Cortex

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

Руководство по развертыванию

Интеллектуальная система анализа биомаркеров и поддержки врачебных решений «Кортекс. Биомаркеры»

02.06.2026

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

Введение

Интеллектуальная система анализа биомаркеров и поддержки врачебных решений «Кортекс. Биомаркеры» — клиент-серверное приложение с микросервисной архитектурой, предназначенное для интеллектуального поиска и анализа медицинской информации по технологии RAG (Retrieval-Augmented Generation).

Система состоит из девяти контейнеров, которые разворачиваются единым файлом Docker Compose.

Все обращения к большим языковым моделям (LLM) проходят через единый шлюз Comrade Router. Данные хранятся в PostgreSQL с расширением pgvector (векторный поиск) и в графовой СУБД Neo4j (граф знаний). Внешний доступ обеспечивается обратным прокси Nginx, который является единой точкой входа в систему.

Состав системы

Сервис (контейнер)	**Назначение**	**Порт (контейнер)**	**Внешний порт**
nginx	Обратный прокси, единая точка входа	80	8081
medrag-fornt	Веб-интерфейс (React / Vite)	80	—
medical-rag-main	Основной backend, REST API /api, JWT-аутентификация	8084	— (через Nginx)
medical-rag	RAG-сервис: векторный и графовый поиск (gRPC)	8081	—
medical-dataloom	Извлечение и структурирование данных из документов (gRPC)	8082	—
medrag-agent	Агент-оркестратор сценариев	—	—
comrade-router	Шлюз маршрутизации запросов к LLM	8080 / 50051	32658
postgres	PostgreSQL + pgvector (базы main, rag)	5432	—
neo4j	Графовая СУБД (база medrag)	7687	—

Подготовка сервера

bash

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

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

```
apt-get install docker-engine docker-compose-v2

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

# ==========================================
# 3. Создать структуру каталогов
# ==========================================
mkdir -p /opt/medrag/storage/medical-rag-main/attachments
mkdir -p /opt/medrag/storage/medical-data-loom
mkdir -p /opt/medrag/storage/medical-rag
mkdir -p /opt/medrag/storage/medrag-agent
mkdir -p /opt/medrag/storage/comrade-router/database
mkdir -p /opt/medrag/storage/nginx
mkdir -p /opt/medrag/storage/postgres/data
mkdir -p /opt/medrag/storage/neo4j/{data,plugins,import}
```

Развертывание

Шаг 1. Создать файл Docker Compose

Файл /opt/medrag/docker-compose.yaml:

yaml

```
services:
 medical-rag-main:
  image: docker.direct.farm/medical-back-main:dev
  entrypoint: ["/root/medical-back-main", "--config", "/root/config-ext.toml"]
  restart: unless-stopped
  volumes:
   - ./storage/medical-rag-main/config-ext.toml:/root/config-ext.toml
   - ./storage/medical-rag-main/attachments:/root/storage/attachments

 medical-dataloom:
  image: docker.direct.farm/medical-data-loom:dev
  volumes:
   - ./storage/medical-data-loom/config-ext.toml:/root/config.toml

 medical-rag:
  image: docker.direct.farm/medical-rag:main
  volumes:
   - ./storage/medical-rag/config-ext.toml:/root/config-ext.toml

 medrag-agent:
  image: docker.direct.farm/medrag-agent
  volumes:
   - ./storage/medrag-agent/config.yaml:/root/config.yaml
```

```
medrag-fornt:
  image: docker.direct.farm/ai-front:latest

comrade-router:
  image: docker.direct.farm/comrade-router
  environment:
    - DB_PATH=/root/database/router.db
  volumes:
    - ./storage/comrade-router/database:/root/database
  ports:
    - "32658:50051"

nginx:
  image: nginx:alpine
  volumes:
    - ./storage/nginx/nginx.conf:/etc/nginx/conf.d/default.conf:ro
  ports:
    - "8081:80"

postgres:
  image: pgvector/pgvector:pg15
  environment:
    - POSTGRES_PASSWORD=замените_на_свой_пароль
  volumes:
    - ./storage/postgres/data:/var/lib/postgresql/data

neo4j:
  image: neo4j:5-enterprise
  environment:
    - NEO4J_AUTH=neo4j/замените_на_свой_пароль
    - NEO4J_ACCEPT_LICENSE_AGREEMENT=yes
    - NEO4J_PLUGINS=["apoc", "graph-data-science"]
    - NEO4J_server_config_strict__validation_enabled=false
  volumes:
    - ./storage/neo4j/data:/data
    - ./storage/neo4j/plugins:/plugins
    - ./storage/neo4j/import:/import
```

