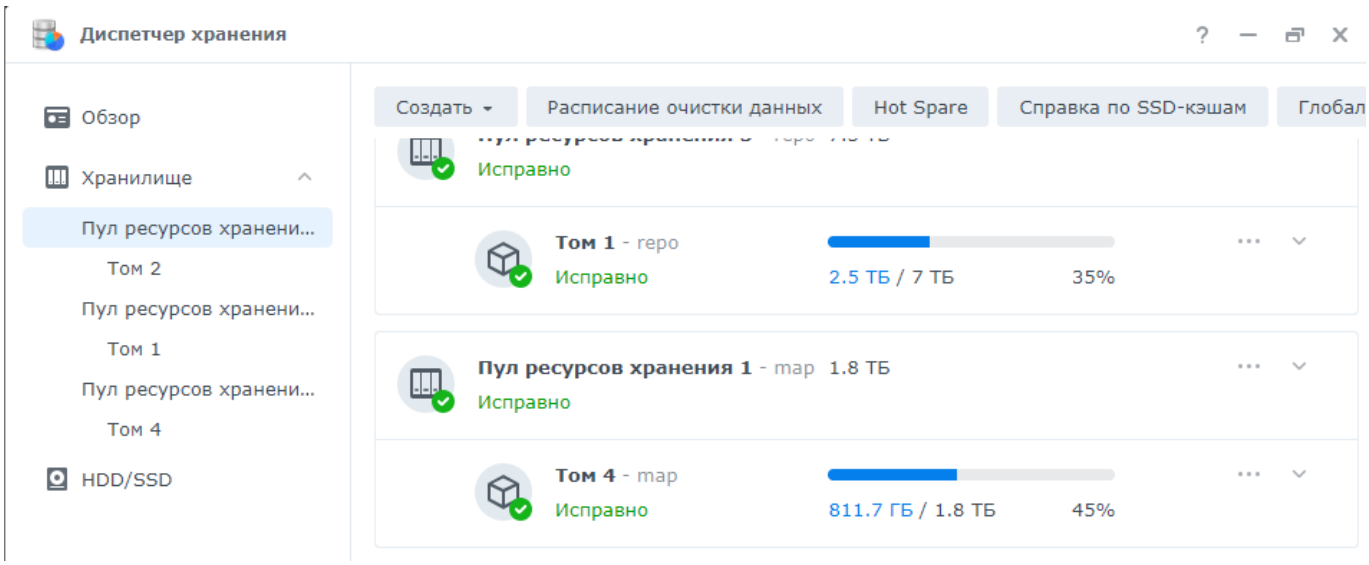
- Прошли почти сутки, мы видим данные, которые уже загрузились, в кансоле

```
eva@word:~/data$ sudo rm new-latest.osm.pbf
eva@word:~/data$ sudo rm new.osm.pbf
eva@word:~/data$ cd ~/data
eva@word:~/data$ ls -n
total 41547728
-rw------- 1 1000 1000 42544864248 Aug 18 19:23 eurasia.osm.pbf
eva@word:~/data$ osm2pgsql -d gis --create --slim  -G --hstore --tag-transform-script ~/src/openstreetmap-carto/opens
treetmap-carto.lua -C 64000 --number-processes 8 -S ~/src/openstreetmap-carto/openstreetmap-carto.style ~/data/eurasi
a.osm.pbf
osm2pgsql version 1.2.1 (64 bit id space)

Allocating memory for dense node cache
Allocating dense node cache in one big chunk
Allocating memory for sparse node cache
Sharing dense sparse
Node-cache: cache=64000MB, maxblocks=1024000*65536, allocation method=11
Mid: pgsql, cache=64000
Setting up table: planet_osm_nodes
Setting up table: planet_osm_ways
Setting up table: planet_osm_rels
Using lua based tag processing pipeline with script /home/eva/src/openstreetmap-carto/openstreetmap-carto.lua
Using projection SRS 3857 (Spherical Mercator)
Setting up table: planet_osm_point
Setting up table: planet_osm_line
Setting up table: planet_osm_polygon
Setting up table: planet_osm_roads

Reading in file: /home/eva/data/eurasia.osm.pbf
Using PBF parser.
Processing: Node(5287016k 279.3k/s) Way(646595k 17.70k/s) Relation(4190750 515.78/s)
```

- При этом заполнение SSD диска уже составило 811Gb при загрузке карты Евразии размером всего 40Gb

