

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «ФРЕШТЕХ»
(ООО «ФРЕШТЕХ»)

Программный продукт «Нейронная система «ВиДок»

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

Содержание

1 Общие положения	3
1.1 Предназначение документа	3
1.2 Краткие сведения о программном продукте «Нейронная система «ВиДок»	3
1.3 Состав компонентов	3
1.4 Требования к инфраструктуре.....	4
2 Подготовка к установке	5
2.1 Требования к серверу	5
2.2 Требования к программному обеспечению	5
2.3 Получение комплекта поставки	5
3 Порядок установки	6
3.1 Подготовка окружения.....	6
3.2 Развёртывание программного продукта «Нейронная система «ВиДок»	6
3.3 Создание учётной записи администратора	7
3.4 Проверка работоспособности.....	7
4 Настройка окружения.....	8
4.1 Файлы конфигурации.....	8
4.2 Настройка базы данных	8
4.3 Настройка брокера сообщений.....	8
5 Проверка установки	9
5.1 Проверка доступности API.....	9
5.2 Проверка аутентификации.....	9
Перечень принятых сокращений.....	10

1 Общие положения

1.1 Предназначение документа

Настоящий документ является инструкцией по установке программного продукта «Нейронная система «ВиДок» (далее – программный продукт «ВиДок»). Документ предназначен для системных администраторов и специалистов по развёртыванию, выполняющих установку и первичную настройку программного продукта «ВиДок».

Инструкция описывает последовательность действий, необходимых для развёртывания программного продукта «ВиДок» в целевой среде, проверки его работоспособности, а также начальной настройки компонентов системы.

1.2 Краткие сведения о программном продукте «Нейронная система «ВиДок»

Программный продукт «Нейронная система «ВиДок» предназначен для автоматизированного приёма, обработки и валидации документов с использованием нейросетевых алгоритмов.

Программный продукт «ВиДок» разработано ООО «Фрештех» (ИНН 7719835308, КПП 771901001, ОГРН 1137746082187, info@frte.ru) на базе следующих технологий: Python 3.12, Django 5.0, Django REST Framework, PostgreSQL 12+, Redis 6+, Celery 5.4, PyTorch 2.4, YOLOv5.

Развёртывание программного продукта «ВиДок» осуществляется в контейнерной среде Docker с использованием Docker Compose, что обеспечивает воспроизводимость конфигурации и упрощает управление компонентами системы.

1.3 Состав компонентов

В состав программного продукта «Нейронная система «ВиДок» входят компоненты, приведённые в таблице 1.

Таблица 1 – Компоненты программного продукта «Нейронная система «ВиДок»

Компонент	Назначение	Имя контейнера (Docker)
API-сервис	Веб-приложение Django, реализующее REST API для приёма заявок и управления системой	ai_documents-back-1
PostgreSQL	Реляционная СУБД для хранения данных заявок, пользователей и результатов обработки	ai_documents-db-1
Redis	Брокер сообщений и сервер кэширования	ai_documents-redis-1
Celery Worker	Служба фоновых задач для асинхронной обработки документов нейросетевыми алгоритмами	ai_documents-celery-1

Celery Beat	Планировщик периодических задач (очистка устаревших файлов, уведомления)	ai_documents-beat-1
-------------	--	---------------------

1.4 Требования к инфраструктуре

Для успешного развёртывания программного продукта «Видок» целевая среда должна удовлетворять требованиям, изложенным в разделе 2 настоящей инструкции. Сетевая конфигурация должна обеспечивать доступность сервера по протоколу HTTPS на портах 80 и 443. Рекомендуется использование обратного прокси-сервера (например, Nginx) для терминирования TLS-соединений.

2 Подготовка к установке

2.1 Требования к серверу

Для развёртывания программного продукта «Нейронная система «ВиДок» необходим сервер со следующими минимальными характеристиками:

- процессор (CPU): 4 ядра, тактовая частота не менее 2,5 ГГц;
- оперативная память (RAM): не менее 16 ГБ;
- дисковое пространство: SSD объёмом не менее 100 ГБ (под ОС, Docker-образы, данные PostgreSQL и загружаемые файлы документов);
- сетевой интерфейс: не менее 100 Мбит/с;
- Docker 24.0 и выше;
- Docker Compose 2.20 и выше.

