Cortex

ООО «КОРТЕКС»
350000, г. Краснодар, ул. Северная, д. 324, 10 этаж
www.cortex.ooo

«Онкологическая МИС»
Руководство по развертыванию

ООО «КОРТЕКС»
350000, г. Краснодар, ул. Северная, д. 324, 10 этаж
www.cortex.ooo

Введение

Программное обеспечение «Онкологическая МИС» — клиент-серверное приложение, состоящее из трёх независимых сервисов. Каждый сервис включает frontend, backend и базу данных PostgreSQL. Рекомендуется развертывание с использованием Docker Compose.

Состав системы

Сервис	Компоненты	Порты	База данных
onco-service	front, back, db	3000 (front), 5000 (back), 35432 (db)	oncodb_shared
dicom-markup	front, back, db	3001 (front), 5001 (back), 35433 (db)	oncodb_shared (общая с onco-service)
research-registry	front, service, db	4173 (front), 5005 (service), 5435 (db)	research_registry

Подготовка сервера

bash

```
# ==========================================
# 1. Установить Docker и docker-compose
# ==========================================
apt-get update
apt-get install docker-engine docker-compose-v2

# ==========================================
# 2. Запустить Docker
# ==========================================
systemctl enable --now docker

# ==========================================
# 3. Создать структуру каталогов
# ==========================================
mkdir -p /opt/onco-service/{krr_tor,db}
mkdir -p /opt/dicom-markup/{uploads,db}
mkdir -p /opt/research-registry/{postgresql/data,logs,files}
```

Развертывание

Шаг 1. Создать файлы конфигурации

onco-service (/opt/onco-service/docker-compose.yml):

```
yaml
```

```
services:
  db:
    container_name: onco-service-db
    image: postgres:15
    restart: always
    volumes:
      - /opt/onco-service/db:/var/lib/postgresql/data
    environment:
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    ports:
      - "35432:5432"
  back:
    image: docker.direct.farm/repository/onco-service:test
    container_name: onco-service-back
    restart: always
    volumes:
      - /opt/onco-service/krr_tor:/krr_tor
    ports:
      - "5000:5000"
    depends_on:
      - db
  front:
    image: docker.direct.farm/repository/onco-service-front:test
    container_name: onco-service-front
    restart: always
    ports:
      - "3000:4173"
    depends_on:
      - back
```

onco-service (/opt/onco-service/.env):

bash

```
# ============================================
# Учётные данные PostgreSQL — ЗАМЕНИТЕ НА СВОИ
# ============================================

# Общая БД для onco-service и dicom-markup
POSTGRES_DB=oncodb_shared

# Пользователь БД
POSTGRES_USER=postgres

# Пароль — ОБЯЗАТЕЛЬНО ЗАМЕНИТЕ!
POSTGRES_PASSWORD=замените_на_свой_пароль
```

```
# ==========================================
# Настройки фронтенда
# ==========================================

# URL API для фронтенда
VITE_APP_API_URL=https://ваш_домен/api
```

dicom-markup (/opt/dicom-markup/docker-compose.yml):

```
yaml
services:
  db:
    container_name: dicom-markup-db
    image: postgres:15
    restart: always
    volumes:
      - /opt/dicom-markup/db:/var/lib/postgresql/data
    environment:
      - POSTGRES_DB=${POSTGRES_DB}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
    ports:
      - "35433:5432"
  back:
    image: docker.direct.farm/repository/dicom-markup:test
    container_name: dicom-markup-back
    restart: always
    ports:
      - "5001:5000"
    volumes:
      - /opt/dicom-markup/uploads:/app/uploads
    depends_on:
      - db
  front:
    image: docker.direct.farm/repository/dicom-markup-front:test
    container_name: dicom-markup-front
    restart: always
    ports:
      - "3001:4173"
    depends_on:
      - back
```

dicom-markup (/opt/dicom-markup/.env):
bash

```
# ==========================================
# Учётные данные PostgreSQL — ЗАМЕНИТЕ НА СВОИ
# ==========================================

# Та же БД, что и у onco-service
```

```
POSTGRES_DB=oncodb_shared

# Пользователь БД
POSTGRES_USER=postgres

# Пароль — ОБЯЗАТЕЛЬНО ЗАМЕНИТЕ!
POSTGRES_PASSWORD=замените_на_свой_пароль
```