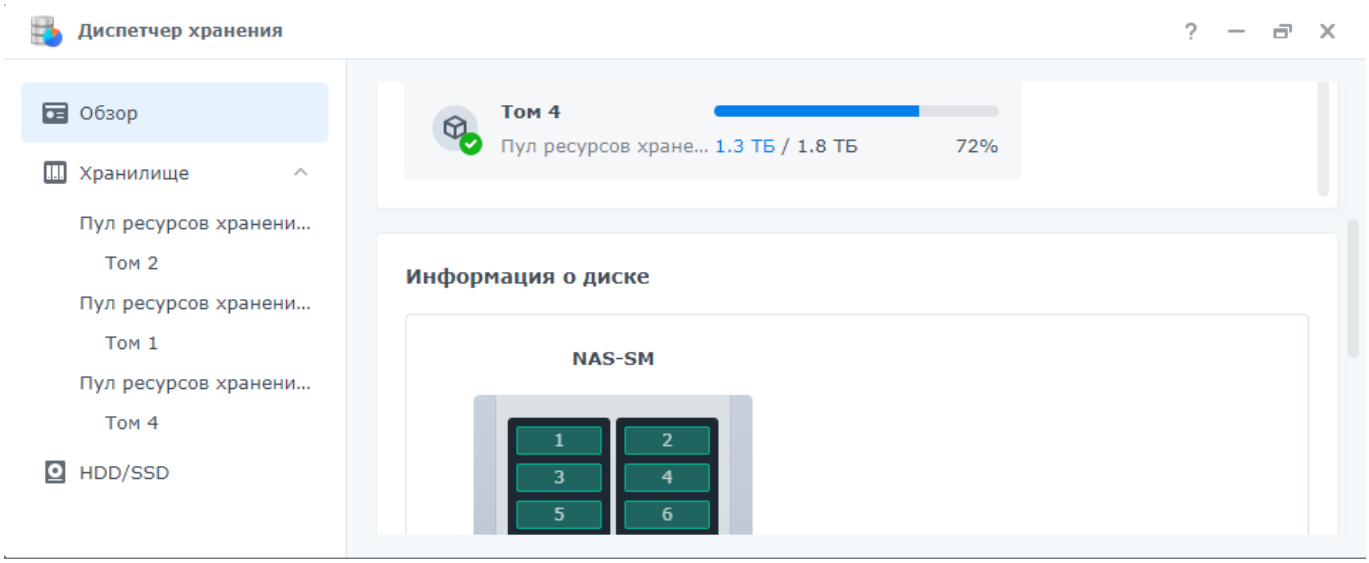


- Прошли почти 2 сутки (48 часов), мы видим данные в кансоле об окончании загрузки

```
Creating geometry index on planet_osm_roads
Creating osm_id index on planet_osm_roads
Creating indexes on planet_osm_roads finished
All indexes on planet_osm_roads created in 3271s
Completed planet_osm_roads
Copying planet_osm_point to cluster by geometry finished
Creating geometry index on planet_osm_point
Stopped table: planet_osm_rels in 5349s
Creating osm_id index on planet_osm_point
Creating indexes on planet_osm_point finished
All indexes on planet_osm_point created in 9090s
Completed planet_osm_point
Copying planet_osm_line to cluster by geometry finished
Creating geometry index on planet_osm_line
Creating osm_id index on planet_osm_line
Creating indexes on planet_osm_line finished
All indexes on planet_osm_line created in 18251s
Completed planet_osm_line
Copying planet_osm_polygon to cluster by geometry finished
Creating geometry index on planet_osm_polygon
Creating osm_id index on planet_osm_polygon
Creating indexes on planet_osm_polygon finished
All indexes on planet_osm_polygon created in 33562s
Completed planet_osm_polygon
Stopped table: planet_osm_ways in 79627s

Osm2pgsql took 149346s overall
node cache: stored: 5148655449(97.38%), storage efficiency: 61.38% (dense blocks: 623556, sparse nodes: 1640221079),
hit rate: 97.47%
eva@word:~/data$
```

- И заполнение SSD диска на 1,3Tb при загрузке карты Евразии размером всего 40Gb



- Вернемся в раздел создание индексов и запустим повторно создание индексов.

