Руководство по развертыванию и установке “Система диагностики электродвигателей”  
Версия 1.0 | 25.08.2025

### 1. ПРЕДВАРИТЕЛЬНЫЕ ТРЕБОВАНИЯ

#### 1.1. Аппаратные требования

| **Компонент** | **Минимальные требования** | **Рекомендуемые требования** |
| --- | --- | --- |
| CPU | 4 ядра | 8 ядер |
| RAM | 8 GB | 16 GB |
| HDD | 100 GB | 500 GB SSD |
| Сеть | 1 Gbps | 10 Gbps |

#### 1.2. Программные требования

* ОС: Ubuntu 20.04 LTS / CentOS 8
* Docker: версия 20.10+
* Docker Compose: версия 2.0+
* Kubernetes (опционально): версия 1.23+

#### 1.3. Сетевые требования

* Открытые порты: 80, 443, 3000, 5432, 9092
* Доступ к интернету для загрузки образов
* Статические IP-адреса для серверов

### 2. УСТАНОВКА И НАСТРОЙКА

#### 2.1. Подготовка среды

bash

*# Обновление системы*

sudo apt update && sudo apt upgrade -y

*# Установка Docker*

curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

*# Установка Docker Compose*

sudo curl -L "https://github.com/docker/compose/releases/download/v2.20.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

*# Добавление пользователя в группу docker*

sudo usermod -aG docker $USER

#### 2.2. Клонирование репозитория

bash

git clone https://github.com/company/motor-diagnostics-system.git

cd motor-diagnostics-system/deploy

#### 2.3. Настройка переменных окружения

Создайте файл .env:

env

# База данных

POSTGRES\_DB=motordb

POSTGRES\_USER=admin

POSTGRES\_PASSWORD=secure\_password\_123

# Redis

REDIS\_PASSWORD=redis\_pass\_456

# JWT Secret

JWT\_SECRET=your\_jwt\_secret\_here

# Доменное имя

DOMAIN=diagnostics.yourcompany.com

### 3. РАЗВЕРТЫВАНИЕ МИКРОСЕРВИСОВ

#### 3.1. Запуск контейнеров

bash

*# Запуск всех сервисов*

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

*# Проверка статуса*

docker-compose ps

#### 3.2. Конфигурация сервисов

API Gateway (Nginx)  
Файл: nginx/nginx.conf

nginx

server {

listen 80;

server\_name diagnostics.yourcompany.com;

location / {

proxy\_pass http://ui:3000;

proxy\_set\_header Host $host;

}

location /api/ {

proxy\_pass http://api-gateway:3000;

proxy\_set\_header Host $host;

}

}

Сервис авторизации  
Файл: auth-service/Dockerfile

dockerfile

FROM node:18-alpine

WORKDIR /app

COPY package\*.json ./

RUN npm ci --only=production

COPY . .

EXPOSE 3001

CMD ["node", "server.js"]

### 4. НАСТРОЙКА БАЗЫ ДАННЫХ

#### 4.1. Инициализация БД

bash

*# Подключение к PostgreSQL*

docker exec -it postgres psql -U admin motordb

*# Выполнение миграций*

\i /docker-entrypoint-initdb.d/init.sql

#### 4.2. Резервное копирование

bash

*# Ежедневный бэкап*

0 2 \* \* \* docker exec postgres pg\_dump -U admin motordb > /backups/motordb\_$(date +%Y%m%d).sql

### 5. КОНФИГУРАЦИЯ БЕЗОПАСНОСТИ

#### 5.1. SSL сертификаты

bash

*# Установка Certbot*

sudo apt install certbot python3-certbot-nginx

*# Получение сертификата*

sudo certbot --nginx -d diagnostics.yourcompany.com

#### 5.2. Настройка firewall

bash

*# Разрешение необходимых портов*

sudo ufw allow 80/tcp

sudo ufw allow 443/tcp

sudo ufw allow 22/tcp

sudo ufw enable

### 6. МОНИТОРИНГ И ЛОГИРОВАНИЕ

#### 6.1. Настройка Prometheus

Файл: prometheus/prometheus.yml

yaml

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'node'

static\_configs:

- targets: ['node-exporter:9100']

- job\_name: 'api'

static\_configs:

- targets: ['api-service:3000']

#### 6.2. Настройка Grafana

* Импорт дашбордов из grafana/dashboards/
* Настройка источников данных

### 7. ПРОВЕРКА РАБОТОСПОСОБНОСТИ

#### 7.1. Тестирование сервисов

bash

*# Проверка здоровья сервисов*

curl http://localhost:3000/health

*# Проверка базы данных*

docker exec postgres pg\_isready -U admin

*# Проверка Redis*

docker exec redis redis-cli ping

#### 7.2. Тестовые данные

bash

*# Загрузка тестовых данных*

docker exec -it api-service node scripts/seed-data.js

### 8. ОБНОВЛЕНИЕ СИСТЕМЫ

#### 8.1. Процесс обновления

bash

*# Остановка сервисов*

docker-compose down

*# Обновление кода*

git pull origin main

*# Пересборка и запуск*

docker-compose build --no-cache

docker-compose up -d

#### 8.2. Откат изменений

bash

*# Восстановление из бэкапа*

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

### 9. АВАРИЙНОЕ ВОССТАНОВЛЕНИЕ

#### 9.1. Восстановление базы данных

bash

*# Остановка приложения*

docker-compose stop api-service

*# Восстановление из бэкапа*

docker exec -i postgres psql -U admin motordb < /backups/motordb\_backup.sql

*# Запуск приложения*

docker-compose start api-service