

Tile map server

Введение

На этой странице показано, как можно использовать OpenStreetMap Carto для реализации тайлового сервера с использованием того же программного обеспечения, что и OpenStreetMap. Он включает в себя пошаговые инструкции по установке Tile Server на основе Ubuntu и ограничен описанием некоторых передовых методов, учитывая, что основной целью этого сайта является предоставление руководств по настройке среды разработки OpenStreetMap Carto и предложение рекомендации по редактированию стиля.

OSM Tile Server — это веб-сервер, специализирующийся на доставке растровых карт, представляющих их как статические тайлы и способный выполнять рендеринг в реальном времени или предоставлять кэшированные изображения. Веб-программное обеспечение, принятое OpenStreetMap, представляет собой HTTP-сервер Apache вместе со специальным подключаемым модулем с именем `mod_tile` и связанным с ним внутренним стеком, способным генерировать плитки во время выполнения; программы и библиотеки объединены в цепочку для создания сервера тайлов.

Как это часто бывает с OpenStreetMap, существует множество способов достижения цели, и почти все компоненты имеют альтернативы, которые имеют различные преимущества и недостатки. В этом руководстве описывается стандартный процесс установки OSM Tile Server, используемого на OpenStreetMap.org.

Он состоит из следующих основных компонентов:

- Mapnik
- Apache
- Mod_tile
- renderd
- osm2pgsql
- PostgreSQL/PostGIS database, to be installed locally (suggested) or remotely (might be slow, depending on the network).
- carto
- openstreetmap-carto

Все упомянутое программное обеспечение является открытым исходным кодом.

Для тайлового сервера требуется база данных PostGIS, в которой хранятся геопространственные объекты, заполненные инструментом `osm2pgsql` из данных OSM. Кроме того, необходим каталог файловой системы, включающий файл `OSM.xml`, символы карты (проверьте подкаталог `openstreetmap-carto/symbols`) и шейп-файлы (проверьте подкаталог `openstreetmap-carto/data`). `OSM.xml` предварительно создается инструментом `carto` из стиля `openstreetmap-carto` (`project.mml` и все связанные файлы `CartoCSS`, включенные в `openstreetmap-carto`).

Когда веб-сервер Apache получает запрос от браузера, он вызывает подключаемый модуль `mod_tile`, который, в свою очередь, проверяет, была ли плитка уже создана (из предыдущего рендеринга) и кэширована, чтобы она была готова к использованию; в этом случае `mod_tile` немедленно отправляет плитку обратно на веб-сервер. И наоборот, если запрос необходимо отобразить, то он ставится в очередь на серверную часть рендеринга, которая отвечает за вызов Mapnik для выполнения фактического рендеринга; `renderd` — это процесс-демон,

включенный в исходники `mod_tile` и связанный с `mod_tile` через очереди UNIX.

визуализированный является стандартным бэкэндом, используемым в настоящее время www.openstreetmap.org, даже если некоторые реализации OSM используют Tiler ; Mapnik извлекает данные из базы данных PostGIS в соответствии с информацией о стиле `openstreetmap-carto` и динамически отображает тайл. `renderd` возвращает созданный тайл на веб-сервер и, в свою очередь, в браузер.

Демон `renderd` реализует механизм очередей с несколькими уровнями приоритета, чтобы обеспечить актуальность просмотра с учетом доступных ресурсов рендеринга. Наивысший приоритет — для рендеринга «на лету» тайлов, еще не находящихся в кэше тайлов, два уровня приоритета для повторного рендеринга устаревших тайлов «на лету» и две очереди фонового пакетного рендеринга. Чтобы избежать проблем с каталогами, которые становятся слишком большими, и чтобы избежать слишком большого количества крошечных файлов, `Mod_tile / renderd` хранит визуализированные тайлы в «метатайлах», в специальной хешированной структуре каталогов.

