ООО «Программы учета»

Программный комплекс «LIGHTING»

Руководство по установке программного комплекса «LIGHTING»

Хабаровск 2024

Оглавление

1. Установка комплекса «LIGHTING»

Для работы серверной части комплекса потребуется Linux-подобная операционная система. Ниже представлена инструкция по установке комплекса на ОС Debian 12.

Установка производится удаленно, при помощи программы Windows Terminal.

2. Установка PostgreSQL 15 на ОС Debian

3. Подготовка к установке

Установку и настройку СУБД PostgreSQL необходимо выполнять с правами пользователя root, поэтому давайте сразу переключимся на root. Для этого пишем команду su и вводим пароль.

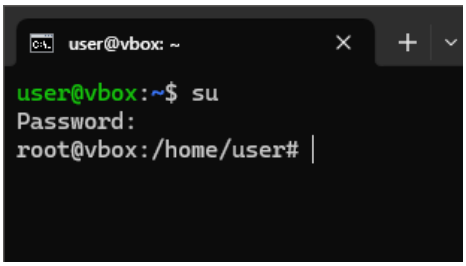

Рисунок 1

Далее проверим, присутствуют ли в системе необходимые нам пакеты. Для этого вводим следующую команду:

```
    apt-cache search postgresql-15
```

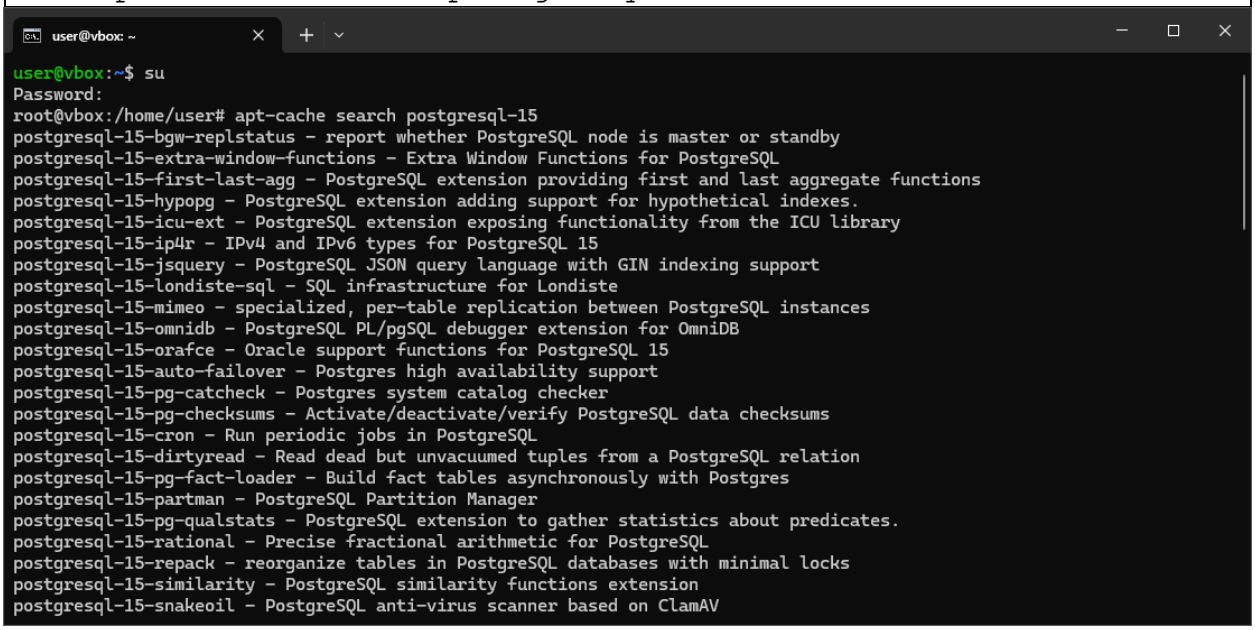

Рисунок 2

В данном примере в Debian 12 присутствует нужная нам версия PostgreSQL 15, и можно переходить к разделу «Установка необходимых пакетов для PostgreSQL». Однако, если поиск пакетов не выдал результатов, необходимо добавить стандартные репозитории Debian 12.

```
    echo -e "deb http://deb.debian.org/debian bookworm main
non-free-firmware\n\
    deb-src http://deb.debian.org/debian bookworm main\n\
    deb          http://security.debian.org/debian-security
bookworm-security main\n\
    deb-src      http://security.debian.org/debian-security
bookworm-security main\n\
```

```
    deb    http://deb.debian.org/debian    bookworm-updates
main\n\
    deb-src  http://deb.debian.org/debian  bookworm-updates
main" >> /etc/apt/sources.list
```

Рисунок 3

После добавления репозиториев необходимо обновить список пакетов, выполнив команду:

```
    apt update
```

Рисунок 4

После обновления списка пакетов должна появиться нужная нам версия PostgreSQL 15, и мы можем перейти к ее установке.

