ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

ООО «КОРТЕКС»

Руководство по развертыванию платформы бизнес-аналитики «Кортекс BI»

г. Краснодар, 2026

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

1. Введение

Настоящее руководство описывает порядок развертывания программного обеспечения платформы бизнес-аналитики «Кортекс BI» (далее — «Кортекс BI», система) на сервере под управлением операционной системы ALT Linux или Astra Linux.

«Кортекс BI» — клиент-серверное веб-приложение. Пользователи работают с системой через веб-браузер («тонкий клиент»), установка программ на рабочие места не требуется. Серверная часть состоит из трёх сервисов, каждый из которых выполняется в отдельном контейнере Docker:

- **shud-bi-frontend** — веб-сервер nginx, который раздаёт клиентское приложение (SPA) и проксирует запросы к программному интерфейсу (`/api/`) на серверное приложение;
- **shud-bi-backend** — серверное приложение (HTTP API): аутентификация, работа с датасетами, виджетами и дашбордами, выполнение запросов к источникам данных, кэширование агрегатов, журнал событий;
- **shud-bi-db** — собственная база данных системы (PostgreSQL 16 с расширением PostGIS 3.4). Сервис необязательный: вместо него может использоваться внешний сервер СУБД PostgreSQL или Postgres Pro.

Развертывание и управление сервисами выполняются средствами Docker Compose по единому файлу `docker-compose.yml`. Образы сервисов `shud-bi-frontend` и `shud-bi-backend` загружаются из приватного реестра образов правообладателя; учётные данные для доступа к реестру предоставляет правообладатель — ООО «Кортекс».

Технические имена `shud-bi-frontend`, `shud-bi-backend`, `shud-bi-db` и имя базы данных `shud_bi` являются служебными идентификаторами компонентов системы и не изменяются.

2. Состав системы

Сервис	Компоненты	Порты	База данных
shud-bi-frontend	Веб-сервер nginx, клиентское приложение (HTML, JavaScript, CSS)	80 внутри контейнера; публикуется на хосте на порт `BI_HTTP_PORT` (по умолчанию 8095)	—
shud-bi-backend	Серверное приложение (исполняемый файл `/bi_server`), встроенные SQL-миграции	8080 внутри сети Docker Compose; на хосте не публикуется	Подключается к `shud_bi` (чтение и запись) и к внешним источникам данных (только чтение)
shud-bi-db (профиль `embedded-db`, необязательный)	Сервер СУБД PostgreSQL 16 с расширением PostGIS	5432 внутри сети Docker Compose; на хосте не публикуется	`shud_bi` — собственная БД системы (схема `bi`),

Сервис	Компоненты	Порты	База данных
	3.4		данные хранятся в томе `bi-db-data`

Взаимодействие компонентов:

1. Браузер пользователя обращается к внешнему веб-серверу nginx по протоколу HTTPS (порт 443).
2. Внешний nginx передаёт запросы на опубликованный порт сервиса `shud-bi-frontend` (по умолчанию 8095).
3. `shud-bi-frontend` отдаёт файлы клиентского приложения, а запросы с путём `/api/` передаёт сервису `shud-bi-backend` (порт 8080) по внутренней сети Docker Compose.
4. `shud-bi-backend` хранит метаданные (датасеты, виджеты, дашборды, пользователей, настройки, кэш, журнал) в БД `shud_bi` и выполняет запросы на чтение к зарегистрированным внешним источникам данных PostgreSQL.

При первом и каждом последующем запуске серверное приложение автоматически применяет недостающие миграции структуры БД `shud_bi`. Применённые миграции фиксируются в таблице `bi.schema_migrations`.

3. Требования к серверу

3.1. Программное обеспечение

- Операционная система (64-разрядная, x86_64) — одна из российских ОС, с которыми совместимо ПО: ALT Linux (Альт Сервер 10 и новее), Astra Linux Special Edition (1.7 и новее), РЕД ОС (7.3 и новее).
- Docker Engine версии 24 и новее.
- Docker Compose версии 2 (команда `docker compose`).
- Веб-сервер nginx на хосте — для публикации системы по протоколу HTTPS (раздел 7).
- Для варианта с внешней БД — сервер СУБД PostgreSQL 16 или Postgres Pro (Standard / Enterprise) 16 и новее с расширением PostGIS 3.

