

DK-Mark

Инструкция по установке

Система маркировки продукции

1. Состав поставки

В архиве dk-mark-deploy.zip находятся следующие файлы:

Файл / папка	Назначение
docker-compose.yml	Описание всех сервисов системы
.env	Все настройки и пароли — единственный файл для редактирования
config.template.yaml	Шаблон конфига бэкенда (не редактировать)
install.sh / install.bat	Первичная установка (Linux / Windows); проверяет Docker и предлагает установить WireGuard
update.sh / update.bat	Обновление системы (Linux / Windows)
agent/config.yaml	Конфигурация агента
agent/install-linux.sh	Установка агента на Linux
agent/install-windows.bat	Установка агента на Windows
agent/update-linux.sh	Обновление агента на Linux
agent/update-windows.bat	Обновление агента на Windows
agent/dk-mark-agent.service	Systemd-сервис для Linux

i config.template.yaml редактировать не нужно — все параметры задаются через .env

2. Требования

2.1 Сервер (бэкенд + фронтенд)

- ОС: Linux (Ubuntu 20.04 / 22.04 / Debian 11+)
- Docker 24+ и Docker Compose v2+ (будут установлены автоматически скриптом install.sh / install.bat, если отсутствуют)
- Открытые порты: 80 (фронт), 8080 (API), 5672 / 5671 (RabbitMQ)
- Минимум 4 ГБ RAM, 20 ГБ дискового пространства

- Доступ в интернет для скачивания образов из реестра
- WireGuard (опционально, для VPN-подключения) — установщик предложит установку, если пакет не найден

2.2 Машина с агентом

- ОС: Windows 10/11 или Linux
- Docker Desktop (Windows) или Docker Engine (Linux) — только для первичной установки
- Принтер этикеток (USB или TCP/IP), подключённый к машине
- Сетевой доступ до сервера (порты 8080 и 5672)

3. Настройка перед установкой

Откройте файл `.env` и заполните все параметры:

```
# База данных
DB_USER=dkmark
DB_PASSWORD=ВАШ_ПАРОЛЬ_БД

# JWT (случайная строка 32+ символа)
JWT_SECRET=замените-на-случайную-строку

# RabbitMQ
RABBITMQ_USER=admin
RABBITMQ_PASSWORD=ВАШ_ПАРОЛЬ_RABBIT

# MinIO (хранилище файлов)
MINIO_USER=minioadmin
MINIO_PASSWORD=ВАШ_ПАРОЛЬ_MINIO

# IP-адрес сервера (для публичных ссылок на файлы)
SERVER_IP=192.168.1.100

# Адреса Честного знака (оставить как есть для продакшна)
CHZ_BASE_URL=https://markirovka.crptech.ru
CHZ_CATALOG_URL=https://catalog.честныйзнак.рф
SUZ_BASE_URL=https://suz.crptech.ru
SUZ_INTEGRATOR_URL=https://suz-integrator.crptech.ru
```

⚠ Это единственный файл который нужно редактировать. Все остальные файлы трогать не нужно.

⚠ Не используйте значение `changeme` в продакшне. Задайте надёжные пароли длиной не менее 16 символов.

4. Установка сервера

4.1 Linux

1. Распакуйте архив dk-mark-deploy.zip на сервере
2. Перейдите в папку с файлами
3. Отредактируйте файл .env (см. раздел 3)
4. Запустите скрипт установки:

```
chmod +x install.sh
./install.sh
```

i При первом запуске скрипт проверит наличие curl, Docker и Docker Compose v2 — отсутствующие пакеты будут установлены автоматически (Docker — через get.docker.com). Если WireGuard отсутствует, скрипт спросит о его установке. После автоустановки Docker потребуется перелогиниться и запустить ./install.sh повторно.

5. Скрипт запросит логин и пароль от реестра образов harbor.pogonchik.online — введите данные, полученные от поставщика
6. После завершения проверьте статус сервисов:

```
docker compose ps
```

4.2 Windows

7. Распакуйте архив dk-mark-deploy.zip
8. Откройте файл .env в любом текстовом редакторе и заполните параметры
9. Откройте командную строку (cmd) от имени администратора
10. Перейдите в папку с файлами
11. Запустите скрипт установки:

```
install.bat
```

i Скрипт требует winget (встроен в Windows 11 и Windows 10 1809+). При отсутствии curl, Docker Desktop или WireGuard они будут установлены через winget; после установки Docker Desktop необходимо запустить его вручную и повторно запустить install.bat. WireGuard устанавливается по запросу.

12. Введите данные для входа в реестр образов harbor.pogonchik.online

i После успешного запуска веб-интерфейс доступен по адресу http://IP_СЕРВЕРА. При первом входе укажите адрес API: http://IP_СЕРВЕРА:8080

5. Установка агента

Агент устанавливается отдельно на каждой машине оператора, к которой подключён принтер этикеток.

5.1 Настройка конфига агента

Откройте файл agent/config.yaml и заполните обязательные поля:

```
backend:
  url: "http://IP_СЕРВЕРА:8080"
  username: "agent"
  password: "пароль_агента"
  company_id: "... " # ID компании из интерфейса
  production_point_id: "... " # ID места сборки из интерфейса
```

```
rabbitmq:
  host: "IP_СЕРВЕРА"
  port: 5672
  username: "admin"           # значение RABBITMQ_USER из .env
  password: "...             # значение RABBITMQ_PASSWORD из .env
```

△ *company_id* и *production_point_id* берутся из веб-интерфейса DK-Mark в разделе Справочники → Места сборки после первого входа.

5.2 Установка на Linux

13. Перейдите в папку agent/

14. Запустите скрипт:

```
chmod +x install-linux.sh
sudo ./install-linux.sh
```

15. Убедитесь что сервис запущен:

```
systemctl status dk-mark-agent
```

5.3 Установка на Windows

16. Откройте командную строку от имени администратора

17. Перейдите в папку agent/

18. Запустите скрипт:

```
install-windows.bat
```

19. Убедитесь что служба запущена:

```
sc query DKMarkAgent
```

6. Обновление

6.1 Обновление сервера

Linux:

```
./update.sh
```

Windows:

```
update.bat
```

6.2 Обновление агента

Linux:

```
sudo ./agent/update-linux.sh
```

Windows (от имени администратора):

```
agent\update-windows.bat
```

7. Проверка работоспособности

Проверка	Ожидаемый результат
----------	---------------------

<code>http://IP_СЕРВЕРА</code>	Открывается страница входа DK-Mark
<code>http://IP_СЕРВЕРА:8080/health</code>	Ответ: {"status":"ok"}
<code>docker compose ps</code>	Все контейнеры в статусе running
<code>systemctl status dk-mark-agent</code>	active (running)
<code>sc query DKMarkAgent</code>	STATE: 4 RUNNING

8. Диагностика

Просмотр логов сервисов

```
docker compose logs -f backend
docker compose logs -f frontend
docker compose logs -f rabbitmq
```

Просмотр логов агента

Linux:

```
journalctl -u dk-mark-agent -f
```

Windows:

```
Просмотр событий → Журналы Windows → Система → DKMarkAgent
```

Перезапуск сервисов

```
docker compose restart backend
docker compose restart # все сервисы
```