

УТВЕРЖДАЮ

Генеральный директор
ООО «ИС ИнтерМед»

Карпов О.Ю.
ОЮ



_____ О.Ю. Карпов
« ____ » _____ 2026 г.

Система интеллектуального анализа медицинской документации

"CheckAIMed"

ИНСТРУКЦИЯ ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ

На 20 листах

Оглавление

Система интеллектуального анализа медицинской документации "CheckAIMed"	1
Описание системы	4
Системные требования	5
Минимальные требования к серверу:	5
Требования к сети:	5
Шаг 1. Подготовка операционной системы	6
Для Ubuntu/Debian:	6
Для CentOS/RHEL:	6
Шаг 2. Установка Docker и Docker Compose	7
Установка Docker:	7
Установка Docker Compose:	7
Шаг 3. Подготовка директории приложения	8
Шаг 4. Настройка переменных окружения	9
Шаг 5. Настройка SSL сертификатов	10
Создание self-signed сертификатов (для тестирования):	10
Для продуктивной среды рекомендуется использовать Let's Encrypt:	10
Шаг 6. Настройка системных параметров	11
Создание директорий для данных:	11
Создание Docker-сети:	11
Шаг 7. Запуск системы	12
Проверка конфигурации:	12
Поэтапный запуск сервисов:	12
Полный запуск системы:	12
Шаг 8. Инициализация и настройка системы	13
Выполнение миграций базы данных:	13
Создание первого администратора:	13
Шаг 9. Проверка работоспособности системы	14
Проверка доступности веб-интерфейсов:	14
Проверка состояния сервисов:	14
Проверка сетевых портов:	14
Шаг 10. Настройка автозапуска и мониторинга	15
Создание systemd службы:	15
Настройка резервного копирования:	15
Шаг 11. Настройка доменного имени и DNS	17
Шаг 12. Первоначальная настройка системы	19
Первоначальная конфигурация:	19
Устранение типичных проблем	19

Проблема: Контейнеры не запускаются	19
Проблема: Сеть medai-network не найдена	19
Проблема: PostgreSQL не принимает подключения	19
Проблема: Health check не возвращает UP	19
Контактная информация для поддержки	20

Описание системы

CheckAIMed — это программный комплекс для анализа медицинских документов, клинических рекомендаций и нормативных документов с использованием AI. Система предназначена для автоматизированной проверки соответствия медицинской документации клиническим рекомендациям, управления пациентами, разметки документов, формирования отчётов и администрирования пользователей.

Архитектура системы реализована на основе микросервисного подхода с применением контейнеризации Docker и включает в себя:

- React UI (doc-admin-front) — веб-интерфейс админ-панели
- Admin API (doc-admin-front-api, порт 8080) — основной API приложения
- Client Docs Analyser (client-docs-analyser, порт 8081) — анализ документов клиентов, очереди обработки, WebSocket-уведомления (/ws-sock)
- Clinrec Analyser (clinrec-analyser, порт 8082) — поиск и загрузка клинических рекомендаций, AI-извлечение условий
- PostgreSQL 15 с Flyway-миграциями, nginx (reverse proxy), JWT/RBAC-авторизация, интеграция с AI через PROXY_TOKEN

Системные требования

Минимальные требования к серверу:

- Процессор: 4–8 ядер CPU (Intel/AMD), рекомендуется 8 ядер
- Оперативная память: 8 GB RAM (рекомендуется 16 GB)
- Дисковое пространство: 100 GB SSD (для production с документами рекомендуется 500 GB и более)
- Операционная система:
 - Ubuntu Server 20.04 LTS или выше
 - CentOS 8/9 или RHEL 8/9
 - Debian 11 или выше
 - Windows Server 2019/2022 с Docker Desktop

Требования к сети:

- Статический IP-адрес или доменное имя
- Открытые порты:
 - 80, 443 (HTTP/HTTPS — nginx)
 - 5432 (PostgreSQL — только для разработки, в production не публикуется наружу)
 - 8080–8082 (backend-сервисы — только для разработки)
 - 3000 (frontend — только для разработки)

Шаг 1. Подготовка операционной системы

Для Ubuntu/Debian:

```
# Обновление системы
sudo apt update && sudo apt upgrade -y

# Установка необходимых пакетов
sudo apt install -y curl wget git nano htop rsync ca-certificates gnupg unzip

# Настройка файрвола
sudo ufw allow 22
sudo ufw allow 80
sudo ufw allow 443

sudo ufw enable
```