3.2. Аппаратные ресурсы

Ресурс	Минимально	Рекомендуется
Процессор	2 ядра x86_64	4 ядра и более
Оперативная память	4 ГБ	8 ГБ и более
Дисковое пространство	20 ГБ	50 ГБ и более (SSD)

Дисковое пространство расходуется на образы контейнеров (около 2 ГБ), данные БД `shud_bi` и кэш агрегированных результатов. Предельный объём кэша задаётся

переменной `BI_CACHE_MAX_BYTES` (по умолчанию 4 ГиБ) и должен учитываться при расчёте размера диска.

3.3. Сетевые требования

- Входящие подключения пользователей: TCP 443 (HTTPS) и TCP 80 (перенаправление на HTTPS).
- Исходящий доступ с сервера к приватному реестру образов правообладателя — на время загрузки и обновления образов.
- Исходящий доступ с сервера к подключаемым источникам данных PostgreSQL (как правило, TCP 5432).
- Порты 8080 (серверное приложение) и 5432 (встроенная БД) извне не публикуются.

4. Подготовка сервера

Все команды раздела выполняются от имени пользователя root (или через `sudo`).

4.1. Установка Docker в ALT Linux

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

# ============================================
# 2. Включить автозапуск и запустить службу Docker
# ============================================
systemctl enable --now docker
```

4.2. Установка Docker в Astra Linux

```
# ============================================
# 1. Установить Docker Engine и Docker Compose v2
#    (имя пакета Compose зависит от версии репозитория:
#     docker-compose-plugin или docker-compose-v2)
# ============================================
apt update
apt install -y docker.io docker-compose-plugin

# ============================================
# 2. Включить автозапуск и запустить службу Docker
# ============================================
systemctl enable --now docker
```

4.3. Установка Docker в РЕД ОС

```
# ============================================
# 1. Установить Docker Engine и Docker Compose v2
#    из репозитория РЕД ОС
# ============================================
dnf install -y docker-ce docker-compose-plugin
```

```
# ============================================
# 2. Включить автозапуск и запустить службу Docker
# ============================================
systemctl enable --now docker
```

4.4. Проверка установки и создание каталога

```
# ============================================
# 1. Проверить версии Docker и Docker Compose
# ============================================
docker version
docker compose version

# ============================================
# 2. (Необязательно) разрешить работу с Docker
#    пользователю развертывания без sudo
# ============================================
usermod -aG docker <пользователь>

# ============================================
# 3. Создать каталог развертывания
# ============================================
mkdir -p /opt/kortex-bi
chmod 750 /opt/kortex-bi
cd /opt/kortex-bi
```

В каталоге `/opt/kortex-bi` размещаются два файла: `docker-compose.yml` (описание сервисов) и `.env` (параметры и секреты). Имя каталога определяет имя проекта Docker Compose (`kortex-bi`), поэтому том встроенной БД получит имя `kortex-bi_bi-db-data`.

5. Развертывание

5.1. Шаг 1. Создать файл docker-compose.yml

Создайте файл `/opt/kortex-bi/docker-compose.yml` следующего содержания. Вместо `<реестр_образов>` укажите адрес приватного реестра образов, вместо `<версия>` — номер поставляемой версии; оба значения сообщает правообладатель.

```
# ============================================
# Платформа бизнес-аналитики «Кортекс BI»
# /opt/kortex-bi/docker-compose.yml
# ============================================
services:
  # ------------------------------------------
  # Встроенная БД системы (PostgreSQL 16 + PostGIS 3.4).
  # Запускается только с профилем embedded-db.
  # Порт на хосте не публикуется: серверное приложение
  # обращается к БД по внутренней сети под именем shud-bi-db.
  # ------------------------------------------
  shud-bi-db:
```

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

```
    image: postgis/postgis:16-3.4
    container_name: shud-bi-db
    profiles: ["embedded-db"]
    restart: unless-stopped
    environment:
      POSTGRES_DB: shud_bi
      POSTGRES_USER: ${POSTGRES_USER:-bi_writer}
      # Обязателен при профиле embedded-db — задаётся в .env
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:-}
    volumes:
      # Постоянное хранилище данных БД
      - bi-db-data:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER:-bi_writer} -d shud_bi"]
      interval: 5s
      timeout: 5s
      retries: 12

  # ------------------------------------------
  # Серверное приложение (HTTP API).
  # Слушает порт 8080 только внутри сети Compose.
  # Миграции БД встроены в приложение и применяются при старте.
  # ------------------------------------------
  shud-bi-backend:
    image: <реестр_образов>/shud-bi-backend:<версия>
    container_name: shud-bi-backend
    restart: unless-stopped
    env_file: .env
    environment:
      BI_HTTP_ADDR: ":8080"
    expose:
      - "8080"

