Руководство по установке программного комплекса «TENEMENT»

Оглавление

Установка комплекса Tenement

Для работы серверной части комплекса Tenement требуется Linux-подобная операционная система. Ниже приведена инструкция по установке комплекса на ОС Debian версии 10 и выше.

Установка производится удаленно, при помощи программы PuTTY.

Установка PostgreSQL 12 на ОС Debian

Подключение репозитория и обновление списка пакетов в системе

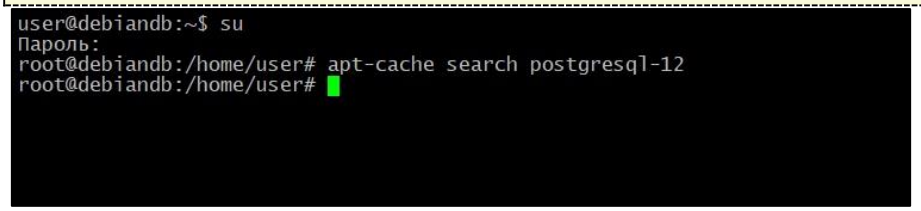
Установку и настройку PostgreSQL необходимо выполнять с правами пользователя root, поэтому давайте сразу переключимся на root.

Для этого пишем команду su и вводим пароль.

Далее проверим, нет ли в системе необходимых нам пакетов.

Для этого вводим следующую команду.

```
apt-cache search postgresql-12
```



В данном примере, в Debian 10 нужной нам версии PostgreSQL нет, поэтому нам нужно подключить дополнительный репозиторий от разработчиков. Если у Вас более новая версия Debian и в стандартных репозиториях есть 12 версия PostgreSQL, то дополнительный репозиторий Вам подключать не нужно, т.е. данный шаг Вы можете пропустить и перейти к разделу: «Установка необходимых пакетов для PostgreSQL».

Для подключения репозитория необходимо ввести следующую команду, которая создает файл в источниках с адресом нужного репозитория.

```
sh -c 'echo " deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg
main " >> /etc/apt/sources.list.d/pgdg.list'
```

Примечание! Здесь обязательно стоить отметить, что этот репозиторий предназначен для Debian 10, для других версий адрес репозитория будет другим, например, если Вам нужно установить PostgreSQL на Debian 9, то в адресе репозитория вместо buster напишите stretch, т.е. замените кодовое имя версии.

После подключения репозитория нам необходимо импортировать ключ подписи репозитория, для этого вводим команду.

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
```

Обновляем список пакетов в системе, командой:

```
apt-get update
```

```
user@debiandb:~$ su
Пароль:
root@debiandb:/home/user# apt-cache search postgresql-12
root@debiandb:/home/user# sh -c 'echo " deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main " >> /etc/apt/so
urces.list.d/pgdg.list'
root@debiandb:/home/user# wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
OK
root@debiandb:/home/user# apt-get update
Сущ:1 http://deb.debian.org/debian buster InRelease
Сущ:2 http://deb.debian.org/debian buster-updates InRelease
Сущ:3 http://security.debian.org/debian-security buster/updates InRelease
Пол:4 http://apt.postgresql.org/pub/repos/apt buster-pgdg InRelease [58,6 kB]
Пол:5 http://apt.postgresql.org/pub/repos/apt buster-pgdg/main amd64 Packages [159 kB]
Получено 218 kB за 9с (24,0 kB/s)
Чтение списков пакетов… Готово
root@debiandb:/home/user#
```

Для проверки того, что теперь нам доступны пакеты PostgreSQL 12, снова запустим команду поиска пакетов.