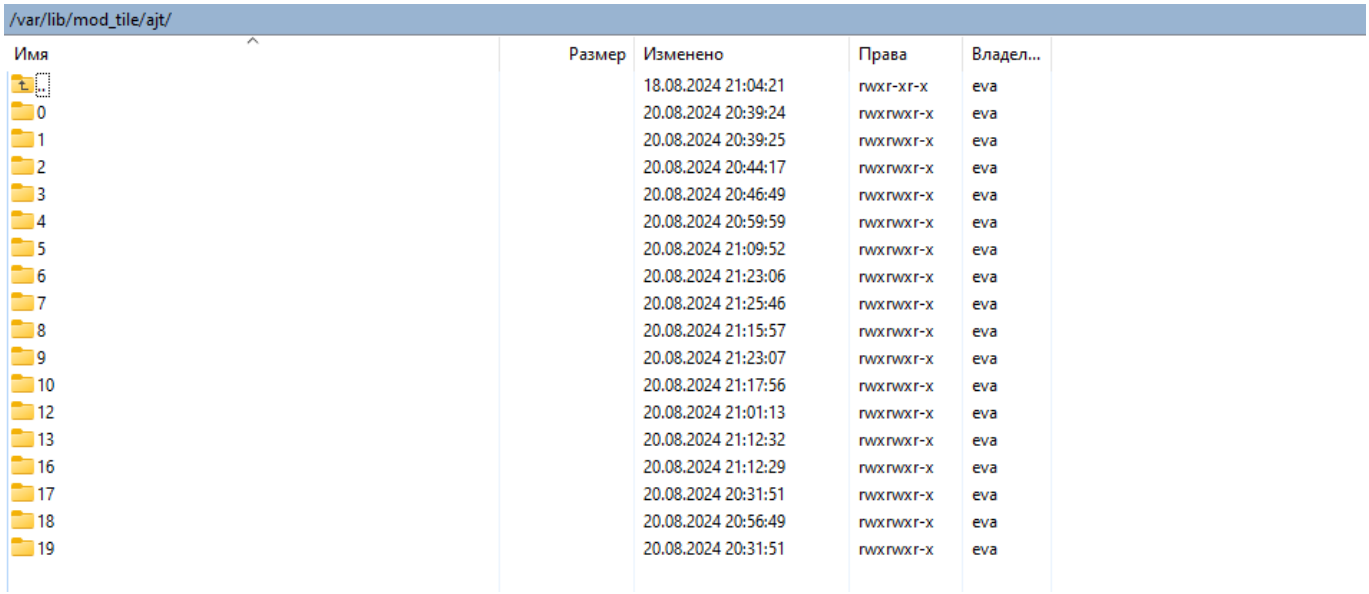
```
eva@word:~/src/openstreetmap-carto$ scripts/get-external-data.py
INFO:root:Starting load of external data into database
INFO:root:Checking table simplified_water_polygons
INFO:root:  Table simplified_water_polygons did not require updating
INFO:root:Checking table water_polygons
INFO:root:  Table water_polygons did not require updating
INFO:root:Checking table icesheet_polygons
INFO:root:  Table icesheet_polygons did not require updating
INFO:root:Checking table icesheet_outlines
INFO:root:  Table icesheet_outlines did not require updating
INFO:root:Checking table ne_110m_admin_0_boundary_lines_land
INFO:root:  Table ne_110m_admin_0_boundary_lines_land did not require updating
eva@word:~/src/openstreetmap-carto$
```

- При приближении мы видим рендер участков карты которые ранее не были записаны в кэш, а те что уже записаны закрыты серыми плитками, т.к. при первых картах эти участки не были в базе данных.