Шаг 2. Создать файлы конфигурации сервисов

medical-rag-main (/opt/medrag/storage/medical-rag-main/config-ext.toml):

toml

```
[Common]
Env = "dev"
ServiceName = "MainService"
ServiceAddress = ":8084"
BasePath = "/api"
```

```
[Postgres]
Dsn = "host=postgres database=main user=main password=замените_на_свой_пароль port=5432
sslmode=disable"

[Agent]
Url = "http://comrade-router:8080/v1"
ApiKey = "замените_на_ключ_Comrade"
```

medical-dataloom (/opt/medrag/storage/medical-data-loom/config-ext.toml):

toml

```
[Common]
Env = "dev"
ServiceName = "data-loom-service"

[GRPC]
Address = ":8082"

[AI]
PromptTemplate = """
These are lab tests. Find the person's data such as sex, height, weight, and age.
Write them in a single block with the heading #Human: in the format:
name,value
If a value is not found, put 'not_found' as the value.
Also write the test results and their reference values in a single block with the heading #Analysis:.
Without units of measurement. Without articles.
If there are multiple reference values, write them with description and use a semicolon as separator.
For decimal numbers, use a dot as the separator between integer and fractional parts.
Format:
name,value,references
Write everything in English except the values.
Here is the lab test text:
"""

[Comrade]
ApiKey = "замените_на_ключ_Comrade"
Url = "http://comrade-router:8080"
ModelText = "openai/gpt-oss-20b"
ModelImage = "qwen2.5-vl-72b-instruct"

[DoclingServe]
Url = "http://192.168.10.80:5001"
```

medical-rag (/opt/medrag/storage/medical-rag/config-ext.toml):

toml

```
[Postgres]
Dsn = "host=postgres database=rag user=postgres password=замените_на_свой_пароль port=5432"

[Comrade]
```

```
ApiKey = "замените_на_ключ_Comrade"
ApiBase = "http://comrade-router:8080/v1"
Model = "openai/gpt-oss-20b"
RerankModel = "BAAI/bge-reranker-v2-m3"
Temperature = 0.3
EmbedModel = "bge-m3:567m"
EmbedDim = 1024

[Neo4j]
URI = "bolt://neo4j:7687"
User = "neo4j"
Password = "замените_на_свой_пароль"
Database = "medrag"
```

medrag-agent (/opt/medrag/storage/medrag-agent/config.yaml):

yaml

```
config:
  router_address: "comrade-router:50051"

openai_host: "http://comrade-router:8080/v1"
openai_key: "замените_на_ключ_Comrade"
openai_model: "openai/gpt-oss-20b"
rag_service_address: "medical-rag:8081"
data_loom_service_address: "medical-dataloom:8082"
```

nginx (/opt/medrag/storage/nginx/nginx.conf):

nginx

```
server {
    listen 80;

    location / {
        proxy_pass http://medrag-fornt:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /api {
        proxy_pass http://medical-rag-main:8084;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Шаг 3. Авторизоваться в реестре образов

bash

```
# ==========================================
# Получить учётные данные у администратора реестра
# ==========================================
docker login docker.direct.farm
```

Шаг 4. Загрузить образы и запустить

bash

```
# ==========================================
# Загрузка образов и запуск всех сервисов
# ==========================================
cd /opt/medrag
docker compose pull
docker compose up -d
```

Шаг 5. Инициализировать базы данных

Образ PostgreSQL создаёт только служебную базу. Базы main и rag, а также пользователя main и базу Neo4j medrag необходимо создать вручную после первого запуска:

bash

```
# ==========================================
# PostgreSQL: роль и базы данных
# ==========================================
cd /opt/medrag
# 1. Роль приложения medical-rag-main
docker compose exec postgres psql -U postgres \
  -c "CREATE ROLE main WITH LOGIN PASSWORD 'замените_на_свой_пароль';"
# 2. База medical-rag-main (владелец — main)
docker compose exec postgres psql -U postgres -c "CREATE DATABASE main OWNER main;"
# 3. База RAG-сервиса (владелец — postgres)
docker compose exec postgres psql -U postgres -c "CREATE DATABASE rag OWNER postgres;"
# 4. Расширение pgvector в базе rag
docker compose exec postgres psql -U postgres -d rag \
  -c "CREATE EXTENSION IF NOT EXISTS vector;"