```
apt-cache search postgresql-12
```

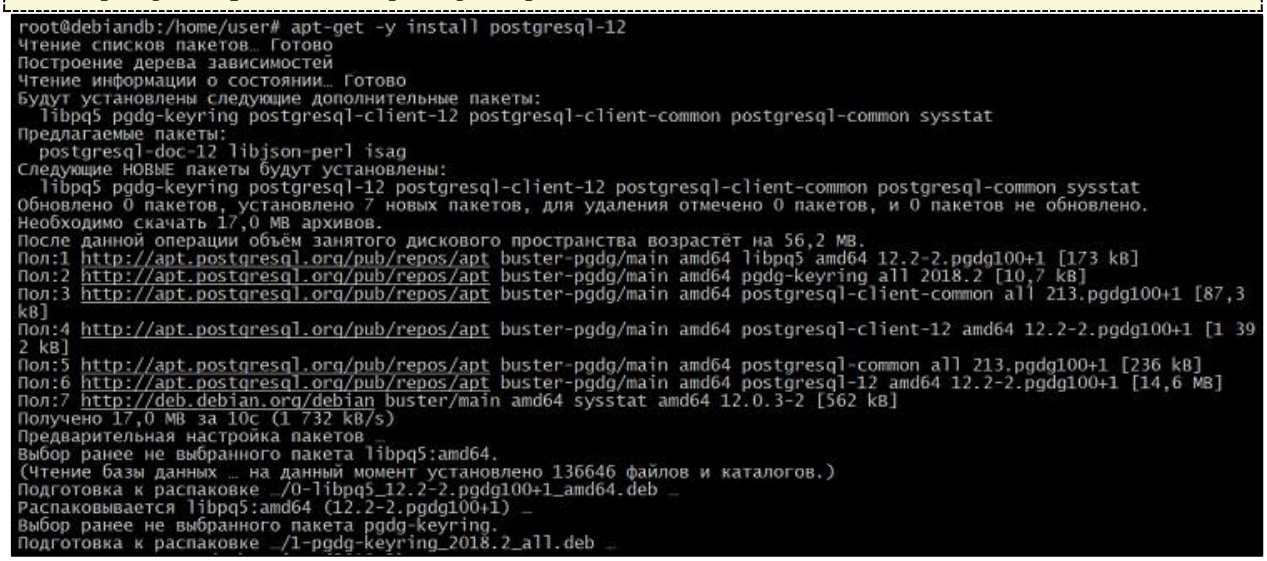
```
root@debiandb:/home/user# apt-cache search postgresql-12
pgagent - job scheduling engine for PostgreSQL
postgresql-pgsphere - Spherical data types for PostgreSQL
postgresql-q3c - PostgreSQL extension used for indexing the sky
postgresql-12 - object-relational SQL database, version 12 server
postgresql-12-asn1oid - ASN.1 OID data type for PostgreSQL
postgresql-12-asn1oid-dbgsym - debug symbols for postgresql-12-asn1oid
postgresql-12-bgw-replstatus - report whether PostgreSQL node is master or standby
postgresql-12-bgw-replstatus-dbgsym - debug symbols for postgresql-12-bgw-replstatus
postgresql-12-cron - Run periodic jobs in PostgreSQL
postgresql-12-cron-dbgsym - debug symbols for postgresql-12-cron
postgresql-12-cstore-fdw - PostgreSQL foreign data wrapper for columnar storage
postgresql-12-cstore-fdw-dbgsym - debug symbols for postgresql-12-cstore-fdw
postgresql-12-dbgsym - debug symbols for postgresql-12
postgresql-12-debversion - Debian version number type for PostgreSQL
postgresql-12-debversion-dbgsym - debug symbols for postgresql-12-debversion
postgresql-12-dirtyread - Read dead but unvacuumed tuples from a PostgreSQL relation
postgresql-12-dirtyread-dbgsym - debug symbols for postgresql-12-dirtyread
postgresql-12-first-last-agg - PostgreSQL extension providing first and last aggregate functions
postgresql-12-first-last-agg-dbgsym - debug symbols for postgresql-12-first-last-agg
postgresql-12-hll - HyperLogLog extension for PostgreSQL
postgresql-12-hll-dbgsym - debug symbols for postgresql-12-hll