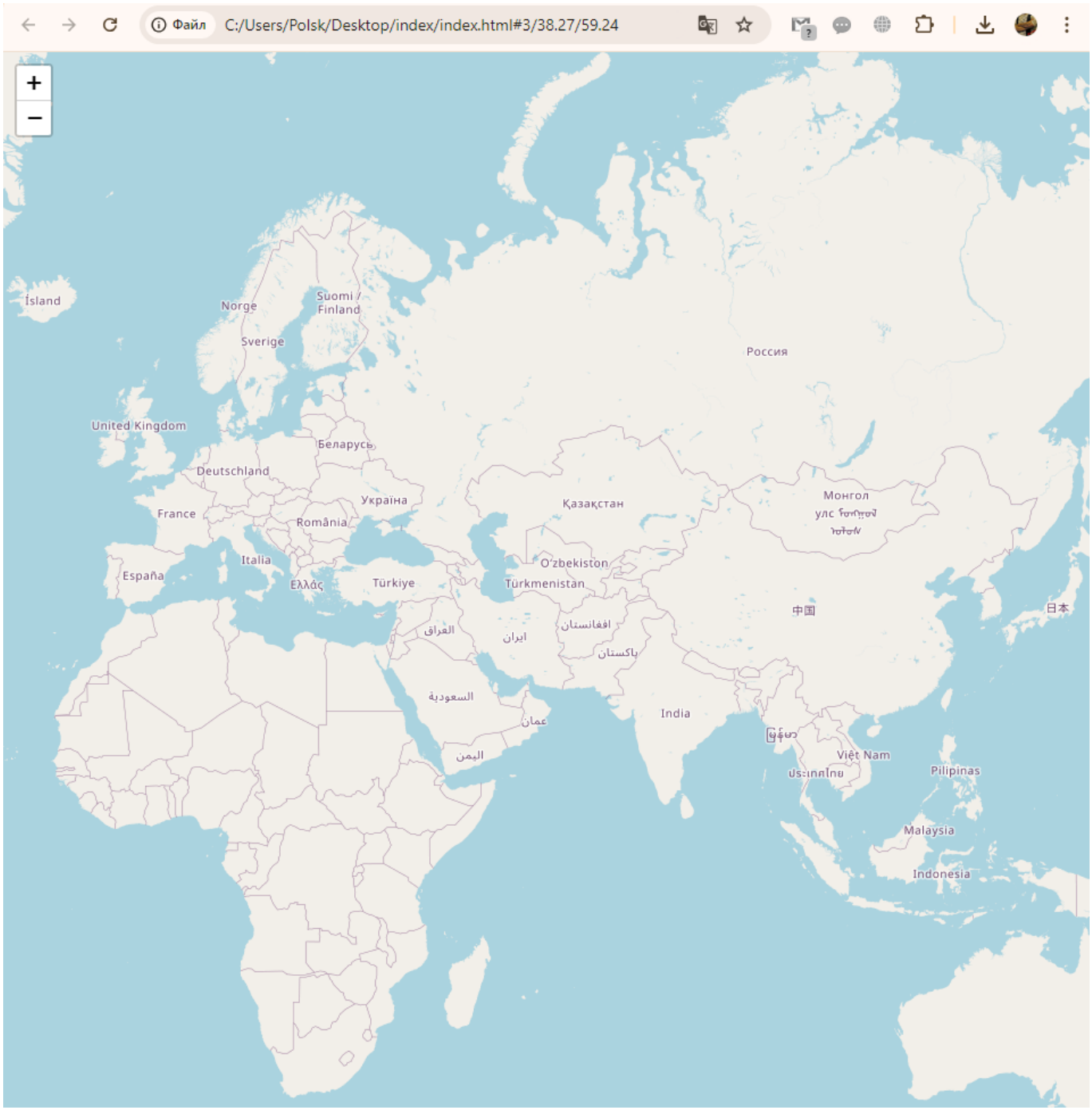


- Заходим в наш сервер в каталог /var/lib/mod_tile/ajt и удаляем все каталоги с кешем