4. Установка необходимых пакетов для PostgreSQL

Для установки PostgreSQL и стандартных базовых утилит необходимо установить пакет postgresql-15, это делается следующей командой.

```
apt -y install postgresql-15
```

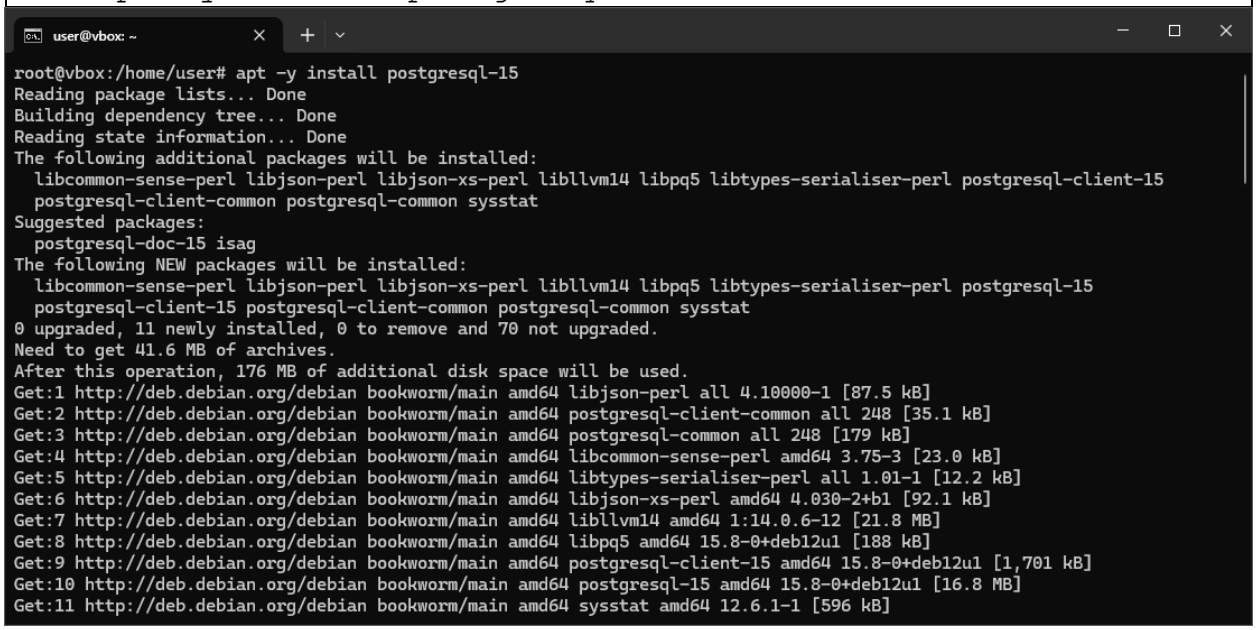

Рисунок 5

5. Проверка установки

Чтобы убедиться, что Postgres установлен и запущен, выполним следующую команду, которая отобразит статус сервиса PostgreSQL.

```
systemctl status postgresql
```

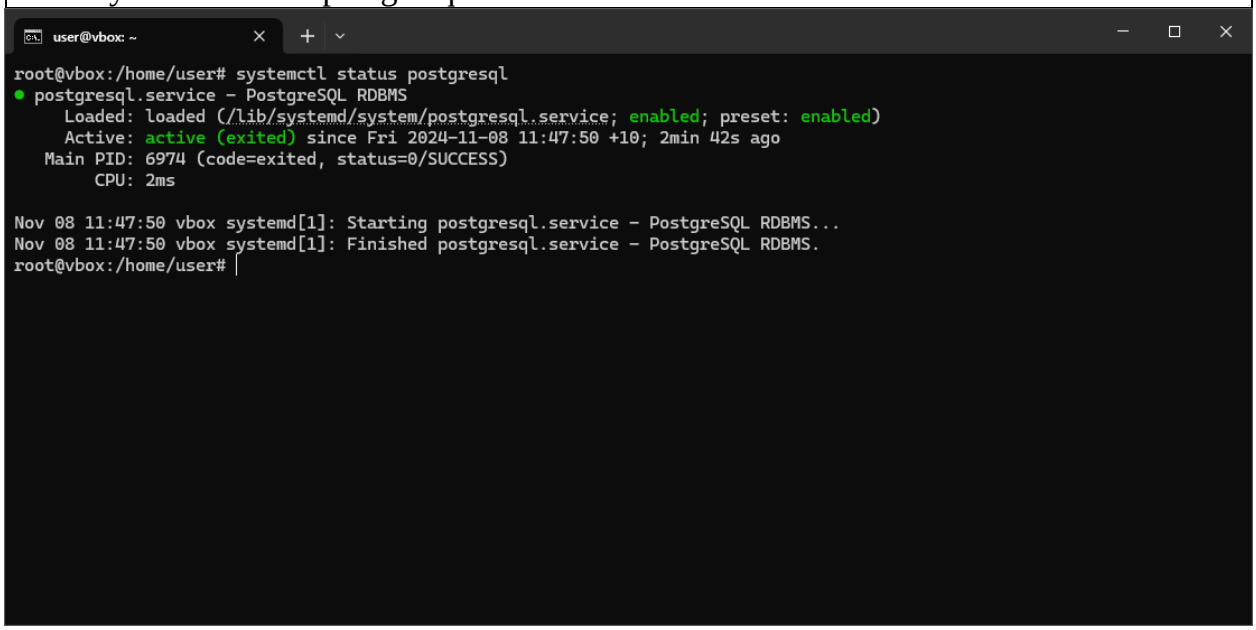

Рисунок 6

PostgreSQL 15 установлен и запущен.

6. Настройка PostgreSQL 15 в Debian 12

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

7. Создание пользователей 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
```

Рисунок 7

Затем запускаем утилиту psql – это консоль для PostgreSQL.