Даже если тайловый сервер динамически генерирует тайлы во время выполнения, они также могут быть предварительно визуализированы для просмотра в автономном режиме с помощью специального инструмента с именем `render_list`, который обычно используется для предварительного рендеринга тайлов с низким уровнем масштабирования и требует значительного времени для выполнения процесса (десятки часов в случае, если вся планета предварительно визуализируется); эта утилита включена в `mod_tile`, а также в другой инструмент с именем `render_expired`, который предоставляет методы для разрешения истечения срока действия тайлов карты. Более подробное описание `render_list` и `render_expired` можно найти на их справочных страницах.

Справочную информацию о методе истечения срока действия тайлов можно найти в механизме истечения срока действия тайлов.

Общая настройка для Ubuntu

Обновите Ubuntu

Убедитесь, что ваша система Ubuntu полностью обновлена:

```
lsb_release -a
```

команда возвращает версию Ubuntu.

```
vladpolskiy@linux:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 22.04 LTS
Release:       22.04
Codename:      jammy
```

Чтобы обновить систему:

```
sudo apt-get update
```

команда загружает списки пакетов из репозитория и «обновляет» их, чтобы получить информацию о новейших версиях пакетов и их зависимостях. Это будет сделано для всех репозитория и PPA. Из <http://linux.die.net/man/8/apt-get>

```
vladpolskiy@linux:~$ sudo apt-get update
[sudo] password for vladpolskiy:
Hit:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [479
kB]
Get:6 http://ru.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[203 kB]
Fetched 1,007 kB in 1s (730 kB/s)
Reading package lists... Done
vladpolskiy@linux:~$
```

```
sudo apt list --upgradable
```

команда выведит списки пакетов готовых к обновлению

```
vladpolskiy@linux:~$ apt list --upgradable
Listing... Done
apparmor/jammy-updates 3.0.4-2ubuntu2.1 amd64 [upgradable from: 3.0.4-2ubuntu2]
apt-utils/jammy-updates 2.4.6 amd64 [upgradable from: 2.4.5]
apt/jammy-updates 2.4.6 amd64 [upgradable from: 2.4.5]
base-files/jammy-updates 12ubuntu4.2 amd64 [upgradable from: 12ubuntu4]
firmware-sof-signed/jammy-updates 2.0-1ubuntu3 all [upgradable from: 2.0-1ubuntu
2]
isc-dhcp-client/jammy-updates 4.4.1-2.3ubuntu2.1 amd64 [upgradable from: 4.4.1-2
.3ubuntu2]
isc-dhcp-common/jammy-updates 4.4.1-2.3ubuntu2.1 amd64 [upgradable from: 4.4.1-2
.3ubuntu2]
libapparmor1/jammy-updates 3.0.4-2ubuntu2.1 amd64 [upgradable from: 3.0.4-2ubunt
u2]
libapt-pkg6.0/jammy-updates 2.4.6 amd64 [upgradable from: 2.4.5]
```

```
sudo apt-get -y upgrade
```

команда обновляет все устаревшие пакеты и применить исправления безопасности с выводом подсказок « Да » или « Нет », спрашивая, нужно ли устанавливать зависимые пакеты или нет. Если вы устанавливаете большой пакет или пакет с большим количеством зависимостей, или устанавливаете несколько пакетов, ответы на эти запросы не позволят вам переключиться на выполнение какой-либо работы.

```
vladpolskiy@linux:~$ sudo apt-get -y upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apparmor apt apt-utils base-files firmware-sof-signed isc-dhcp-client
  isc-dhcp-common libapparmor1 libapt-pkg6.0 libc-bin libc6 libgstreamer1.0-0
  libjson-c5 libldap-2.5-0 libldap-common libnetplan0 libnss-systemd
  libpam-systemd libsystemd0 libudev1 libusb-1.0-0 linux-firmware locales
  motd-news-config netplan.io python-apt-common python3-apt
  python3-distupgrade python3-gi python3-software-properties snapd
  software-properties-common systemd systemd-sysv systemd-timesyncd
  ubuntu-advantage-tools ubuntu-release-upgrader-core udev
38 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 282 MB of archives.
After this operation, 1,059 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 motd-news-con
fig all 12ubuntu4.2 [4,612 B]
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6 amd64 2
.35-0ubuntu3.1 [3,235 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 base-files am
d64 12ubuntu4.2 [62.7 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc-bin amd6
```

```
sudo apt upgrade
```