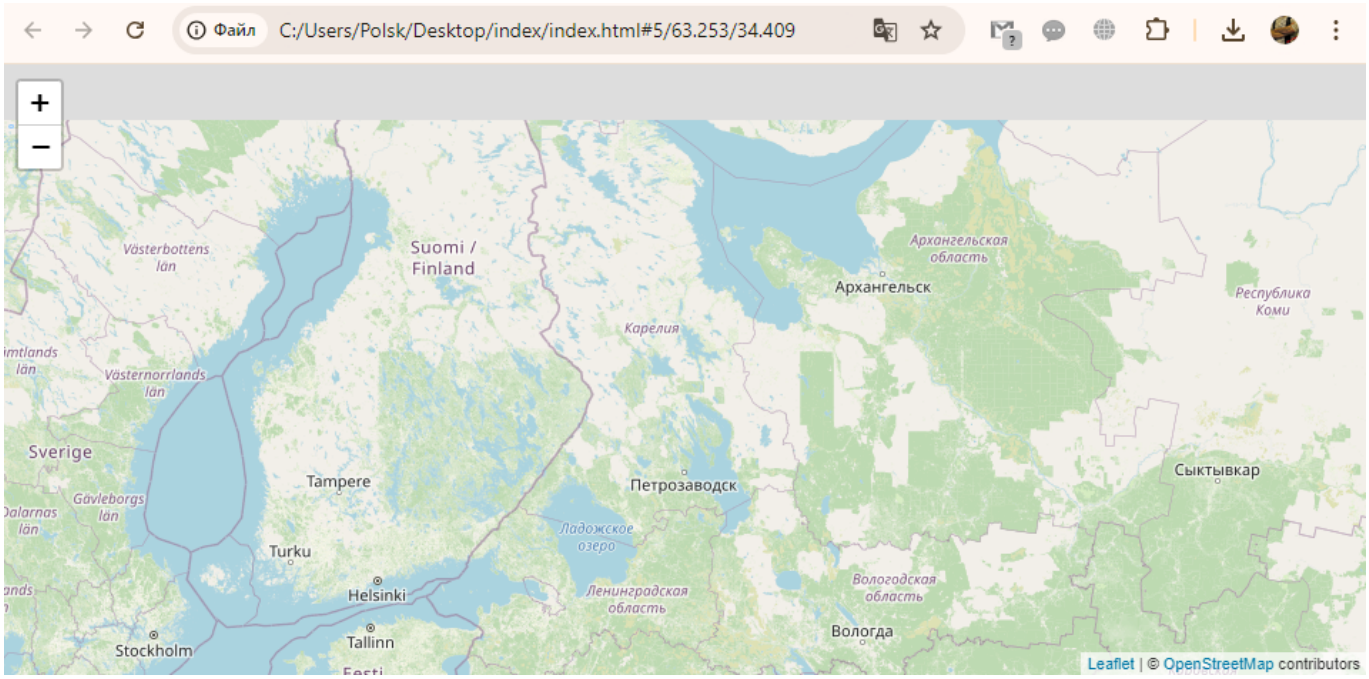
прошлого рендера



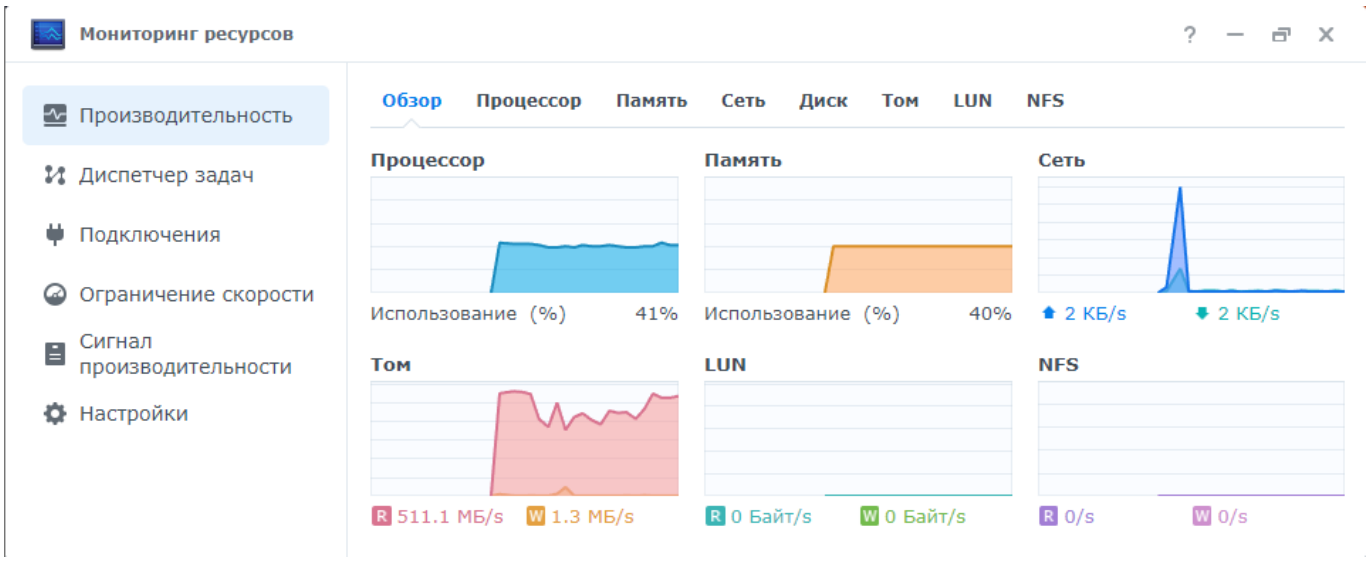
- Возвращаемся в наш веб-браузер и обновляем страницу. Перед нами карта с названиями государств Европы и Азии, при этом видим, что Африка и Австралия просто белые и без названий, т.к. мы их не грузили в базу данных. Остается провести предварительный рендер, чтобы пользователи не ждали загрузки.