Для CentOS/RHEL:

```
# Обновление системы
sudo dnf update -y

# Установка необходимых пакетов
sudo dnf install -y curl wget git nano htop rsync ca-certificates gnupg unzip

# Настройка файрвола
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=443/tcp

sudo firewall-cmd --reload
```

Шаг 2. Установка Docker и Docker Compose

Установка Docker:

```
# Удаление старых версий
sudo apt remove docker docker-engine docker.io containerd runc

# Установка зависимостей
sudo apt install -y apt-transport-https ca-certificates curl gnupg lsb-release

# Добавление официального GPG ключа Docker
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

# Добавление репозитория Docker
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

# Установка Docker
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io

# Добавление пользователя в группу docker
sudo usermod -aG docker $USER

# Включение автозапуска Docker
sudo systemctl enable docker
sudo systemctl start docker
```

Установка Docker Compose:

```
# Скачивание Docker Compose
sudo curl -L
"https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose

# Установка прав на выполнение
sudo chmod +x /usr/local/bin/docker-compose

# Проверка установки
docker compose version
```

Шаг 3. Подготовка директории приложения

```
# Создание рабочей директории
mkdir -p ~/medai-app

# Копирование файлов развёртывания (при деплое через CI/CD они копируются
автоматически)
# docker-compose.yml, docker-compose.db.yml, nginx/nginx.conf, deploy.sh, .env.example
cd ~/medai-app

# Установка прав на скрипты
chmod +x deploy.sh deploy-gitlab.sh stop.sh
```

Шаг 4. Настройка переменных окружения

Создайте файл .env в корневой директории проекта:

```
nano .env
```

Содержимое файла .env (на основе .env.example):

```
# PostgreSQL
POSTGRES_DATA_PATH=/var/lib/docker/volumes/medai_postgres_data/_data
POSTGRES_DB=medai
POSTGRES_USER=medai
POSTGRES_PASSWORD=your_secure_password_here

# Профиль Spring
SPRING_PROFILES_ACTIVE=prod

# Docker-образы (обязательно указать актуальные теги из registry)
BACKEND_IMAGE=registry.example.com/group/project/backend:latest
CLIENT_ANALYSER_IMAGE=registry.example.com/group/project/client-analyser-backend:latest
CLINREC_ANALYSER_IMAGE=registry.example.com/group/project/clinrec-analyser-backend:latest
FRONTEND_IMAGE=registry.example.com/group/project/frontend:latest

# Идентификатор заказчика (выбор prompt-ов и сегментация данных)
CUSTOMER_NAME=YOUR_CUSTOMER

# Секреты (обязательно заменить на production-значения!)
JWT_SECRET=your_secret_jwt_key_here_minimum_32_characters
PROXY_TOKEN=your_proxy_token_here

# Пути к данным на хосте
DOCUMENTS_PATH=~/.medai/docs/clients
CUSTOMER_PROMPTS_PATH=~/.medai/prompts
CLINREC_OUTPUT_PATH=~/.medai/clinrec/output
CLINREC_YAML_PATH=~/.medai/clinrec/yaml

# URL фронтенда (для client-docs-analyser)
FRONTEND_BASE_URL=https://your-domain.ru

# Установка прав доступа
chmod 600 .env
```

Шаг 5. Настройка SSL сертификатов

Создание self-signed сертификатов (для тестирования):

```
# Создание директории для SSL
```

```
mkdir -p ssl
```

```
# Генерация приватного ключа
```

```
openssl genrsa -out ssl/nginx.key 2048
```

```
# Генерация сертификата
```

```
openssl req -new -x509 -key ssl/nginx.key -out ssl/nginx.crt -days 365 -subj  
"/C=RU/ST=Moscow/L=Moscow/O=YourOrganization/CN=your-domain.ru"
```

Для продуктивной среды рекомендуется использовать Let's Encrypt:

```
# Установка certbot
```

```
sudo apt install -y certbot
```

```
# Получение сертификата
```

```
sudo certbot certonly --standalone -d your-domain.ru
```

```
# Копирование сертификатов
```

```
sudo cp /etc/letsencrypt/live/your-domain.ru/fullchain.pem ssl/nginx.crt
```

```
sudo cp /etc/letsencrypt/live/your-domain.ru/privkey.pem ssl/nginx.key
```