команда обновляет все устаревшие пакеты и применить исправления безопасности

```
vladpolskiy@linux:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vladpolskiy@linux:~$
```

Установите необходимые инструменты

Основные элементы:

```
sudo apt-get -y install ca-certificates gnupg curl unzip gdal-bin \
tar wget bzip2 build-essential clang python3-psycopg2 python3-yaml \
python3-requests postgresql-client
```

```

vladpolskiy@linux:~$ sudo apt-get -y install ca-certificates gnupg curl unzip gdal-bin \
tar wget bzip2 build-essential clang python3-psycopg2 python3-yaml \
python3-requests postgresql-client
[sudo] password for vladpolskiy:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20211016).
ca-certificates set to manually installed.
python3-requests is already the newest version (2.25.1+dfsg-2).
python3-requests set to manually installed.
python3-yaml is already the newest version (5.4.1-lubuntu1).
python3-yaml set to manually installed.
tar is already the newest version (1.34+dfsg-1build3).
tar set to manually installed.
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
curl is already the newest version (7.81.0-lubuntu1.3).
curl set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following additional packages will be installed:
  binfmt-support clang-14 dpkg-dev fakeroot g++ g++-11 gcc gcc-11 gdal-data

```

Дополнительные элементы:

```

sudo apt-get -y install munin-node munin protobuf-c-compiler libtiff5-dev
libcairomm-1.0-dev libagg-dev lua5.1 liblua5.1-0-dev

```

```

vladpolskiy@linux:~$ sudo apt-get -y install munin-node munin protobuf-c-compiler
libtiff5-dev
libcairomm-1.0-dev libagg-dev lua5.1 liblua5.1-0-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  jo jq libalgorithm-c3-perl libauthen-sasl-perl libb-hooks-endofscope-perl
  libb-hooks-op-check-perl libcgi-fast-perl libcgi-pm-perl libclass-c3-perl
  libclass-c3-xs-perl libclass-data-inheritable-perl
  libclass-method-modifiers-perl libclass-xsaccessor-perl libclone-perl
  libcommon-sense-perl libdata-optlist-perl libdate-manip-perl libdbil
  libdeflate-dev libdevel-callchecker-perl libdevel-caller-perl
  libdevel-globaldestruction-perl libdevel-lexalias-perl
  libdevel-stacktrace-perl libdist-checkconflicts-perl
  libdynamloader-functions-perl libemail-date-format-perl libencode-locale-perl
  libeval-closure-perl libexception-class-perl libexporter-tiny-perl
  libfcgi-bin libfcgi-perl libfcgi0ldbl libfile-copy-recursive-perl

```

Проверьте предварительные условия, предложенные [openstreetmap-carto](#).

```

FROM ubuntu:bionic

```

```

# # Зависимости стиля

```

```

RUN apt-get update && apt-get install --no-install-recommends -y \
  ca-certificates curl gnupg postgresql-client python3 python3-distutils \
  fonts-hanazono fonts-noto-cjk fonts-noto-hinted fonts-noto-unhinted \
  mapnik-utils nodejs npm ttf-unifont unzip && rm -rf /var/lib/apt/lists/*

```

```

# Kosmtik with plugins, forcing prefix to /usr because bionic sets

```

```
# npm prefix to /usr/local, which breaks the install
RUN npm set prefix /usr && npm install -g kosmtik

РАБОЧИЙ КАТАЛОГ /usr/lib/node_modules/kosmtik/
RUN kosmtik plugins --install kosmtik-overpass-layer \
    --install kosmtik-fetch-remote \
    --install kosmtik-overlay \
    --install kosmtik-open-in-josm \
    --install kosmtik-map-compare \
    --install kosmtik-osm-data-overlay \
    --install kosmtik-mapnik-reference \
    --install kosmtik-geojson-overlay \
    && cp /root/.config/kosmtik.yml /tmp/.kosmtik-config.yml

# Closing section
RUN mkdir -p /openstreetmap-carto
WORKDIR /openstreetmap-carto

USER 1000
CMD sh scripts/docker-startup.sh kosmtik =====
```