research-registry (/opt/research-registry/docker-compose.yml):

```
yaml
services:
  research-registry-db:
    image: postgres:17.4-alpine
    restart: always
    container_name: research-registry-db
    environment:
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: ${POSTGRES_DB}
    ports:
      - "5435:5432"
    volumes:
      - /opt/research-registry/postgresql/data:/var/lib/postgresql/data
    networks:
      - research-registry
  research-registry-service:
    image: docker.direct.farm/repository/research-registry-service:develop
    container_name: research-registry-service
    restart: always
    environment:
      - DEBUG=true
      - DEBUG_SQL=false
      - POSTGRES_HOST=research-registry-db
      - POSTGRES_PORT=5432
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
      - POSTGRES_DB=${POSTGRES_DB}
      - JWT_SECRET_KEY=${JWT_SECRET_KEY}
      - ACCESS_TOKEN_EXPIRED=36000
      - REFRESH_TOKEN_EXPIRED=72000
      - STORAGE_DICOM=/root/files/dicom
      - STORAGE_SVS=/root/files/svs
      - LOG_FILE_PATH=/root/logs/app.log
    volumes:
      - /opt/research-registry/logs:/root/logs
      - /opt/research-registry/files:/root/files
    ports:
      - "5005:5000"
    depends_on:
      - research-registry-db
```

```
    networks:
      - research-registry
  research-registry-front:
    image: docker.direct.farm/repository/research-registry-front:develop
    container_name: research-registry-front
    restart: always
    environment:
      - VITE_API_URL=${VITE_API_URL}
    ports:
      - "4173:4173"
    networks:
      - research-registry
networks:
  research-registry:
```

research-registry (/opt/research-registry/.env):

bash

```
# ============================================
# Учётные данные PostgreSQL — ЗАМЕНИТЕ НА СВОИ
# ============================================

POSTGRES_USER=postgres
POSTGRES_PASSWORD=замените_на_свой_пароль
POSTGRES_DB=research_registry          # Отдельная БД реестра исследований

# ============================================
# JWT аутентификация — ЗАМЕНИТЕ НА СВОЙ
# ============================================
# Сгенерировать: openssl rand -hex 64
JWT_SECRET_KEY=замените_на_свой_секретный_ключ

# ============================================
# URL API для фронтенда — ЗАМЕНИТЕ НА СВОЙ
# ============================================

VITE_API_URL=http://ваш_IP:5005   # ⚠️ замените на свой!
```

Шаг 2. Авторизоваться в реестре образов

bash

```
# ============================================
# Получить учётные данные у администратора реестра
# ============================================

docker login docker.direct.farm
```

ООО «КОРТЕКС»
350000, г. Краснодар, ул. Северная, д. 324, 10 этаж
www.cortex.ooo

Шаг 3. Загрузить образы и запустить

bash

```
# ============================================
# Запуск всех сервисов
# ============================================

# 1. Onco Service
cd /opt/onco-service && docker compose pull && docker compose up -d

# 2. DICOM Markup
cd /opt/dicom-markup && docker compose pull && docker compose up -d

# 3. Research Registry
cd /opt/research-registry && docker compose pull && docker compose up -d
```

Шаг 4. Проверить доступность

bash

```
# ============================================
# Проверка доступности сервисов (HTTP заголовки)
# ============================================

curl -I http://localhost:3000

curl -I http://localhost:5000

curl -I http://localhost:3001

curl -I http://localhost:5001

curl -I http://localhost:4173

curl -I http://localhost:5005
```

Диагностика

В случае проблем выполнить:

bash

```
# ============================================
# Логи сервисов (последние 50 строк)
# ============================================

# 1. Onco Service Backend
docker logs onco-service-back --tail 50
```

```
# 2. Onco Service Frontend
docker logs onco-service-front --tail 50

# 3. DICOM Markup Backend
docker logs dicom-markup-back --tail 50

# 4. Research Registry Service
docker logs research-registry-service --tail 50
```

bash

```
# ==========================================
# Логи баз данных (последние 50 строк)
# ==========================================

# 1. PostgreSQL (Onco Service)
docker logs onco-service-db --tail 50

# 2. PostgreSQL (DICOM Markup)
docker logs dicom-markup-db --tail 50

# 3. PostgreSQL (Research Registry)
docker logs research-registry-db --tail 50
```

bash

```
# ==========================================
# Проверить, что контейнеры запущены
# ==========================================

docker ps
```