Наличие графического процессора (GPU) с поддержкой CUDA ускоряет работу нейросетевого модуля и является рекомендуемым, но не обязательным требованием.

2.2 Требования к программному обеспечению

На сервере должно быть установлено следующее программное обеспечение:

- операционная система: Debian 11/12 или Ubuntu 20.04/22.04 LTS (64-разрядная);
- Docker Engine 24.0 и выше;
- Docker Compose Plugin 2.20 и выше (входит в состав Docker Desktop и Docker Engine).

Для управления сервером необходимо наличие SSH-доступа с правами суперпользователя (sudo) или непосредственного входа в систему.

2.3 Получение комплекта поставки

Комплект поставки программного продукта «Нейронная система «ВиДок» предоставляется ООО «Фрештех» в соответствии с заключённым договором и включает:

- архив с исходным кодом программного продукта «ВиДок» или ссылку на защищённый репозиторий;
- файлы Docker Compose (docker-compose.yml, docker-compose.local.yml);
- шаблон файла переменных окружения (.env.example);
- настоящую инструкцию по установке.

Для получения комплекта поставки необходимо обратиться к разработчику по адресу электронной почты info@frte.ru.

3 Порядок установки

3.1 Подготовка окружения

Перед развёртыванием программного продукта «Нейронная система «ВиДок» необходимо установить Docker Engine и Docker Compose Plugin на целевом сервере. Для ОС Debian/Ubuntu выполните следующие команды в терминале:

```
# Обновление списка пакетов
```

```
sudo apt-get update
```

```
# Установка зависимостей
```

```
sudo apt-get install -y ca-certificates curl gnupg
```

```
# Добавление официального репозитория Docker
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --  
dearmor \
```

```
  -o /etc/apt/keyrings/docker.gpg
```

```
# Установка Docker Engine и Docker Compose Plugin
```

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-  
compose-plugin
```

После установки убедитесь в корректной работе Docker, выполнив команду:

```
docker --version
```

Перенесите полученный комплект поставки на сервер и перейдите в директорию с файлами программного продукта «Видок». Создайте файл переменных окружения на основе шаблона:

```
cp .env.example .env
```

Отредактируйте файл .env в соответствии с параметрами целевого окружения (см. раздел 4).

3.2 Развёртывание программного продукта «Нейронная система «ВиДок»

Для сборки Docker-образов и запуска всех компонентов программного продукта «Видок» выполните следующую команду из директории с файлами комплекта поставки:

```
docker-compose -f docker-compose.local.yml up --build -d
```

Ключ --build обеспечивает пересборку образов при первом запуске или при обновлении исходного кода. Ключ -d запускает контейнеры в фоновом режиме (detached mode).

Процесс сборки и запуска занимает от 5 до 15 минут в зависимости от скорости интернет-соединения и производительности сервера.

Для применения миграций базы данных выполните:

```
docker exec -it ai_documents-back-1 python manage.py migrate
```

3.3 Создание учётной записи администратора

После успешного запуска контейнеров необходимо создать учётную запись администратора для доступа к административному интерфейсу системы. Выполните следующие команды:

```
# Подключение к контейнеру API-сервиса
docker exec -it ai_documents-back-1 bash
```

```
# Создание учётной записи суперпользователя
python manage.py createsuperuser
```

Команда `createsuperuser` запустит интерактивный мастер, в котором необходимо последовательно ввести имя пользователя, адрес электронной почты и пароль. Минимальная длина пароля — 8 символов.

Для выхода из контейнера введите:

```
exit
```

3.4 Проверка работоспособности

После завершения установки убедитесь в том, что все контейнеры запущены и работают корректно, выполнив команду:

```
docker-compose -f docker-compose.local.yml ps
```

В выводе команды все контейнеры должны иметь статус `Up`. Пример ожидаемого вывода:

NAME	IMAGE	STATUS	PORTS
ai_documents-back-1	back	Up	0.0.0.0:8000->8000/tcp
ai_documents-db-1	postgres	Up	5432/tcp
ai_documents-redis-1	redis	Up	6379/tcp
ai_documents-celery-1	back	Up	
ai_documents-beat-1	back	Up	

4 Настройка окружения

4.1 Файлы конфигурации

Конфигурация программного продукта «Видок» осуществляется через файл переменных окружения `.env`, расположенный в корневой директории комплекта поставки. Файл содержит следующие основные параметры:

Параметр	Описание	Пример значения
SECRET_KEY	Секретный ключ Django (обязателен для подписи сессий и токенов)	<code>your-secret-key-here</code>
DEBUG	Режим отладки (False для производственной среды)	<code>False</code>
ALLOWED_HOSTS	Разрешённые хосты (IP или доменное имя сервера)	<code>192.168.1.100</code>
DB_NAME	Наименование базы данных PostgreSQL	<code>vidok_db</code>
DB_USER	Пользователь базы данных	<code>vidok_user</code>
DB_PASSWORD	Пароль пользователя базы данных	<code>strong_password</code>
DB_HOST	Хост СУБД (имя Docker-сервиса)	<code>db</code>
DB_PORT	Порт СУБД	<code>5432</code>
REDIS_URL	URL подключения к Redis	<code>redis://redis:6379/0</code>

4.2 Настройка базы данных

Параметры подключения к базе данных PostgreSQL задаются в файле `.env` (параметры `DB_NAME`, `DB_USER`, `DB_PASSWORD`, `DB_HOST`, `DB_PORT`). При первом запуске Docker Compose автоматически создаёт базу данных и пользователя с указанными учётными данными.

Для применения миграций Django к базе данных выполните команду:

```
docker exec -it ai_documents-back-1 python manage.py migrate
```

Для создания резервной копии базы данных используйте команду:

```
docker exec ai_documents-db-1 pg_dump -U vidok_user vidok_db > backup.sql
```

4.3 Настройка брокера сообщений

Для подключения Celery к брокеру сообщений Redis задайте параметр `REDIS_URL` в файле `.env`. По умолчанию Redis запускается в том же Docker Compose окружении и доступен по имени Docker-сервиса `redis`:

```
REDIS_URL=redis://redis:6379/0
```

Для работы Celery Beat (планировщика периодических задач) дополнительная настройка не требуется — планировщик использует тот же параметр `REDIS_URL`.

5 Проверка установки

5.1 Проверка доступности API

После успешного развёртывания API-сервис должен быть доступен по адресу сервера на порту 8000 (или на стандартном порту 80/443 при использовании обратного прокси-сервера).

Для проверки доступности Swagger-документации откройте в браузере:

`http://<адрес_сервера>:8000/api/v1/swagger/`

Должна отобразиться интерактивная страница документации Swagger UI со списком доступных эндпоинтов API. Если страница не открывается, проверьте статус контейнера `ai_documents-back-1` командой `docker logs ai_documents-back-1`.

Для проверки доступности административного интерфейса Django откройте:

http://<адрес_сервера>:8000/admin/

5.2 Проверка аутентификации

Для проверки работоспособности механизма JWT-аутентификации выполните запрос к эндпоинту получения токена, используя учётные данные созданного администратора:

```
curl -X POST http://<адрес_сервера>:8000/api/auth/jwt/create/ \
  -H "Content-Type: application/json" \
  -d '{"username": "admin", "password": "your_password"}'
```

В случае успешной аутентификации сервер вернёт ответ в формате JSON, содержащий `access` и `refresh` токены:

```
{"access": "eyJ...", "refresh": "eyJ..."}
```

Получение токена подтверждает корректность установки ПП «Видок» и работоспособность основных компонентов системы: API-сервиса, базы данных PostgreSQL и брокера Redis.

Перечень принятых сокращений

Сокращение	Расшифровка
API	Application Programming Interface – программный интерфейс приложения
CPU	Central Processing Unit – центральный процессор
Docker	Платформа контейнеризации приложений
HTTPS	HyperText Transfer Protocol Secure – защищённый протокол передачи гипертекста
ОС	Операционная система
Программный продукт «Видок»	Программный продукт «Нейронная система «ВиДок»
RAM	Random Access Memory – оперативная память
REST	Representational State Transfer – архитектурный стиль взаимодействия
SSD	Solid-State Drive – твердотельный накопитель
SSH	Secure Shell – протокол защищённого удалённого доступа
СУБД	Система управления базами данных