установим npm, диспетчер пакетов Node.js. Для этого установите пакет npm с помощью apt:

```
sudo apt install npm
```

```
vladpolskiy@linux:~$ sudo apt install npm
[sudo] password for vladpolskiy:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gyp javascript-common libc-ares2 libdata-dump-perl libfile-basedir-perl
  libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl
  libfont-afm-perl libhtml-form-perl libhtml-format-perl libhtml-tree-perl
  libhttp-cookies-perl libhttp-daemon-perl libhttp-negotiate-perl
  libio-stringy-perl libipc-system-simple-perl libjs-events libjs-highlight.js
  libjs-inherits libjs-is-typedarray libjs-psl libjs-source-map
  libis-sprintf-is libis-typedarray-to-buffer liblwm-protocol-https-perl
```

перейдем в режим суперпользователя

```
sudo -i
```

```
vladpolskiy@linux:~$ sudo -i
root@linux:~#
```

проверим зависимости стиля

```
apt-get update && apt-get install --no-install-recommends -y \
  ca-certificates curl gnupg postgresql-client python3 python3-distutils \
  fonts-hanazono fonts-noto-cjk fonts-noto-hinted fonts-noto-unhinted \
  mapnik-utils nodejs npm ttf-unifont unzip && rm -rf /var/lib/apt/lists/*
```

```
vladpolskiy@linux:~$ sudo -i
root@linux:~# apt-get update && apt-get install --no-install-recommends -y \
  ca-certificates curl gnupg postgresql-client python3 python3-distutils \
  fonts-hanazono fonts-noto-cjk fonts-noto-hinted fonts-noto-unhinted \
  mapnik-utils nodejs npm ttf-unifont unzip && rm -rf /var/lib/apt/lists/*
Hit:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://ru.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 324 kB in 1s (350 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package ttf-unifont is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
 fonts-unifont

E: Package 'ttf-unifont' has no installation candidate
root@linux:~# █
```

E: Package 'ttf-unifont' has no installation candidate

пакет 'ttf-unifont' установим позже (сейчас не критично)

к проверке установки kosmtik вернемся после установки Node.js

Для последующих шагов установки мы предполагаем, что:

cd - по умолчанию используется ваш домашний каталог.

Настроить swap

SWAP (своп) — это механизм виртуальной памяти, при котором часть данных из оперативной памяти (ОЗУ) перемещается на хранение на HDD (жёсткий диск), SSD (твёрдотельный накопитель).

Импорт картографических данных и управление ими занимают много оперативной памяти, и обычно требуется swap.

Чтобы проверить, настроен ли раздел подкачки в вашей системе, используйте одну из следующих двух команд:

```
swapon -s
```

```
root@linux:~# swapon -s
Filename                                Type              Size              Used              P
riority
/swap.img                               file              4194300           0                 -
2
```

в данном случае установлено:

- 4gb файла подкачки

```
free -m
```

Отображение количества свободной и используемой памяти в системе (отметьте строку,

указывающую Swap):

```
root@linux:~# free -m
              total        used         free       shared    buff/cache   available
Mem:           15988         554         9557           2         5876        15093
Swap:          4095           0         4095
```

в данном случае используется:

- ОЗУ: 544mb из доступных 16gb
- файл подкачки: 0mb из доступных 4gb.