Наиболее частые проблемы:

- Неверные учётные данные СУБД — проверить .env файлы во всех трёх сервисах
- Недоступность реестра образов — проверить docker login
- CORS-ошибки в браузере — проверить конфигурацию Nginx
- База данных не запускается — проверить права на каталоги /opt/*/db
- Бэкенд не может подключиться к БД — убедиться, что POSTGRES_HOST совпадает с именем контейнера БД

Настройка Nginx (опционально)

Для доступа по доменному имени добавьте конфигурацию:

```
# ==========================================
# Nginx конфигурация
# ==========================================
```

```
# ==========================================
# HTTPS сервер (порт 443)
# ==========================================
server {
    listen 443 ssl;
    server_name ваш_домен;

    # SSL-сертификаты — укажите свои пути
    ssl_certificate     /etc/nginx/ssl/cert.crt;
    ssl_certificate_key /etc/nginx/ssl/private.key;

    # Проксирование фронтенда (Onco Service)
    location / {
        proxy_pass http://ваш_IP:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    # Проксирование API (Onco Service Backend)
    location /api/ {
        proxy_pass http://ваш_IP:5000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

# ==========================================
# HTTP сервер (порт 80) — перенаправление на HTTPS
# ==========================================
server {
    listen 80;
    server_name ваш_домен;
    return 301 https://$host$request_uri;
}
```

Резервное копирование

bash

```
# ==========================================
# Создать каталог для бекапов
# ==========================================
```

ООО «КОРТЕКС»
350000, г. Краснодар, ул. Северная, д. 324, 10 этаж
www.cortex.ooo

```
mkdir -p /backup
```

bash

```
# ==========================================
# Дампы баз данных (PostgreSQL)
# ==========================================

# 1. Дамп БД Onco Service
docker exec onco-service-db pg_dumpall -U postgres > /backup/dump_onco_$(date
+%Y%m%d).sql

# 2. Дамп БД DICOM Markup
docker exec dicom-markup-db pg_dumpall -U postgres > /backup/dump_dicom_$(date
+%Y%m%d).sql

# 3. Дамп БД Research Registry
docker exec research-registry-db pg_dumpall -U postgres >
/backup/dump_registry_$(date +%Y%m%d).sql
```

bash

```
# ==========================================
# Архивация файлов томов
# ==========================================

# 1. Файлы Onco Service (каталог krr_tor)
tar -czf /backup/onco_files_$(date +%Y%m%d).tar.gz -C /opt/onco-service/krr_tor .

# 2. Файлы DICOM Markup (каталог uploads)
tar -czf /backup/dicom_uploads_$(date +%Y%m%d).tar.gz -C /opt/dicom-markup/uploads
.

# 3. Файлы Research Registry (весь каталог)
tar -czf /backup/registry_opt_$(date +%Y%m%d).tar.gz -C /opt/research-registry .
```

Профилактика БД

При снижении производительности выполнить:

bash

```
# ==========================================
# Оптимизация производительности БД (VACUUM ANALYZE)
# ==========================================
# Выполнять при снижении производительности
# ==========================================

# 1. Onco Service DB
docker exec onco-service-db psql -U postgres -d oncodb_shared -c "VACUUM ANALYZE;"
```

ООО «КОРТЕКС»
350000, г. Краснодар, ул. Северная, д. 324, 10 этаж
www.cortex.ooo

```
# 2. DICOM Markup DB
docker exec dicom-markup-db psql -U postgres -d oncodb_shared -c "VACUUM ANALYZE;"

# 3. Research Registry DB
docker exec research-registry-db psql -U postgres -d research_registry -c "VACUUM
ANALYZE;"
```

Примечания

- Все пароли, IP и секретные ключи в примерах — заглушки. Обязательно замените их на собственные.
- Для генерации надёжного JWT-секрета: openssl rand -hex 64
- Для генерации пароля PostgreSQL: openssl rand -base64 16
- onco-service и dicom-markup используют общую базу данных oncodb_shared. Убедитесь, что пароли в обоих .env совпадают.
- research-registry использует отдельную базу данных research_registry.
- В продакшн-среде рекомендуется использовать always restart policy и настроить мониторинг.