- Процесс рендера достаточно медленный, но мы можем наблюдать преобразование новых регионов, а те, что еще не попали в кэш имеют плитки серого цвета



- При этом сервер нагружен на 40% по мониторингу процессора, и почти на максимум работает ssd, учитывая, что установлен AMD EPYC™ в 32 ядра/64 потока и 256gb оперативной памяти.



- Следующая команда запустит предварительный рендер всей карты с 0 до 10 уровня в 16 потоков, чтобы пользователи не ждали загрузки тайлов, а использовали уже готовые области карт

```
sudo -u eva render_list -a -n 16  --map=ajt --tile-dir=/var/lib/mod_tile/ajt
--min-zoom=0 --max-zoom=10 --force
```

- Ниже приведены параметры **render_list**

```
Использование: render_list [ОПЦИЯ]...
  -a, --all визуализировать все плитки в заданном диапазоне масштабирования вместо
чтения из STDIN
  -f, --force отображать плитки, даже если они кажутся текущими
```

-m, --map=MAP отображает тайлы на этой карте (по умолчанию «по умолчанию»)
-l, --max-load=ЗАГРУЗИТЬ спящий режим, если нагрузка настолько высока (по умолчанию 16)
-s, --socket=SOCKET имя доменного сокета unix для связи с renderd
-n, --num-threads=N количество параллельных потоков запросов (по умолчанию 1)
-t, --tile-dir Каталог тайлового кэша (по умолчанию «/var/lib/mod_tile»)
-z, --min-zoom=Ввод фильтра ZOOM для рендеринга только тех фрагментов, которые больше или равны этому уровню масштабирования (по умолчанию – 0)
-Z, --max-zoom=Ввод фильтра ZOOM для рендеринга только плиток, меньших или равных этому уровню масштабирования (по умолчанию — 20)
Если вы используете --all, вы можете ограничить диапазон плиток, добавив следующие параметры:
-x, --min-x=X минимальная координата X плитки
-X, --max-x=X максимальная координата X тайла
-y, --min-y=Y минимальная координата Y тайла
-Y, --max-y=Y максимальная координата тайла Y
Без --all отправить список тайлов для рендеринга из STDIN в формате:
X Y Я
например
0 0 1
0 1 1
1 0 1
1 1 1
Вышеупомянутое приведет к отображению всех 4 плиток при масштабировании 1.

```
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Aug 28 00:09:36 2024 from 192.168.1.103
eva@map-ea:~$ sudo -u eva render_list -a -n 16  --map=ajt --tile-dir=/var/lib/mo
d_tile/ajt --min-zoom=0 --max-zoom=18 --force
debug: init_storage_backend: initialising file storage backend at: /var/lib/mod_
tile/ajt
Rendering client
Starting 16 rendering threads
Rendering all tiles from zoom 0 to zoom 18
Rendering all tiles for zoom 0 from (0, 0) to (0, 0)
Rendering all tiles for zoom 1 from (0, 0) to (1, 1)
Rendering all tiles for zoom 2 from (0, 0) to (3, 3)
Rendering all tiles for zoom 3 from (0, 0) to (7, 7)
Rendering all tiles for zoom 4 from (0, 0) to (15, 15)
Rendering all tiles for zoom 5 from (0, 0) to (31, 31)
Rendering all tiles for zoom 6 from (0, 0) to (63, 63)
Rendering all tiles for zoom 7 from (0, 0) to (127, 127)
Rendering all tiles for zoom 8 from (0, 0) to (255, 255)
Rendering all tiles for zoom 9 from (0, 0) to (511, 511)
Rendering all tiles for zoom 10 from (0, 0) to (1023, 1023)
Rendering all tiles for zoom 11 from (0, 0) to (2047, 2047)
Rendering all tiles for zoom 12 from (0, 0) to (4095, 4095)
Rendering all tiles for zoom 13 from (0, 0) to (8191, 8191)
Rendering all tiles for zoom 14 from (0, 0) to (16383, 16383)
```

Дополнения и Файлы

- ОС Ubuntu Server 20.04 LTS
- Файл для просмотра плиток и проверки render
- Настройка сервера PostgreSQL

- Настройка PostgreSQL под Linux

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:linux_server:tile_map_server_worl_20.04.4

Last update: **2025/02/01 09:47**