Если у вас нет активного раздела подкачки, особенно если ваша физическая память мала, вам следует добавить файл подкачки. Создайте файл подкачки размером 2G в корневой файловой системе:

```
sudo fallocate -l 2G /swapfile
```

```
root@linux:~# sudo fallocate -l 2G /swapfile
root@linux:~#
```

Установим правильный тип разрешений.

```
sudo chmod 600 /swapfile
```

```
root@linux:~# sudo chmod 600 /swapfile
```

Используйте mkswap утилиту, чтобы настроить файл как область подкачки Linux:

```
sudo mkswap /swapfile
```

```
root@linux:~# sudo mkswap /swapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=e869d3f9-7178-4137-b24a-8ca6aa9861cd
```

Включить файл подкачки

```
sudo mkswap /swapfile
```

```
root@linux:~# sudo swapon /swapfile
```

проверим результат

```
free -m
```

```
root@linux:~# free -m
              total        used         free       shared    buff/cache   available
Mem:           15988         563         9544           2         5880        15083
Swap:          6143           0         6143
```

Чтобы сделать изменение постоянным, откройте /etc/fstab файл и добавьте следующую строку:

[/etc/fstab](#)

```
/swapfile swap swap defaults 0 0
```


Откроем файл `fstab`, находящийся в папке `ets` в редакторе:

```
vi /etc/fstab
```

```
root@linux:~# vi /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-jrw4QwMxqmJ79HHL40Qqhgeu7arDVuEX7m0XAxo6FZweuDMYBoOz7
wKhmMxC3dKS / ext4 defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/1a8376de-3530-4bea-9ac0-d85acfb9a56 /boot ext4 defaults 0 1
/swap.img none swap sw 0 0
/swapfile swap swap defaults 0 0
~
```

- `i` -начало редактирования
- `esc` -конец редактирования
- `:wq!` -сохранение и выход из редактора

```
:wq!
```

[Swap в linux](#)

[Рекомендованный размер swap в linux](#)

Настройка переменных локализации

- Locale — это набор переменных среды, которые определяют язык, страну и настройки кодировки символов (или любые другие особые предпочтения варианта) для ваших приложений и сеанса оболочки в системе Linux. Эти переменные среды используются системными библиотеками и локальными приложениями в системе.
- Конфигурационный файл находится `/etc/default/locale`.
- Просмотрим, какие локали в настоящее время определены для текущей учетной записи пользователя:

```
locale
```

```
root@linux:~# locale
LANG=en_US.UTF-8
LANGUAGE=
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

Чтобы установить локализацию ru_RU:

```
export LANGUAGE=ru_RU.UTF-8
export LANG=ru_RU.UTF-8
export LC_ALL=ru_RU.UTF-8
```

- Экспортированные переменные можно поместить в файл /etc/environment.
- Новые локали также можно создать, выполнив:

```
sudo locale-gen ru_RU ru_RU.UTF-8
sudo dpkg-reconfigure locales
```

- если ошибка изменения локализации:
- `-bash: warning: setlocale: LC_ALL: cannot change locale`

Создание пользователя UNIX

- Мы предполагаем, что вы уже создали пользователя для входа во время установки Ubuntu Server, который будет использоваться для запуска сервера тайлов. Предположим, что выбранное вами имя пользователя — alisa. В этом документе каждый раз, когда будет упоминается alisa, замените его своим фактическим именем пользователя.
- Если вам нужно создать нового пользователя:

```
sudo useradd -m alisa
sudo passwd alisa
```

```
root@linux:~# sudo useradd -m alisa
sudo passwd alisa
New password:
Retype new password: █
```

- Установите пароль при появлении запроса.
- Повторите пароль для правильности ввода.

Установить Git

Иногда [Git](#) может быть уже предустановлен.