```
psql
```

Рисунок 8

Первым делом нам нужно задать пароль для пользователя postgres.

```
\password postgres
```

Рисунок 9

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

```
    CREATE USER lighting WITH PASSWORD 'd14MrHN4DKPFdJP'
CREATEDB CREATEROLE;
    CREATE USER master WITH PASSWORD '441XkTObxpH5Zyy'
CREATEDB;
    CREATE USER editor WITH PASSWORD 'h125RMnI1aXYBjS';
    CREATE USER manager WITH PASSWORD '0At825q4a0qjLWz';
    CREATE USER viewer WITH PASSWORD 'QG9aS58N2opK3vc';
```

Рисунок 10

Далее создадим базу данных.

```
CREATE DATABASE ownlight WITH OWNER = lighting;
```

Рисунок 11

Все готово, выходим из консоли.

```
\q
```

Рисунок 12

Теперь создадим схему ownlight в базе данных ownlight. Для этого зайдем в psql под пользователем lighting.

```
psql -h localhost -U lighting -d ownlight
```

Рисунок 13

Создаем основную схему.

```
CREATE SCHEMA ownlight;
```

Рисунок 14

Укажем привилегии для других созданных пользователей.

```
GRANT ALL PRIVILEGES ON DATABASE ownlight TO master;
GRANT CONNECT ON DATABASE ownlight TO editor;
GRANT SELECT ON ALL TABLES IN SCHEMA ownlight TO editor;
GRANT CONNECT ON DATABASE ownlight TO manager;
```

```
GRANT CONNECT ON DATABASE ownlight TO viewer;
```

Рисунок 15

Все готово, выходим из консоли.

```
\q
```

Рисунок 16

Для того чтобы переключиться обратно на пользователя root, используем команду.

```
exit
```

Рисунок 17

8. 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, в редакторе nano.

```
nano /etc/postgresql/15/main/postgresql.conf
```

Рисунок 18

Находим следующую строку.

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

Рисунок 19

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

```
listen_addresses = '*'
```

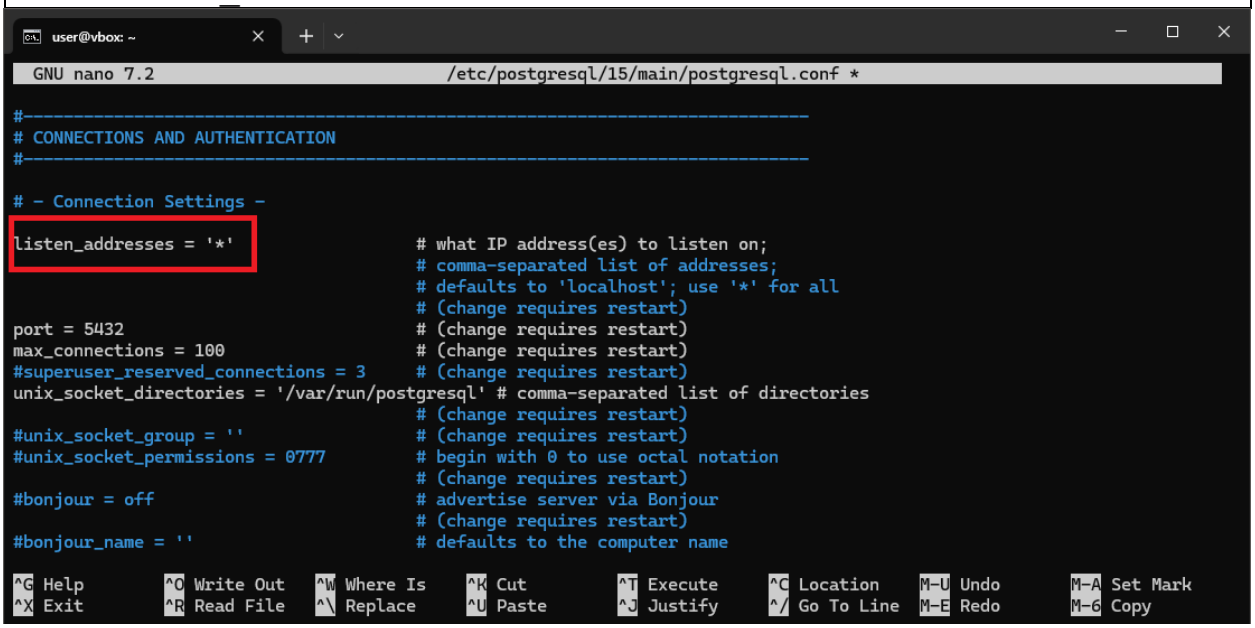

Рисунок 20

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

Теперь разрешим подключение из сети, для примера разрешим подключаться из сети 192.168.1.0/24. Для этого открываем файл pg_hba.conf.