postgresql-12-hypopg - PostgreSQL extension adding support for hypothetical indexes.
postgresql-12-hypopg-dbgsym - debug symbols for postgresql-12-hypopg
postgresql-12-icu-ext - PostgreSQL extension exposing functionality from the ICU library
postgresql-12-icu-ext-dbgsym - debug symbols for postgresql-12-icu-ext
postgresql-12-ip4r - IPv4 and IPv6 types for PostgreSQL 12
postgresql-12-ip4r-dbgsym - debug symbols for postgresql-12-ip4r
postgresql-12-jsquery - PostgreSQL JSON query language with GIN indexing support
postgresql-12-jsquery-dbgsym - debug symbols for postgresql-12-jsquery
postgresql-12-mimeo - specialized, per-table replication between PostgreSQL instances
postgresql-12-mysql-fdw - Postgres 12 Foreign Data Wrapper for MySQL
postgresql-12-mysql-fdw-dbgsym - debug symbols for postgresql-12-mysql-fdw
postgresql-12-numeral - numeral datatypes for PostgreSQL
postgresql-12-numeral-dbgsym - debug symbols for postgresql-12-numeral
postgresql-12-ogr-fdw - PostgreSQL foreign data wrapper for OGR
postgresql-12-ogr-fdw-dbgsym - debug symbols for postgresql-12-ogr-fdw
```

Теперь нужные пакеты нам доступны и мы можем переходить к установке PostgreSQL 12.

Установка необходимых пакетов для PostgreSQL

Для установки PostgreSQL 12 и базовых стандартных утилит необходимо установить пакет postgresql-12, это делается следующей командой.

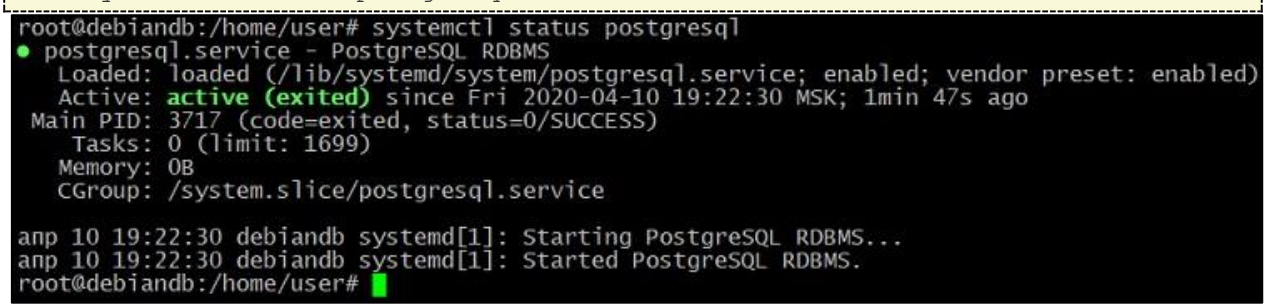
```
apt-get -y install postgresql-12
```



Проверка установки.

Чтобы проверить, установился и запущен ли Postgres, выполним следующую команду, которая покажет статус сервиса PostgreSQL.