Шаг 6. Настройка системных параметров

Создание директорий для данных:

```
# Создание директорий для volumes и данных приложения
```

```
mkdir -p ~/medai/docs/clients
```

```
mkdir -p ~/medai/prompts
```

```
mkdir -p ~/medai/clinrec/output
```

```
mkdir -p ~/medai/clinrec/yaml
```

```
mkdir -p ~/medai-app/ssl
```

Создание Docker-сети:

```
# Сеть medai-network должна существовать до запуска compose
```

```
docker network create medai-network
```

Шаг 7. Запуск системы

Проверка конфигурации:

```
# Проверка синтаксиса docker-compose.yml
docker compose config
```

Поэтапный запуск сервисов:

1. Запуск базы данных:

```
# Запуск PostgreSQL
docker compose -f docker-compose.db.yml up -d

# Проверка готовности PostgreSQL
docker exec medai-postgres pg_isready -U medai -d medai
```

2. Запуск backend-сервисов и frontend:

```
# Запуск всех сервисов приложения
docker compose up -d --force-recreate --pull always

# Или используйте скрипт автоматического развёртывания:
./deploy.sh
```

3. Альтернатива — скрипт deploy-gitlab.sh (для GitLab CI):

```
# Загружает образы из .env и .env.images
./deploy-gitlab.sh
```

4. Перезапуск nginx (при изменении конфигурации):

```
docker restart medai-nginx
```

5. Проверка запущенных контейнеров:

```
docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"
```

Полный запуск системы:

```
# Полный запуск одной командой (после настройки .env)
./deploy.sh

# Просмотр статуса всех контейнеров
docker-compose ps

# Просмотр логов
docker-compose logs -f
```

Шаг 8. Инициализация и настройка системы

Выполнение миграций базы данных:

Миграции базы данных выполняются автоматически при старте backend-сервисов (Flyway)
Проверить логи миграций можно в контейнере backend:
`docker compose logs backend | grep -i flyway`

Создание первого администратора:

Первого пользователя создайте через веб-интерфейс админ-панели
или через API авторизации (раздел «Управление доступом»)

При первом входе назначьте роль SUPERADMIN или ADMIN
(роли создаются миграцией V46__seed_rbac_permissions_and_roles.sql)

Смените пароль после первого входа в настройках профиля

Шаг 9. Проверка работоспособности системы

Проверка доступности веб-интерфейсов:

- Веб-интерфейс: <https://your-domain.ru> (или <http://localhost> для локальной установки)
- Admin API health: <http://localhost/api/actuator/health>
- Client Analyser health: <http://localhost/api/client-analysis/actuator/health>
- Clinrec Analyser health: <http://localhost/api/clinrec-analysis/actuator/health>

Проверка состояния сервисов:

```
# Проверка статуса всех контейнеров  
docker-compose ps
```

```
# Проверка логов конкретного сервиса  
docker compose logs backend  
docker compose logs client-analyser-backend  
docker compose logs clinrec-analyser-backend
```

```
# Проверка подключения к базе данных  
docker compose exec postgres psql -U medai -d medai -c "SELECT version();"
```

Проверка сетевых портов:

```
# Проверка портов  
netstat -tulpn | grep -E ":(80|443|5432|8080|8081|8082|3000)"
```

Шаг 10. Настройка автозапуска и мониторинга

Создание systemd службы:

```
sudo nano /etc/systemd/system/checkaimed.service
```

Содержимое файла службы:

```
[Unit]
Description=CheckAIMed System
Requires=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/home/deploy/medai-app
ExecStart=/usr/bin/docker compose up -d
ExecStop=/usr/bin/docker compose down
TimeoutStartSec=0
User=deploy

[Install]
WantedBy=multi-user.target
# Включение службы
sudo systemctl enable checkaimed.service
sudo systemctl start checkaimed.service
```

```
# Проверка статуса
sudo systemctl status checkaimed.service
```

Настройка резервного копирования:

```
# Создание скрипта резервного копирования
sudo nano ~/medai-app/backup.sh
```