```
nano /etc/postgresql/15/main/pg_hba.conf
```

Рисунок 21

Ищем следующие строки.

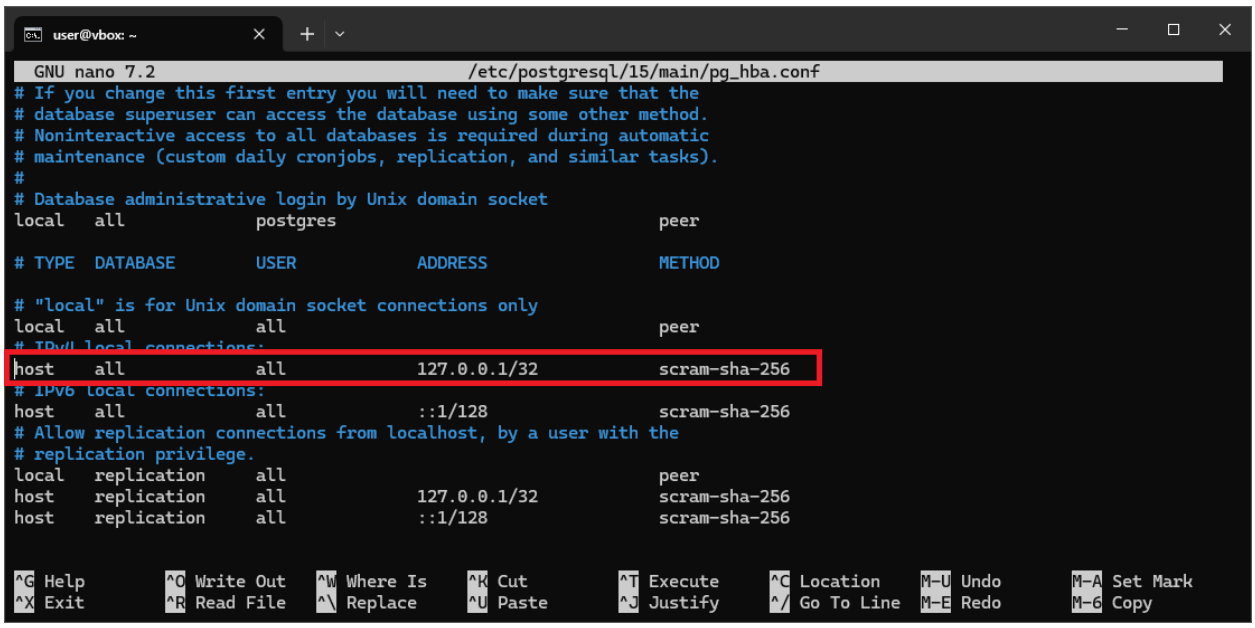

Рисунок 22

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

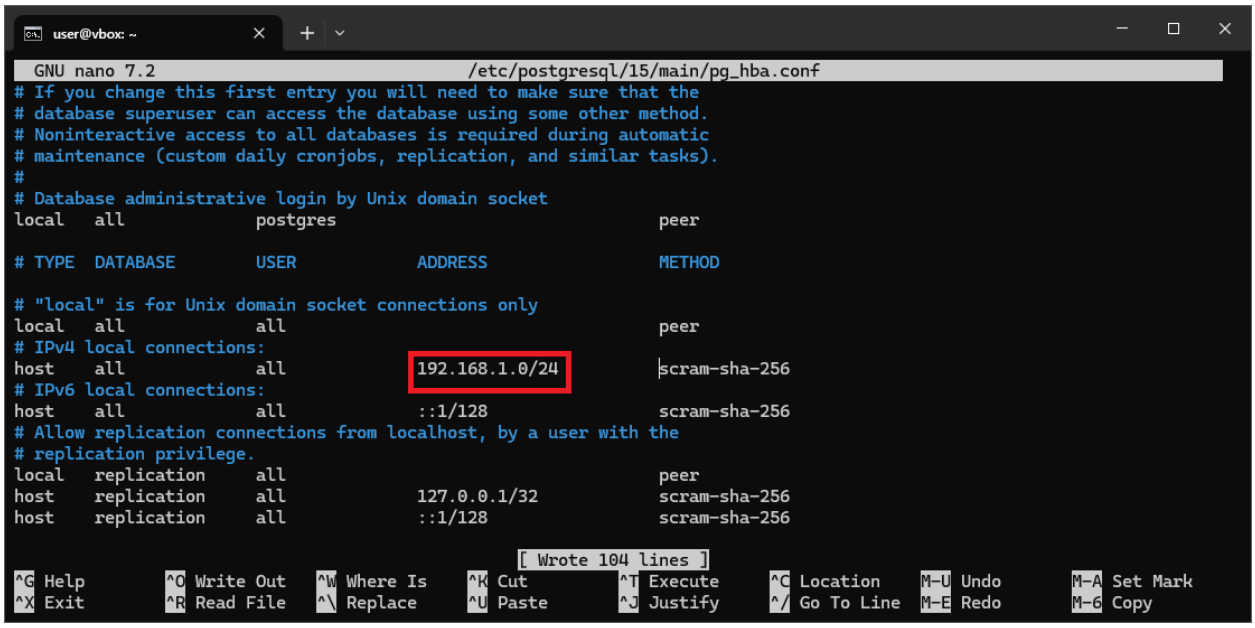

Рисунок 23

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

Перезапускаем PostgreSQL, чтобы изменения вступили в силу.

```
systemctl restart postgresql
```

Рисунок 24

9. Установка среды выполнения ядра ASP.NET 8 в Debian 12

Для входа под учетной записью суперпользователя используем команду и вводим пароль.

