

Инструкция по установке экземпляра программы «Defbox»

Оглавление

Инструкция по установке экземпляра ПО	3
Характеристики сервиса	3
Требования	3
Описание процесса настройки	3
Подготовка Проекта	3
Настройка переменных окружения	3
Необходимо указать значения:	4
Сборка и запуск контейнеров	4
Первоначальная настройка проекта	4
Проверка работоспособности	5
Настройка Nginx	5
Проксирование запросов:	5
Дополнительные настройки	5

Инструкция по установке экземпляра ПО

Характеристики сервиса

Серверная часть написана на языке Python 3, с использованием фреймворка Django. Клиентская часть написана на языке TypeScript с использованием программной платформы с открытым исходным кодом [React.js](#) Поддерживает WebSocket соединения для функционала работы с терминалом. Исходные коды сервиса оптимизированы для работы с СУБД PostgreSQL.

Веб-версия программы распространяется в виде интернет-сервиса, специальные действия по установке на стороне пользователей не требуются.

Для получения доступа к сервису не требуется специальное оборудование. Пользователь может воспользоваться любым персональным компьютером, смартфоном или планшетом с доступом к сети интернет.

На устройстве должен быть установлен любой из веб-браузеров:

- Chrome и браузеры, построенные на его базе (например, Yandex.Browser) 78+;
- Microsoft Edge 44+;
- Mozilla FireFox 70+;
- Apple Safari 12+.

Отображение информации возможно только при наличии соединения с интернет по каналу связи wi-fi или через сеть передачи данных провайдера мобильной связи.

Требования

Операционная система

- ОС Linux (рекомендуется Ubuntu 20.04+ / Debian 11+)
- Поддержка Docker

Основное программное обеспечение

- Docker 20.10 и выше
- Docker Compose 1.29 и выше (или встроенный `docker compose`)

Контейнеры и сервисы (разворачиваются автоматически)

- PostgreSQL 12 и выше
- Redis 5.0 и выше
- Nginx (последняя стабильная версия)
- Backend-приложение (Python / Django + Gunicorn)
- WebSocket-сервер (Django Channels + Daphne)
- Celery (workers и планировщик задач)

Дополнительно

- Доступ к сети Интернет (для сборки контейнеров)
- SSL-сертификат для работы по HTTPS (возможно использование самоподписанного)

Описание процесса настройки

Подготовка Проекта

Для дальнейших шагов нужно распаковать архив с исходными кодами сервиса:

```
unzip project.zip
cd project
```

Настройка переменных окружения

Для работы проекта нужно создать файл с переменными окружения `.env`

```
SECRET_KEY=your_secret_key

DB_URL=postgresql://tg:password@postgres:5432/tg
DB_PASS=your_db_password

CELERY_URL=redis://redis:6379/0

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=

STRIPE_API_KEY=
STRIPE_WEBHOOK_SIGNING_SECRET=

MAILCHIMP_API_KEY=
MAILCHIMP_ENABLED=

PROXMOX_ENDPOINT=
PROXMOX_USER=
PROXMOX_PASSWORD=
```

Необходимо указать значения:

DB_URL — строка подключения к PostgreSQL
DB_PASS — пароль базы данных
SECRET_KEY — секретный ключ приложения
CELERY_URL — адрес Redis
STRIPE_API_KEY — ключ платежной системы

Сборка и запуск контейнеров

Для запуска контейнеров нужно команду:

```
docker compose up --build -d
```

Будут автоматически запущены следующие сервисы:

- backend (Gunicorn)
- web_socket (Daphne)
- nginx
- postgres
- redis
- celery workers
- celery beat
- migrations

Первоначальная настройка проекта

Сервис `migrations` выполняется автоматически и:

- применяет миграции базы данных
- собирает статические файлы

При необходимости можно выполнить миграции вручную:

```
docker compose run migrations
```

Проверка работоспособности

После запуска сервисов:

- Приложение доступно по адресу:
`http://localhost` или `https://your-domain`
- Проверка backend:

```
curl http://localhost/healthcheck
```

Настройка Nginx

Nginx запускается в контейнере и использует внешний конфигурационный файл:

```
/opt/nginx.conf
```

Также используются директории:

- `/opt/static/`
- `/opt/static-dev/`
- `/opt/static-v2/`

Проксирование запросов:

- HTTP → backend (Gunicorn)
- WebSocket → web_socket (Daphne)

Пример настройки WebSocket:

```
location /ws/ {  
    proxy_pass http://web_socket:8000;  
    proxy_http_version 1.1  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade"
```

Дополнительные настройки

Используются Docker volumes:

- **postgresdata** — база данных
- **static-volume** — статические файлы