```
systemctl status postgresql
```



PostgreSQL 12 установлен и работает.

Настройка PostgreSQL 12 в Debian 10

PostgreSQL установлен, теперь необходимо выполнить настройку, в частности создать пользователя, указать какие сетевые интерфейсы будет прослушивать сервер, а также разрешить подключения по сети.

Создание пользователя и базы данных в PostgreSQL

После установки к серверу PostgreSQL мы можем подключиться только с помощью системного пользователя postgres, причем без пароля.

Давайте переключимся на пользователя postgres (*данная учетная запись была создана автоматически во время установки PostgreSQL*).

```
su - postgres
```

Затем запускаем утилиту psql – это консоль для PostgreSQL.

```
psql
```

Первым делом нам нужно задать пароль для пользователя postgres.

```
\password postgres
```

Затем создаем нового пользователя на сервере PostgreSQL, так как работать от имени postgres крайне не рекомендуется.

```
CREATE ROLE tenement WITH NOSUPERUSER NOCREATEDB NOREPLICATION CREATEROLE
CONNECTION LIMIT 10 INHERIT LOGIN PASSWORD 'FwicOzcrvn';
```

где tenement – это имя пользователя, ‘FwicOzcrvn’ – это его пароль.

Далее создадим базу данных.

```
CREATE DATABASE tenement WITH OWNER = tenement ENCODING = 'UTF8'
LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8' CONNECTION LIMIT = 200;
```

где tenement – это имя новой базы данных.

Все готово, выходим из консоли.

```
\q
```

Создать схему tenement в БД tenement и пользователя для приложения.

Для этого зайдем в psql под пользователем tenement

```
psql -h localhost tenement tenement
```

Создаем пользователя для приложения tenementapi

```
CREATE ROLE tenementapi WITH NOSUPERUSER NOCREATEDB NOCREATEROLE
NOREPLICATION CONNECTION LIMIT 200 INHERIT LOGIN PASSWORD 'e5oqJkDcfV';
```

где tenementapi – это имя пользователя, ‘e5oqJkDcfV’– это его пароль.

Создаем основную схему БД.

```
CREATE SCHEMA tenement;
```

Укажем привилегии пользователя tenementapi в схеме tenement.

```
GRANT USAGE ON SCHEMA tenement TO tenementapi;


ALTER DEFAULT PRIVILEGES IN SCHEMA tenement GRANT SELECT, INSERT, UPDATE,
DELETE ON TABLES TO tenementapi;
```

```
    ALTER DEFAULT PRIVILEGES IN SCHEMA tenement GRANT USAGE, SELECT ON
SEQUENCES TO tenementapi;
```

Все готово, выходим из консоли.

```
    \q
```

Для переключения обратно на root вводим exit.

```
    exit
```

Разрешаем подключение к PostgreSQL по сети

По умолчанию PostgreSQL прослушивает только адрес localhost, поэтому для того чтобы была возможность подключаться по сети, нужно указать, какие сетевые интерфейсы будет просушивать PostgreSQL. Для примера укажем, что прослушивать нужно все доступные интерфейсы. Если у Вас несколько сетевых интерфейсов, и Вы хотите, чтобы PostgreSQL использовал только один конкретный, то Вы его можете указать именно здесь.

Открываем файл postgresql.conf, например, редактором nano.

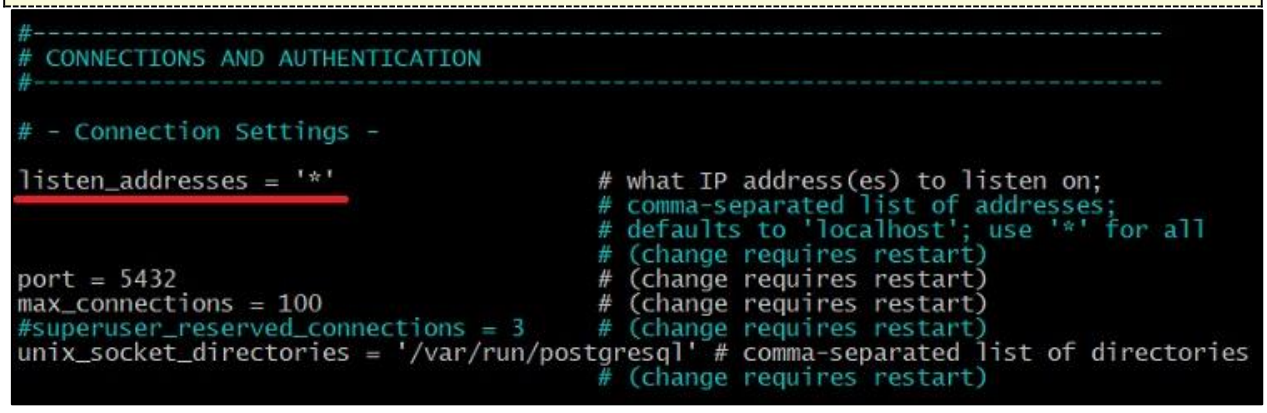
```
    nano /etc/postgresql/12/main/postgresql.conf
```

Находим следующую строку.

```
    #listen_addresses = 'localhost'
```

и вносим следующие изменения (*вместо звездочки Вы в случае необходимости указываете IP адрес нужного интерфейса*).