```
su -
```

Рисунок 25

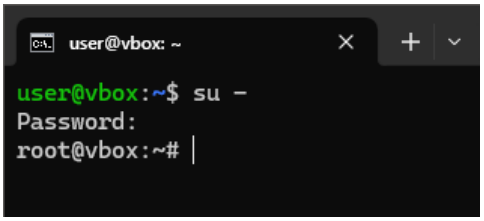

Рисунок 26

Для установки версии с помощью apt потребуется выполнить несколько команд: добавить ключ подписи пакета Microsoft в список доверенных ключей и добавить репозиторий пакетов.

```
wget
https://packages.microsoft.com/config/debian/12/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb

dpkg -i packages-microsoft-prod.deb

rm packages-microsoft-prod.deb
```

Рисунок 27

Выполним установку среды выполнения ядра ASP.NET 8, используя следующую команду.

```
apt update && apt install -y aspnetcore-runtime-8.0
```

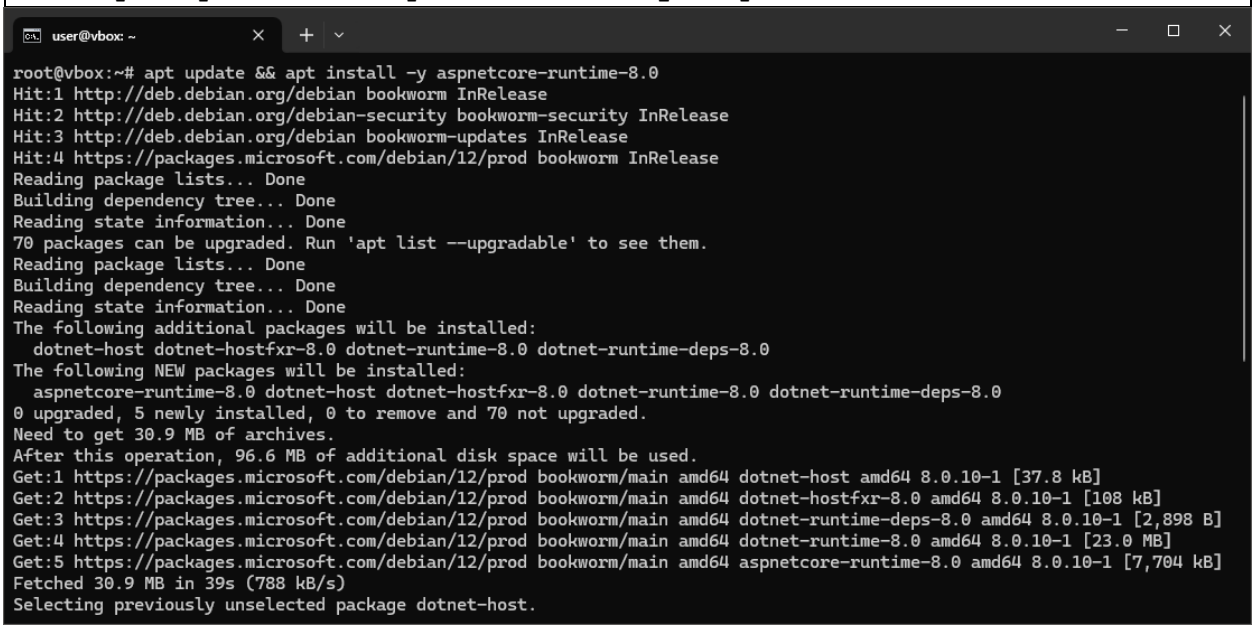

Рисунок 28

Чтобы убедиться, что ASP.NET 8 установлен, выполним следующую команду, которая отобразит информацию об установленном пакете.

```
dotnet --info
```

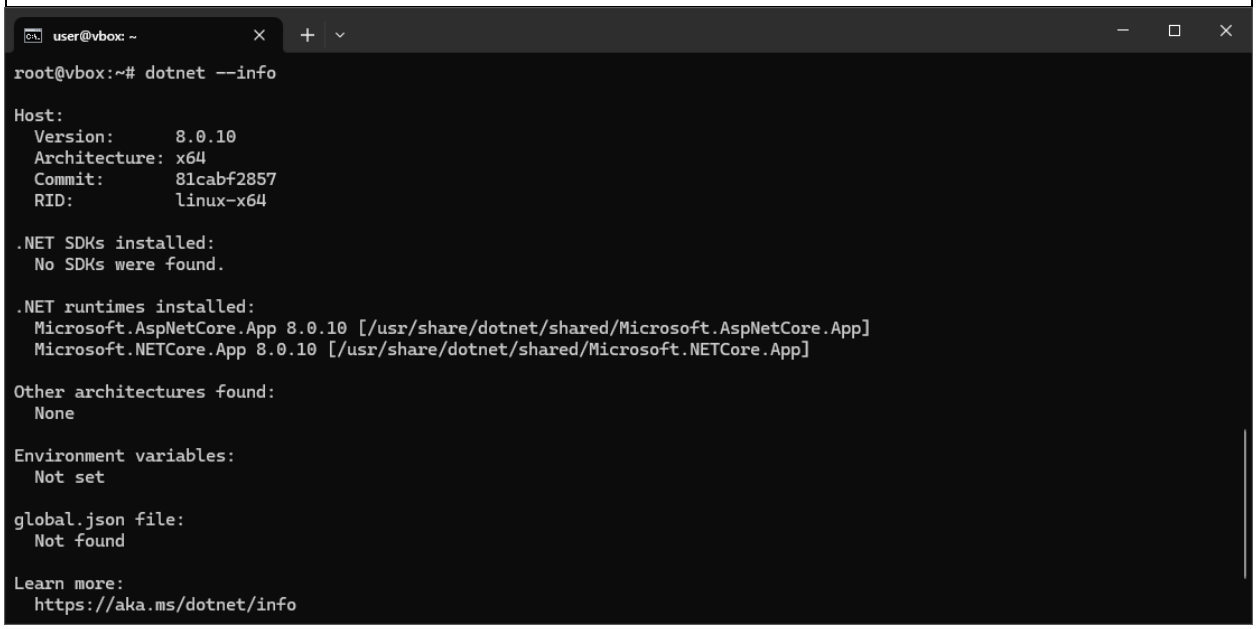

Рисунок 29

ASP.NET 8 установлен.

10. Установка web-сервера Nginx в Debian 12

Для входа под учетной записью суперпользователя используем команду и вводим пароль.