```
git --version
```

```
root@linux:~# git --version
git version 2.34.1
```

в случае отсутствия установим:

```
apt-get install -y git
```

```
root@linux:~# apt-get install -y git
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет git самой новой версии (1:2.34.1-lubuntul.4).
git помечен как установленный вручную.
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов
, и 0 пакетов не обновлено.
```

- Справочник Pro Git

Установка библиотеки Mapnik

Mapnik используется для рендеринга данных OpenStreetMap в тайлы, управляемые веб-сервером Apache через renderd и mod_tile. Эта установка предусматривает, что все команды выполняются в окружении обычной учетной записи (не root) с использованием «sudo». Не пытайтесь и не делайте все в окружении root; система не будет работать. (выход: exit)

```
exit
```

```
root@linux:~# exit
logout
```

Создайте в корне каталог src для загрузки исходников

```
mkdir -p ~/src
```

```
vladpolskiy@linux:~$
mkdir -p ~/src
```

и перейдите в него:

```
cd ~/src
```

```
vladpolskiy@linux:~$
cd ~/src
vladpolskiy@linux:~/src$
```

```
sudo apt install python-is-python3
```

df

Зависимость FreeType в Ubuntu 16.04 LTS

Установить библиотеку Марпик из пакета

Или установите Марпик из исходников

Установить Boost из пакета

В качестве альтернативы установите последнюю версию Boost из исходного кода

Установите HarfBuzz из пакета

Установите HarfBuzz из исходного кода

Собрать библиотеку Марпик из исходников

Убедитесь, что Марпик был правильно установлен

Настроить брандмауэр

Установить HTTP-сервер Apache

Как узнать IP-адрес вашего сервера

Установить Mod_tile из пакета

Mod_tile/render для Ubuntu 18.04 и Ubuntu 20.04

Mod_tile/рендеринг для Ubuntu 21.04

Установить Mod_tile из исходников

Установка Python

Установите Yaml и диспетчер пакетов для Python

Установите утилиты Mapnik

Установить openstreetmap-carto

Установите шрифты, необходимые для openstreetmap-carto

Старый унифонт Средний шрифт

Установите Node.js

Версия дистрибутива из менеджера пакетов APT

Установите Node.js с помощью инструмента управления версиями

Проверьте версии Node.js

Установите carto и создайте XML-таблицу стилей Mapnik

Установите PostgreSQL и PostGIS

Создайте экземпляр PostGIS

Установите переменные среды

Добавьте пользователя и предоставьте доступ к базе данных gis

Включение удаленного доступа к PostgreSQL

Настройка базы данных

Минимальные требования к настройке

Необязательные дополнительные требования к настройке

Установить Osm2pgsql

Сгенерировать Osm2pgsql из исходников

Получите извлечение данных OpenStreetMap

Загрузить данные в PostGIS

Создайте папку данных

Создание индексов и предоставление пользователям

Настроить визуализацию

Настроить Apache

Предварительный рендеринг тайлов

Устранение неполадок Apache, mod_tile и renderd

Формат имен тайлов сервера тайлов OpenStreetMap

Развертывание собственной карты Slippy Map

OpenLayers

Листовка

Дополнения и файлы

- [Установка сервера тайлов OpenStreetMap в Ubuntu](#)
- [Создание тайлового сервера вручную \(Ubuntu 24.04\)](#)
- [Создание тайлового сервера вручную \(Ubuntu 20.04\)](#)
- [PostgreSQL](#)
- [Поднимаем собственный репозиторий пакетов для Ubuntu \(Debian\)](#)
- [Как установить все доступные обновления для выпуска Ubuntu перед обновлением](#)
- [mod_tile](#)

Упомянутые команды:

- `sudo apt update` — обновляет список пакетов последней информацией из настроенных репозиториев.
- `sudo apt update` — устанавливает доступные обновления для вашей текущей версии Ubuntu.
- `sudo apt dist-upgrade` — выполняет более тщательное обновление, которое более тщательно обрабатывает зависимости пакетов.
- `sudo apt autoremove` — удаляет из системы неиспользуемые или устаревшие пакеты.
- `sudo restart` — перезагружает систему, чтобы все изменения вступили в силу.
- `sudo apt install update-manager-core` — устанавливает основной пакет менеджера обновлений, если он еще не установлен.
- `sudo do-release-upgrade` — запускает процесс обновления выпуска Ubuntu.

From:

<http://www.book51.ru/> - **book51.ru**

Permanent link:

http://www.book51.ru/doku.php?id=software:linux_server:tile_map_server

Last update: **2024/05/19 22:33**