```
    listen_addresses = '*'
```



Сохраняем изменения сочетанием клавиш CTRL+O и подтверждаем нажатием Enter, затем просто закрываем редактор nano сочетанием клавиш CTRL+X.

Теперь разрешим подключение из сети, для примера разрешим подключаться из сети 192.168.1.0/24 с методом аутентификации md5.

Для этого открываем файл pg_hba.conf

```
    nano /etc/postgresql/12/main/pg_hba.conf
```

Ищем следующие строки.

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
```

И указываем нужную нам сеть (*если IPv6 Вы не будете использовать, то можете закомментировать соответствующие строки знаком #*).

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             192.168.1.0/24          md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
```

Далее точно так же сохраняем изменения сочетанием клавиш CTRL+O, подтверждаем нажатием Enter и закрываем редактор nano сочетанием клавиш CTRL+X.

Перезапускаем PostgreSQL, чтобы изменения вступили в силу.

```
systemctl restart postgresql
```

Установка стандартной версии JRE/JDK

Установим стандартную версию Java, которая поставляется вместе с Debian. По умолчанию вместе с Debian 10 идет Open JDK 11, открытая версия JRE и JDK, совместимая с Java 11.

Обновляем список пакетов в системе, командой.

```
apt-get update
```

Запросим версию Java (чтобы проверить, установлена ли Java в данной системе).

```
java -version
```

Если на сервере, Java не установлена, вы увидите такой вывод.

```
-bash: java: command not found
```

Установим OpenJDK.

```
sudo apt install default-jre
```

Данная команда установит Java Runtime Environment (JRE), что позволит вам запускать почти все программы Java.

Запросите версию установленной программы.

```
java -version
```

Вы увидите такой вывод.

```
root@debian:/home/user# java -version
openjdk version "11.0.18" 2023-01-17
OpenJDK Runtime Environment (build 11.0.18+10-post-Debian-1deb11u1)
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Debian-1deb11u1, mixed mode, sharing)
```

Кроме JRE понадобиться Java Development Kit (JDK), чтобы скомпилировать и запустить определенное программное обеспечение на базе Java. Чтобы установить JDK, выполните следующую команду.

```
sudo apt install default-jdk
```

Убедитесь, что версия JDK установлена, проверив версию javac, компилятора Java.

```
javac -version
```

```
root@debian:/home/user# javac -version
javac 11.0.18
```

Установка web-сервера Nginx в Debian

Для входа под учетной записью суперпользователя воспользуйтесь командой.

```
su
```

Введите пароль root, после чего будет доступна установка и настройка Nginx.

Обновите репозитории и операционную систему, для этого понадобятся команда.

```
sudo apt update && sudo apt upgrade –y
```

Выполняем установку wget для загрузки файлов по сети.

```
apt-get install -y wget
```

Скачиваем и добавляем ключ Nginx Inc. на нашу систему.

```
wget --quiet -O - http://nginx.org/keys/nginx_signing.key | apt-key add -
```

Устанавливаем Nginx из ветки Stable.

```
echo "deb http://nginx.org/packages/debian/ $(lsb_release -sc)
nginx">/etc/apt/sources.list.d/nginx.list
```

```
echo "deb-src http://nginx.org/packages/debian/ $(lsb_release -sc)
nginx">>/etc/apt/sources.list.d/nginx.list
```

Обновляем список пакетов.

```
apt-get update
```

Устанавливаем Nginx и OpenSSL.

```
apt-get install -y nginx
apt-get install -y openssl
```

После установки запускаем Nginx.

```
systemctl start nginx
```

Проверим факта запуска Nginx.

```
systemctl status nginx
```

Строка «Active: active (running)» говорит о работе Nginx.

Проверим открытые порты.

```
netstat -ltupn | grep nginx
```

tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 129551/nginx: master

Видим, что Nginx ожидает соединения на 80 порту на всех сетевых интерфейсах.

Базовая настройка web-сервера Nginx

Создадим директорию для хранения SSL сертификатов и DH-ключей, а также создаем файл с параметрами для DHE-шифров.