```
su
```

Рисунок 30

Скачиваем и добавляем ключ Nginx Inc. в нашу систему.

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

Рисунок 31

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
```

Рисунок 32

Обновляем список пакетов.

```
apt update
```

Рисунок 33

Устанавливаем Nginx и OpenSSL.

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

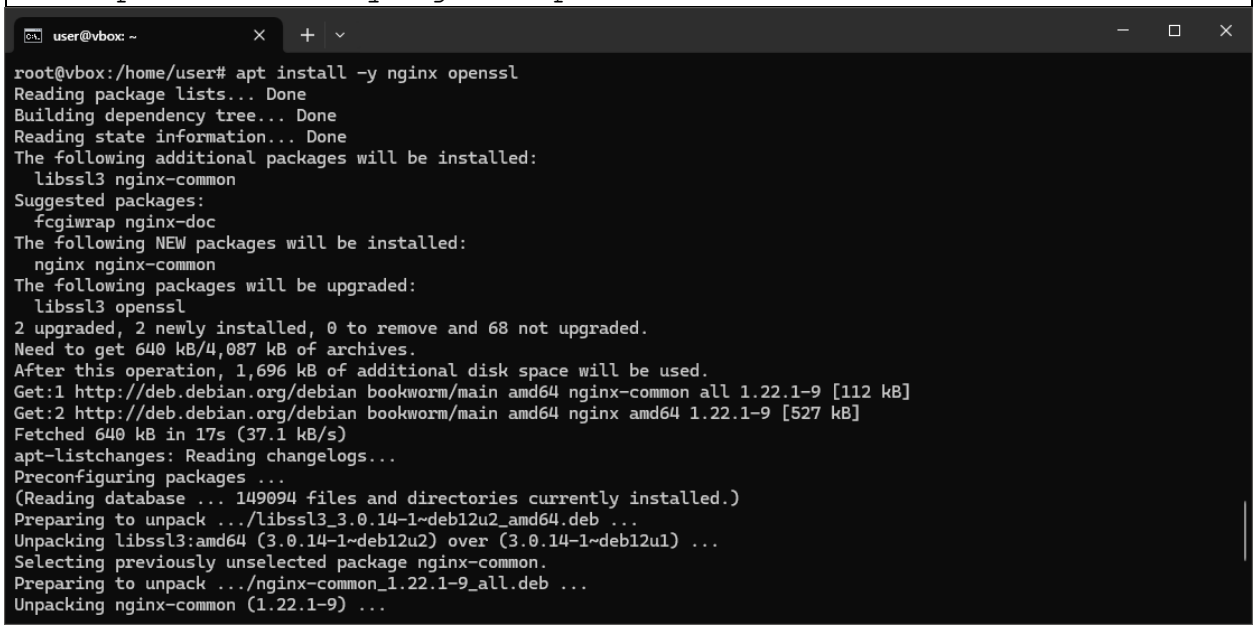

Рисунок 34

После установки запускаем Nginx.

```
systemctl start nginx
```

Рисунок 35

Проверим факт запуска Nginx.

```
systemctl status nginx
```

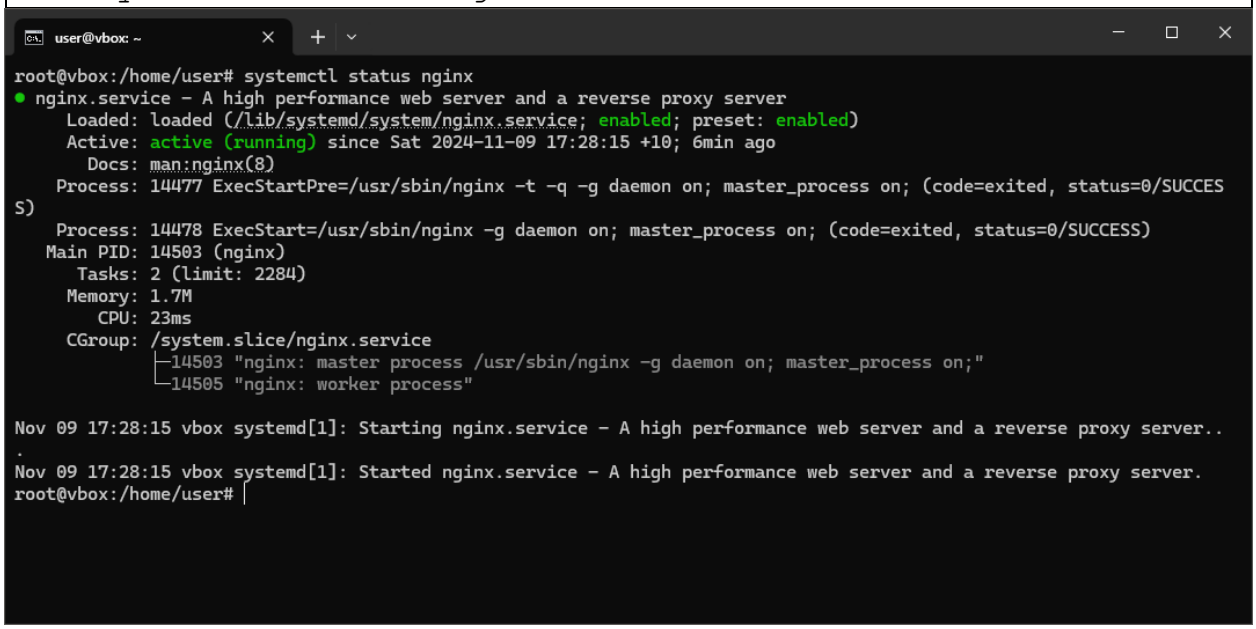

Рисунок 36

Nginx запущен.

Проверим открытые порты.

```
ss -ltupn | grep nginx
```

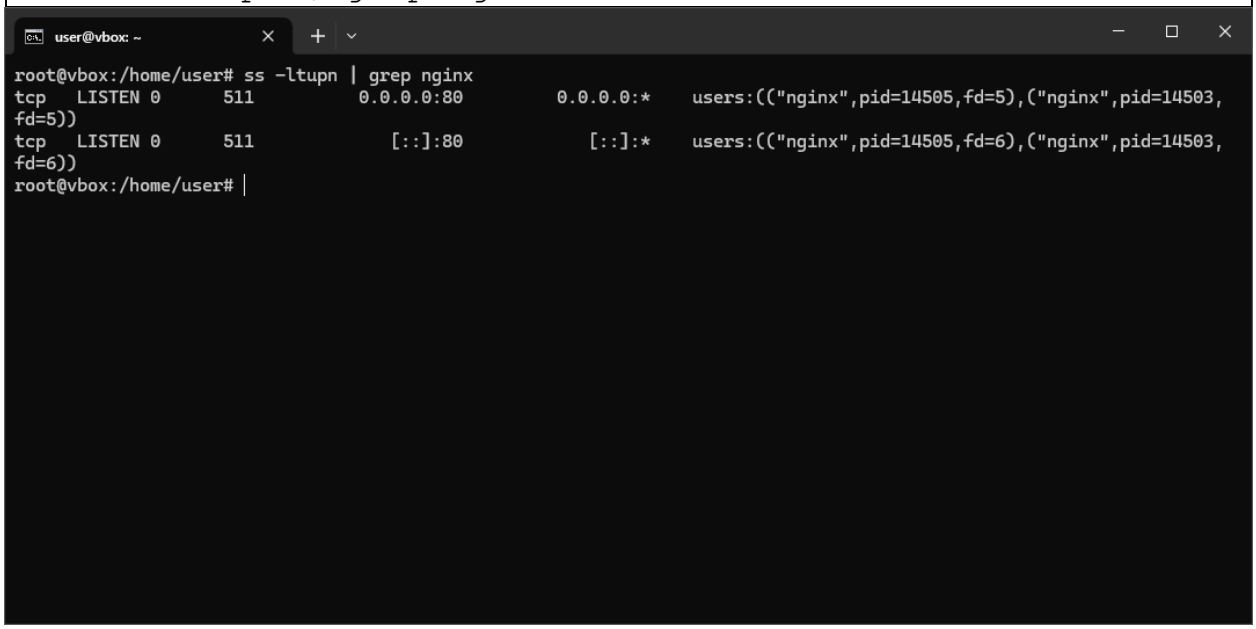

Рисунок 37

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

11. Базовая настройка web-сервера Nginx в Debian 12

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
```

Рисунок 38

Скачаем готовый файл настроек приложения Nginx.

```
wget
https://gist.githubusercontent.com/CHERTS/8e9ecf4fbfb76555
6311a88e5106174b/raw/nginx.conf -O /etc/nginx/nginx.conf
```

Рисунок 39

Создадим файл с конфигурацией для подключения к нашему приложению.

```
echo 'server {
    listen 5050;
    server_name _;

    location / {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}' >> /etc/nginx/conf.d/LightingDispatcher.conf
```

Рисунок 40

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

После этого проверяем конфигурацию Nginx.

