****

**ОПИСАНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ РАЗВЕРТЫВАНИЯ ПРОЕКТА ИЗ ИСХОДНОГО КОДА**

**ПРОГРАММНЫЙ КОМПЛЕКС ПРЕДУПРЕЖДЕНИЯ ВОДИТЕЛЕЙ О ПОЯВЛЕНИИ ДИКИХ ЖИВОТНЫХ НА ПРОЕЗЖЕЙ ЧАСТИ "ИСОВ-Л"**

**2024**

Для развёртывания проекта используется контейнеризация с помощью **Docker**. Этот процесс автоматизирует сборку и настройку всех зависимостей, обеспечивая удобство работы в различных средах и гарантируя идентичность окружения на всех стадиях разработки и эксплуатации.

**Требования:**

1. **Docker**: Для работы с проектом необходимо иметь установленный Docker на локальной машине. Docker обеспечивает создание изолированных контейнеров, в которых разворачивается весь стек приложения. Убедитесь, что на машине установлена последняя версия Docker, следуя инструкциям на официальном сайте [Docker](https://docs.docker.com/engine/install/).
2. **Docker Compose**: Для оркестрации контейнеров используется Docker Compose, который автоматически настраивает и запускает все необходимые контейнеры, согласно описанию в файле docker-compose.yml.

**Сборка проекта:**

1. **Клонирование репозитория**:

Для начала нужно клонировать репозиторий проекта с помощью команды:

git clone \*\*\*\*\*\*\*\*vis.systems:vis7715827/iron-alpha.git

1. **Настройка переменных окружения**:

cd iron\_alpha/

cp .env.example .env

Задать значения переменным окружения в соответствии с конфигурацией комплекса (см. раздел [Описание переменных конфигурации](#udwh80p16o8s)).

Важно обратить внимание на настройку ключа DB\_HOST, который должен быть установлен в значение db (название контейнера с БД):

DB\_HOST=db

1. **Запуск сборки проекта**:

Сборка и запуск всех контейнеров, необходимых для работы проекта, выполняются с помощью команды:

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

Эта команда создаёт и запускает все сервисы, указанные в файле docker-compose.yml, а также выполняет сборку образов, если они еще не существуют.

**Результат:**

После успешного выполнения этих шагов веб-интерфейс проекта будет доступен по URL-адресу, который указывается в конфигурации (переменная **URL**). Для доступа к автодокументации API используется Swagger, который будет доступен по адресу:

* **http://{URL}/apidocs/**

**Описание переменных конфигурации:**

Для корректной работы системы требуется настроить ряд переменных окружения, таких как настройки базы данных, серверные параметры, IP-адреса устройств и параметры безопасности. Каждая из переменных отвечает за определённые аспекты работы системы, включая шифрование, подключение к базе данных, управление лицензиями, доступ к удалённым устройствам и другие параметры.

Вся информация о переменных и их значениях представлена в таблице ниже:

| **Переменная** | **Описание** |
| --- | --- |
| **SECRET\_KEY** | Секретный ключ для шифрования. |
| **DB\_HOST** | Хост базы данных. |
| **DB\_PORT** | Порт базы данных. |
| **DB\_NAME** | Имя базы данных. |
| **DB\_USER** | Пользователь базы данных. |
| **DB\_PASS** | Пароль пользователя базы данных. |
| **LICENSE\_SECRET\_KEY** | Секретный ключ для управления лицензиями. |
| **URL** | URL-адрес основного сервера. |
| **WEB\_SERVER\_PORT** | Порт для работы веб-сервера. |
| **CONTROLLER\_IP** | IP-адрес контроллера. |
| **CONTROLLER\_PORT** | Порт контроллера. |
| **DEVICE\_IP** | IP-адрес устройства. |
| **DEVICE\_PORT** | Порт устройства. |
| **DEVICE\_USER** | Имя пользователя устройства. |
| **DEVICE\_PASS** | Пароль пользователя устройства. |
| **SEC\_DEVICE\_IP** | IP-адрес дополнительного устройства безопасности. |
| **SEC\_DEVICE\_PORT** | Порт дополнительного устройства безопасности. |
| **SEC\_DEVICE\_USER** | Имя пользователя дополнительного устройства безопасности. |
| **SEC\_DEVICE\_PASS** | Пароль пользователя дополнительного устройства безопасности. |
| **AWS\_URL** | URL для доступа к хранилищу на облачном сервере. |

Этот процесс позволяет автоматизировать настройку среды, сборку проекта и его запуск, обеспечивая высокую степень повторяемости и совместимости во всех этапах разработки и эксплуатации.