

Инструкция по локальному развертыванию Netbox

Требования:

Зависимость	Мин.версия
Python	3.8
PostgreSQL	12
Redis	4.0

Настройка БД PostgreSQL

1. Установка
2.

```
sudo apt update  
sudo apt install -y postgresql
```
3. Проверить, что версия PostgreSQL 12 и выше

```
psql -V
```
4. Войти в CLI postgres

```
sudo -u postgres psql
```
5. Необходимо создать пользователя для Netbox и базу данных, выполнив следующую команду:
6.

```
CREATE DATABASE netbox;
```
7.

```
CREATE USER netbox WITH PASSWORD 'J5brHrAXFLQsif0K';
```
8.

```
ALTER DATABASE netbox OWNER TO netbox;
```
9. -- следующие 2 команды нужны для PostgreSQL 15 и выше
10.

```
\connect netbox;  
GRANT CREATE ON SCHEMA public TO netbox;
```

*Пароль указан в качестве примера. Важно использовать свой сложный пароль.

Установка Redis

1. Установить командой

```
sudo apt install -y redis-server
```
2. Проверить версию командой (должна быть не ниже 4.0):

```
redis-server -v
```

Возможно, вы захотите изменить конфигурацию Redis в `/etc/redis.conf` или `/etc/redis/redis.conf`, однако в большинстве случаев достаточно конфигурации по умолчанию.

Установка Netbox

1. Установить необходимые пакеты:

```
sudo apt install -y python3 python3-pip python3-venv python3-dev build-essential libxml2-dev libxslt1-dev libffi-dev libpq-dev libssl-dev zlib1g-dev
```

2. Создать директорию:

```
3. sudo mkdir -p /opt/netbox/  
   cd /opt/netbox/
```

4. Распаковать содержимое архива `netbox-3.7.1.7z` в директорию `/opt/netbox/`

5. Создать пользователя для Netbox:

```
6. sudo adduser --system --group netbox  
7. sudo chown --recursive netbox /opt/netbox/netbox/media/  
8. sudo chown --recursive netbox /opt/netbox/netbox/reports/  
   sudo chown --recursive netbox /opt/netbox/netbox/scripts/
```

9. Скопировать файлы из директории "configuration" в `/opt/netbox/netbox/netbox/`

10. Сгенерировать `secret_key`

```
python3 /opt/netbox/netbox/generate_secret_key.py
```

11. Конфигурация NetboxPRO. Скопировать файл конфигурации следующей командой:

```
12. cd /opt/netbox/netbox/netbox/  
    sudo cp configuration_example.py configuration.py
```

13. Базовые настройки для работы в файле `configuration.py`:

1. Указать `ALLOWED_HOSTS`. Это список допустимых имен хостов и IP-адресов, с помощью которых можно связаться с этим сервером. Вы должны указать хотя бы одно имя или IP-адрес.
2. `ALLOWED_HOSTS = ['netbox.example.com', '192.0.2.123']`
3. # Если ещё не известно какие домены или адреса будут использоваться то указать все адреса
`ALLOWED_HOSTS = ['*']`
4. Указать параметр `DATABASE` для подключения к БД. Этот параметр содержит сведения о конфигурации базы данных. Вы должны указать имя пользователя и пароль, которые использовались при настройке PostgreSQL.

```

5. DATABASE = {
6.     'NAME': 'netbox',           # Database name
7.     'USER': 'netbox',         # PostgreSQL username
8.     'PASSWORD': 'J5brHrAXFLQSif0K', # PostgreSQL password
9.     'HOST': 'localhost',     # Database server
10.    'PORT': '',               # Database port (leave
    blank for default)
11.    'CONN_MAX_AGE': 300,      # Max database connection
    age (seconds)
    }

```

12. **SECRET_KEY**. Этому параметру должен быть присвоен случайно сгенерированный ключ, используемый в качестве соли для хэширования и связанных с ним криптографических функций. (Обратите внимание, однако, что он никогда напрямую не используется при шифровании секретных данных.) Этот ключ должен быть уникальным для данной установки, и рекомендуется, чтобы его длина составляла не менее 50 символов. Он не должен использоваться совместно за пределами локальной системы.

14. Запустить установку Netbox

```

15. sudo chmod +x /opt/netbox/upgrade.sh
    sudo /opt/netbox/upgrade.sh

```

16. Активировать виртуальное окружение

```
source /opt/netbox/venv/bin/activate
```

17. Создать superuser

```

18. cd /opt/netbox/netbox
    python3 manage.py createsuperuser

```

19. Протестировать работоспособность

```
python3 manage.py runserver 0.0.0.0:8000 --insecure
```

Чтобы выйти из development сервера нажать ctrl+C

Настройка Gunicorn

Для запуска WSGI приложений требуется Gunicorn

1. Скопировать файл gunicorn.py

```
sudo cp /opt/netbox/contrib/gunicorn.py /opt/netbox/gunicorn.py
```

2. Мы будем использовать демон для контроля за процессами gunicorn и Netbox

```

3. sudo cp -v /opt/netbox/contrib/*.service /etc/systemd/system/
    sudo systemctl daemon-reload

```

4. Запустим сервисы netbox и netbox-rq

```
5. sudo systemctl start netbox netbox-rq
sudo systemctl enable netbox netbox-rq
```

6. Проверим статус

```
systemctl status netbox.service
```

Вывод должен быть такой:

```
• netbox.service - NetBox WSGI Service
  Loaded: loaded (/etc/systemd/system/netbox.service; enabled;
vendor preset: enabled)
  Active: active (running) since Mon 2021-08-30 04:02:36 UTC; 14h
ago
    Docs: https://docs.netbox.dev/
   Main PID: 1140492 (gunicorn)
     Tasks: 19 (limit: 4683)
    Memory: 666.2M
    CGroup: /system.slice/netbox.service
            └─1140492 /opt/netbox/venv/bin/python3
/opt/netbox/venv/bin/gunicorn --pid /va>
            └─1140513 /opt/netbox/venv/bin/python3
/opt/netbox/venv/bin/gunicorn --pid /va>
            └─1140514 /opt/netbox/venv/bin/python3
/opt/netbox/venv/bin/gunicorn --pid /va>
    ...
```

Настройка nginx

Мы будем использовать nginx для обслуживания static-файлов.

Отредактировать файл */opt/netbox/contrib/nginx.conf* под нужную конфигурацию.

1. Установим nginx

```
sudo apt install -y nginx
```

2. Скопируем nginx.conf

```
sudo cp /opt/netbox/contrib/nginx.conf /etc/nginx/sites-
available/netbox
```

3. Удаляем дефолтный конфиг-файл и создаем линк на наш файл конфига

```
4. sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/netbox /etc/nginx/sites-
enabled/netbox
```

5. Перезапускаем nginx, чтобы применить нашу новую конфигурацию.

```
sudo systemctl restart nginx
```