```
nginx -t
```

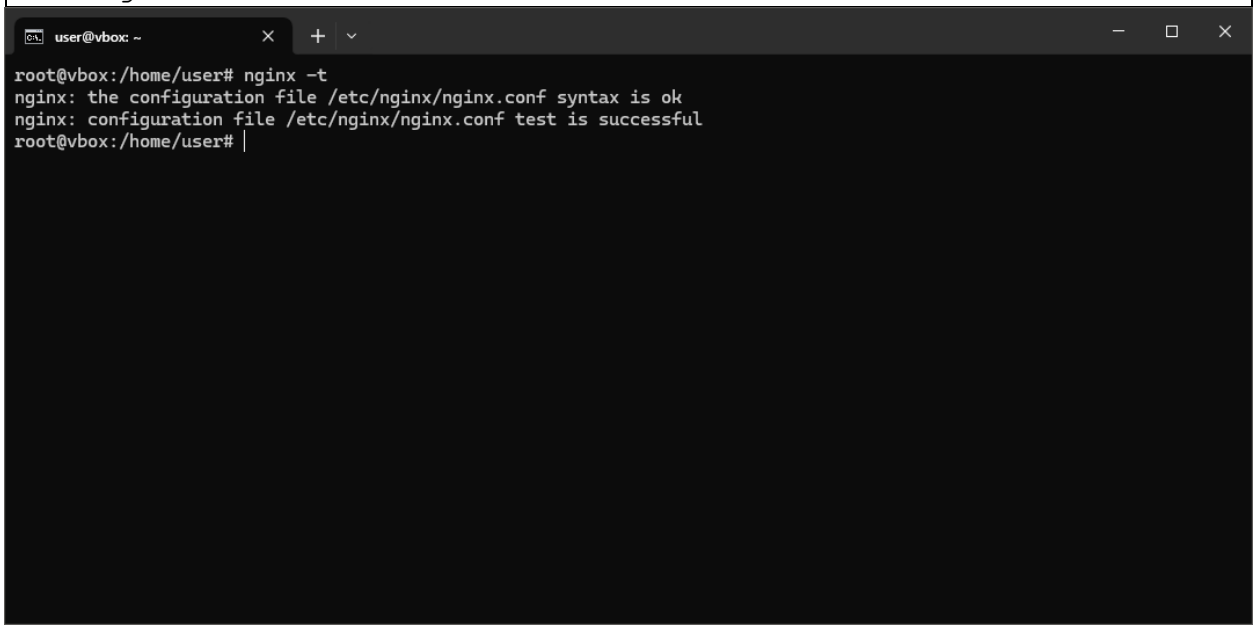

Рисунок 41

Ошибок нет, теперь перезагрузим Nginx.

```
nginx -s reload
```

Рисунок 42

12. Настройка и запуск серверного приложения «LIGHTING»

Создадим пользователя dispatcher, под которым будем запускать приложение.

```
sudo adduser dispatcher
```

Рисунок 43

Переключимся на пользователя dispatcher.

```
su dispatcher
```

Рисунок 44

Создадим каталог для приложения.

```
mkdir /home/dispatcher/Server
```

Рисунок 45

Переносим файлы из полученного вами архива в каталог /home/dispatcher/Server.

Замените содержимое файла secrets.json ({postger_ip} и {password}) на ваши установленные значения.

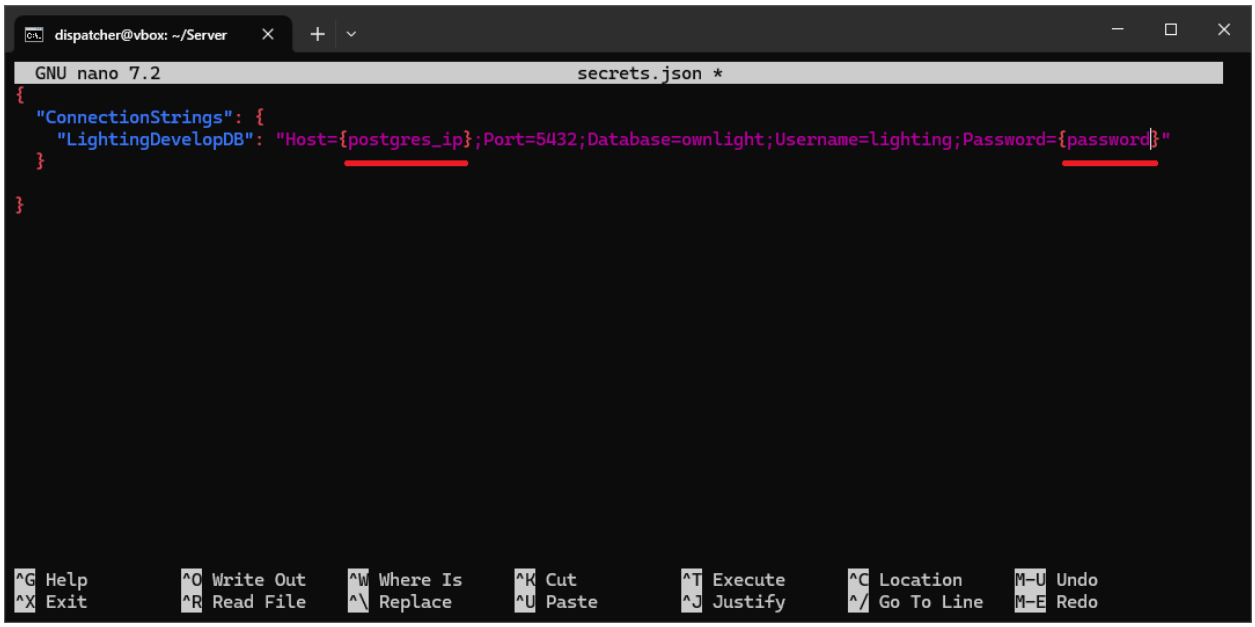

Рисунок 46

Запустим приложение.

```
cd /home/dispatcher/Server

dotnet LightingDispatcher.Server.dll
```

Рисунок 47

Теперь по адресу http://IP-адрес-вашего-сервера:5050 Вы сможете увидеть запущенное приложение.