# ==========================================
# Neo4j: граф знаний (база medrag)
# ==========================================
docker compose exec neo4j cypher-shell -u neo4j -p 'замените_на_свой_пароль' \
  "CREATE DATABASE medrag IF NOT EXISTS;"
```

Шаг 6. Проверить доступность

bash

```
# ==========================================
# Статус контейнеров и проверка точки входа
```

```
# ============================================
cd /opt/medrag
docker compose ps
curl -I http://localhost:8081
curl -I http://localhost:8081/api
```

Диагностика

В случае проблем выполнить:

bash

```
# ============================================
# Логи сервисов (последние 50 строк)
# ============================================
cd /opt/medrag
docker compose logs medical-rag-main --tail 50
docker compose logs medical-rag      --tail 50
docker compose logs medical-dataloom --tail 50
docker compose logs medrag-agent     --tail 50
docker compose logs comrade-router   --tail 50
```

bash

```
# ============================================
# Логи баз данных и прокси
# ============================================
docker compose logs postgres --tail 50
docker compose logs neo4j    --tail 50
docker compose logs nginx    --tail 50
```

Наиболее частые проблемы:

- Базы данных не созданы — выполнить Шаг 5 (роль main, базы main, rag и medrag).
- Неверные учётные данные СУБД — проверить пароли в config-ext.toml и в docker-compose.yaml (POSTGRES_PASSWORD, NEO4J_AUTH).
- Недоступность реестра образов — проверить docker login docker.direct.farm.
- Ошибки обращения к LLM — проверить доступность comrade-router и корректность ApiKey во всех сервисах.
- Сервис data-loom не обрабатывает документы — проверить доступность внешнего сервиса DoclingServe (http://192.168.10.80:5001).
- Сайт недоступен по адресу — проверить, что порт 8081 открыт и контейнер nginx запущен.

Резервное копирование

bash

```
# ============================================
```

```
# Создать каталог для бекапов
# ==========================================
mkdir -p /backup
```

bash

```
# ==========================================
# Дампы баз данных PostgreSQL
# ==========================================
cd /opt/medrag
# 1. База medical-rag-main
docker compose exec -T postgres pg_dump -U postgres main > /backup/medrag_main_$(date +%Y%m%d).sql
# 2. База RAG-сервиса
docker compose exec -T postgres pg_dump -U postgres rag  > /backup/medrag_rag_$(date +%Y%m%d).sql
```

bash

```
# ==========================================
# Архивация томов (Neo4j, вложения, БД роутера)
# ==========================================
tar -czf /backup/medrag_storage_$(date +%Y%m%d).tar.gz -C /opt/medrag/storage .
```

Профилактика БД

При снижении производительности выполнить:

bash

```
# ==========================================
# Оптимизация производительности БД (VACUUM ANALYZE)
# ==========================================
cd /opt/medrag
# 1. База medical-rag-main
docker compose exec postgres psql -U postgres -d main -c "VACUUM ANALYZE;"
# 2. База RAG-сервиса
docker compose exec postgres psql -U postgres -d rag  -c "VACUUM ANALYZE;"
```

Примечания

- Все пароли, ключи и IP-адреса в примерах — заглушки. Обязательно замените их на собственные.
- Для генерации надёжного пароля PostgreSQL: openssl rand -base64 16.
- Пароль PostgreSQL должен совпадать в трёх местах: POSTGRES_PASSWORD в docker-compose.yaml, Dsn в config-ext.toml сервисов medical-rag-main и medical-rag, а также в команде создания роли main (Шаг 5).
- Пароль Neo4j должен совпадать в NEO4J_AUTH (docker-compose.yaml) и в секции [Neo4j] конфигурации medical-rag.

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

- Все обращения к LLM идут только через comrade-router; прямой доступ сервисов к внешним моделям не требуется. Ключи ApiKey выдаёт администратор Comrade Router.
- Сервис medical-dataloom использует внешний сервис DoclingServe для распознавания документов — убедитесь в сетевой доступности указанного адреса.
- Внешние порты системы: 8081 (веб-интерфейс и API через Nginx) и 32658 (gRPC шлюза comrade-router).
- В продакшн-среде рекомендуется использовать политику restart: always и настроить мониторинг контейнеров.