Содержимое скрипта:

```
#!/bin/bash
BACKUP_DIR="/opt/backups/checkaimed"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# Резервная копия PostgreSQL
docker compose exec -T postgres pg_dump -U medai medai | gzip >
$BACKUP_DIR/postgres_$DATE.sql.gz

# Резервная копия документов клиентов
tar -czf $BACKUP_DIR/documents_$DATE.tar.gz ~/medai/docs/clients/

# Резервная копия prompt-ов и результатов clinrec
tar -czf $BACKUP_DIR/prompts_$DATE.tar.gz ~/medai/prompts/

# Удаление старых копий (старше 7 дней)
find $BACKUP_DIR -name "*.gz" -mtime +7 -delete
```

```
echo "Backup completed: $DATE"
```

```
# Установка прав на выполнение
```

```
sudo chmod +x ~/medai-app/backup.sh
```

```
# Добавление в crontab (ежедневно в 2:00)
```

```
echo "0 2 * * * ~/medai-app/backup.sh >> /var/log/checkaimed_backup.log 2>&1" | crontab -
```

Шаг 11. Настройка доменного имени и DNS

11.1. Настройка DNS записей

Если вы используете реальный домен, настройте DNS записи у вашего регистратора:

A записи:

`your-domain.ru IN A your-server-ip`

`www.your-domain.ru IN A your-server-ip`

Если это локальная установка без публичного домена, отредактируйте файл `hosts`:

Windows: `C:\Windows\System32\drivers\etc\hosts`

Linux/Mac: `/etc/hosts`

Добавьте строки:

`your-server-ip your-domain.ru`

`your-server-ip www.your-domain.ru`

Отредактируйте файл `nginx/nginx.conf`. Найдите строку:

`server_name localhost;`

Замените на:

`server_name your-domain.ru www.your-domain.ru;`

11.2. Обновление переменных окружения (`.env`)

В корне проекта отредактируйте файл `.env`:

`# Настройки приложения`

`FRONTEND_BASE_URL=https://your-domain.ru`

`CUSTOMER_NAME=YOUR_CUSTOMER`

11.3. SSL сертификаты

Сертификаты должны находиться в папке `ssl/`:

`ssl/nginx.crt` — SSL сертификат

`ssl/nginx.key` — Приватный ключ

Для самоподписанных сертификатов (локальная разработка):

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ssl/nginx.key -out ssl/nginx.crt  
-subj "/CN=your-domain.ru"
```

Для Let's Encrypt (продакшен): используйте `certbot` и скопируйте сертификаты в папку `ssl/`.

11.4. Перезапуск сервисов

После изменения конфигурации выполните:

```
docker compose restart nginx
```

```
docker compose restart backend
```

```
docker compose restart client-analyser-backend clinrec-analyser-backend frontend
```

Шаг 12. Первоначальная настройка системы

12.1. Вход в систему

1. Откройте браузер и перейдите по адресу: `https://your-domain.ru`
2. Создайте первого пользователя через раздел «Управление доступом»:
 - Назначьте роль SUPERADMIN для полного доступа к системе
 - Или роль ADMIN для администрирования без суперправ
3. Смените пароль после первого входа в настройках профиля.

Первоначальная конфигурация:

- Настройка заказчика: убедитесь, что CUSTOMER_NAME в .env соответствует набору промпт-ов
- Создание пользователей: добавьте врачей, руководителей и администраторов с нужными ролями
- Загрузка документов: проверьте загрузку ZIP-архивов и обработку документов пациентов
- Настройка клинических рекомендаций: загрузите или импортируйте клинреки через clinrec-analyser
- Проверка AI: убедитесь, что PROXY_TOKEN настроен и анализ документов выполняется корректно

Устранение типичных проблем

Проблема: Контейнеры не запускаются

```
# Проверка доступности портов
sudo netstat -tulpn | grep LISTEN
```

```
# Очистка и перезапуск
docker-compose down
docker system prune -f
docker-compose up -d
```

Проблема: Сеть medai-network не найдена

```
# Создание Docker-сети
docker network create medai-network
```

```
# Проверка существующих сетей
docker network ls | grep medai
```

Проблема: PostgreSQL не принимает подключения

```
# Проверка логов
docker-compose logs postgres
```

```
# Проверка подключения
docker compose exec postgres psql -U medai -d medai
```

Проблема: Health check не возвращает UP

```
# Проверка логов backend-сервисов
docker compose logs backend
docker compose logs client-analyser-backend
# Проверьте JWT_SECRET и PROXY_TOKEN в .env
docker compose up -d --force-recreate
```

Контактная информация для поддержки

Техническая поддержка системы:

- **Телефон:** +7 (929) 110-64-42
- **WhatsApp:** +7 (929) 110-64-42
- **Telegram:** https://t.me/telepacs_support_bot
- **Электронная почта:** info@misintermed.ru