```
mkdir /etc/nginx/ssl
openssl dhparam -out /etc/nginx/ssl/dhparams.pem 2048
```

Создадим директории для хранения настроек Web-сайтов.

```
mkdir /etc/nginx/sites-available /etc/nginx/sites-enabled
```

Скачать уже готовый файл для Nginx.

```
wget
https://gist.githubusercontent.com/CHERTS/8e9ecf4fbfb765556311a88e5106174b/raw/nginx.conf -O /etc/nginx/nginx.conf
```

Запускаем команду ***echo $PATH*** и проверяем содержится ли в путях /usr/local/sbin. Если нет, тогда вручную прописываем путь: ***PATH=/usr/sbin/:$PATH***.

После этого проверяем конфигурацию Nginx.

```
nginx -t
```

Если ошибок нет, то перезагружаем конфигурацию Nginx.

```
nginx -s reload
```

Теперь по адресу *http://IP-адрес-вашего-сервера* Вы сможете увидеть приветственную страницу «Welcome to nginx!».

Чтобы выяснить ip-адрес Вашего сервера, выполняем.

```
ip addr show <название_сетевого_интерфейса>| grep inet | awk '{ print $2; }' | sed 's/\/.*$//'.
```

“Название_сетевого_интерфейса” можно получить, запустив команду ***ipconfig***. Адрес можно выбрать любой из полученного списка.

Настройка и запуск серверного приложения Tenement.

Внимание приведена примерная инструкция установки и запуска приложения. Уточнить порядок действий нужно у технической поддержки.

Создадим пользователя, под которым будем запускать приложение.

```
sudo adduser tenementapi-back
```

где tenementapi-back имя пользователя

переключаемся на пользователя tenementapi-back

```
su tenementapi-back
```

переходим в домашний каталог

```
cd
```

создаем каталоги

```
mkdir /home/tenementapi-back/tenement
mkdir /home/tenementapi-back/files
```

переносим файлы с именами: tenementapi и tenementapi-tenement из полученного вами пакета в каталог /home/tenementapi-back/

переносим файл с именем: tenement-1.0-SNAPSHOT.jar из полученного вами пакета в каталог /home/tenementapi-back/tenement

запустим приложение

```
./tenementapi-tenement start
```

переносим файл с именем: tenement-front.conf из полученного вами пакета в каталог /etc/nginx/conf.d/

создаем каталоги

```
mkdir /web/tenement-front/www
mkdir /web/tenement-front/logs
```

переназначаем владельца каталога

```
chown -R nginx /web/tenement-front/logs
```

копируем содержимое каталога www из полученного вами пакета в каталог /web/tenement-front/www

перезапускаем nginx

```
systemctl restart nginx
```

Описание расположения файлов системы

Компонентов серверной части приложения располагаются в каталоге /home/tenementapi-back/

Скрипты tenementapi и tenementapi-tenement отвечают за запуск приложения.

В каталоге /home/tenementapi-back/tenement/ расположен, основной исполняемый файл tenement-1.0-SNAPSHOT.jar, а также лог файлы с расширением *.out.

В каталоге /home/tenementapi-back/files/ хранятся файлы, которые были загружены в программу во время ее работы, либо были сгенерированы программой.

В каталоге /web/tenement-front/www/ расположены файлы web-интерфейса.

В каталоге /etc/nginx/conf.d/ расположены файлы конфигурации web-сервера nginx.

В каталоге /var/lib/postgresql/13/main/ расположены файлы PostgreSQL включая файлы БД tenement.

Контакты для обращения в службу поддержки:

e-mail: **support@prog-uchet.ru**

В случае получения обращения на электронную почту в приделах рабочего времени службы технической поддержки, они принимаются к рассмотрению в течение одного рабочего часа после получения.

В случае получения обращения на электронную почту вне приделов рабочего времени службы технической поддержки, они принимаются к рассмотрению в течение рабочего часа с начала следующего рабочего дня.

Режим работы службы технической поддержки: Пн-Пят с 9:00 по 18:00 (Хабаровский край)