  # ------------------------------------------
  # Веб-сервер nginx: клиентское приложение + прокси /api/.
  # Публикуется на хосте на порт BI_HTTP_PORT (по умолчанию 8095).
  # ------------------------------------------
  shud-bi-frontend:
    image: <реестр_образов>/shud-bi-frontend:<версия>
    container_name: shud-bi-frontend
    restart: unless-stopped
    depends_on:
      - shud-bi-backend
    ports:
      - "${BI_HTTP_PORT:-8095}:80"

# ------------------------------------------
# Том встроенной БД. Сохраняется при «docker compose down»,
# удаляется только командой «docker compose down -v».
# ------------------------------------------
volumes:
  bi-db-data:
```

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

5.2. Шаг 2. Создать файл параметров .env

Создайте файл `/opt/kortex-bi/.env`. Файл содержит пароли и секретные ключи, поэтому доступ к нему должен быть ограничен.

Предварительно получите случайные значения секретов:

```
# ============================================
# Секреты для BI_JWT_SECRET и BI_SECRET_KEY (каждый — отдельной командой)
# ============================================
openssl rand -base64 32
openssl rand -base64 32

# ============================================
# Пароль пользователя БД (шестнадцатеричный — без символов,
# требующих экранирования в строке подключения)
# ============================================
openssl rand -hex 16
```

Содержимое файла `.env`:

```
# ============================================
# Публикация веб-интерфейса на хосте
# ============================================
# Порт хоста, на который публикуется shud-bi-frontend
BI_HTTP_PORT=8095

# ============================================
# Собственная БД системы shud_bi — ЗАМЕНИТЕ ПАРОЛИ НА СВОИ
# ============================================
# Вариант А (встроенная БД, профиль embedded-db):
# пароль в строке подключения и POSTGRES_PASSWORD должны совпадать
BI_WRITER_DSN=postgresql://bi_writer:замените_на_свой_пароль@shud-bi-db:5432/
shud_bi
POSTGRES_USER=bi_writer
POSTGRES_PASSWORD=замените_на_свой_пароль
# Вариант Б (внешний PostgreSQL): укажите адрес внешнего сервера,
# строки POSTGRES_USER и POSTGRES_PASSWORD удалите
#
BI_WRITER_DSN=postgresql://bi_writer:замените_на_свой_пароль@<адрес_сервера_БД>:5
432/shud_bi

# ============================================
# Интеграция с DataPlatform (необязательно)
# ============================================
# Оставьте пустым, если интеграция с DataPlatform не используется.
# Источники данных PostgreSQL регистрируются через веб-интерфейс.
BI_READER_DSN=

# ============================================
# Секреты — ОБЯЗАТЕЛЬНО ЗАМЕНИТЕ
# ============================================
# Ключ подписи токенов сеанса (openssl rand -base64 32)
BI_JWT_SECRET=замените_на_свой_секретный_ключ
# Ключ шифрования паролей источников данных, 32 байта
# (openssl rand -base64 32). Без него регистрация источников невозможна.
BI_SECRET_KEY=замените_на_свой_секретный_ключ
```

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

```
# ==========================================
# Учётная запись администратора
# ==========================================
# Создаётся при первом запуске, если пользователь с таким логином
# отсутствует. Последующее изменение значений существующую
# учётную запись не меняет.
BI_ADMIN_USER=admin
BI_ADMIN_PASSWORD=замените_на_свой_пароль

# ==========================================
# Кэш агрегированных результатов (необязательно)
# ==========================================
# Предельный объём кэша в байтах (по умолчанию 4 ГиБ)
BI_CACHE_MAX_BYTES=4294967296
# Срок хранения записи кэша (по умолчанию 2h)
BI_CACHE_TTL=2h
```

Ограничьте доступ к файлу:

```
chmod 600 /opt/kortex-bi/.env
```

Назначение переменных:

Переменная	**Обязательна**	**Назначение**
BI_HTTP_PORT	нет	Порт хоста для сервиса shud-bi-frontend (по умолчанию 8095)
BI_WRITER_DSN	да	Строка подключения к собственной БД `shud_bi` (роль с правом записи, владелец схемы `bi`)
POSTGRES_USER, POSTGRES_PASSWORD	только для варианта A	Учётная запись, создаваемая при первой инициализации встроенной БД
BI_READER_DSN	нет	Строка подключения интеграции с DataPlatform; пустое значение — коннектор не используется
BI_JWT_SECRET	да	Секретный ключ подписи токенов сеанса пользователей (при едином входе — общий ключ с внешней системой)
BI_SECRET_KEY	да (для регистрации источников)	Ключ шифрования (AES-256-GCM) паролей подключения к источникам данных; 32 байта в виде текста, base64 или hex
BI_ADMIN_USER, BI_ADMIN_PASSWORD	нет	Логин и пароль администратора, создаваемого при первом запуске (по умолчанию `admin` / `admin` — обязательно замените)

Переменная	Обязательна	Назначение
BI_CACHE_MAX_BYTES	нет	Предельный суммарный объём кэша, байт (по умолчанию 4294967296)
BI_CACHE_TTL	нет	Срок хранения записей кэша (по умолчанию `2h`)

Переменная `BI_HTTP_ADDR` (адрес прослушивания серверного приложения) задаётся в `docker-compose.yml` и в `.env` не указывается.

5.3. Шаг 3. Выбрать вариант размещения БД

Вариант А — встроенная БД (рекомендуется)

БД `shud_bi` работает в контейнере `shud-bi-db` на том же сервере. Все команды Docker Compose выполняются с ключом `--profile embedded-db`. При первом запуске контейнер создаёт БД `shud_bi` и пользователя из `POSTGRES_USER` / `POSTGRES_PASSWORD`; расширение PostGIS и схема `bi` создаются миграциями серверного приложения автоматически.

Вариант Б — внешний сервер PostgreSQL 16 или Postgres Pro с PostGIS

Команды Docker Compose выполняются без ключа `--profile`, контейнер `shud-bi-db` не запускается. Администратор внешнего сервера создаёт роль и БД (выполняется от имени суперпользователя PostgreSQL):

```
-- ============================================
-- Роль и БД системы «Кортекс BI»
-- ============================================
CREATE ROLE bi_writer LOGIN PASSWORD 'замените_на_свой_пароль';
CREATE DATABASE shud_bi OWNER bi_writer;

-- ============================================
-- Расширение PostGIS (требует прав суперпользователя)
-- ============================================
\c shud_bi
CREATE EXTENSION IF NOT EXISTS postgis;
```

Сервер PostgreSQL должен принимать подключения с сервера развертывания (параметр `listen_addresses` и запись в `pg_hba.conf`). В `.env` переменная `BI_WRITER_DSN` указывает адрес внешнего сервера (см. шаг 2).

5.4. Шаг 4. Загрузить образы и запустить систему

```
# ============================================
# 1. Авторизоваться в приватном реестре образов
#    (учётные данные предоставляет правообладатель)
# ============================================
docker login <реестр_образов>

# ============================================
# 2. Загрузить образы и запустить сервисы
```

```
# ============================================
cd /opt/kortex-bi

# Вариант А — встроенная БД
docker compose --profile embedded-db pull
docker compose --profile embedded-db up -d

# Вариант Б — внешний PostgreSQL
docker compose pull && docker compose up -d
```

При первом запуске в варианте А контейнер `shud-bi-db` несколько секунд инициализирует БД. Если серверное приложение запустится раньше, чем БД станет доступна, оно завершится с ошибкой и будет автоматически перезапущено (политика `restart: unless-stopped`); это штатное поведение.

5.5. Шаг 5. Проверить работоспособность

```
# ============================================
# 1. Состояние контейнеров (все — Up; shud-bi-db — healthy)
# ============================================
cd /opt/kortex-bi
docker compose --profile embedded-db ps

# ============================================
# 2. Применение миграций БД (журнал серверного приложения)
# ============================================
docker logs shud-bi-backend 2>&1 | grep "migrations applied"

# ============================================
# 3. Проверка работоспособности API
# ============================================
curl -s http://localhost:8095/api/health
```

Серверное приложение пишет журнал в стандартный вывод в формате JSON (одна запись — одна строка). После успешного запуска журнал содержит строки:

```
{"time":"…","level":"INFO","msg":"migrations applied"}
{"time":"…","level":"INFO","msg":"http listening","addr":":8080"}
```

Ответ проверки работоспособности:

- `{"status":"ok"}` (HTTP 200) — серверное приложение работает, БД `shud_bi` доступна;
- `{"status":"db down"}` (HTTP 503) — серверное приложение не может подключиться к БД `shud_bi`.

Откройте в браузере адрес системы (до настройки HTTPS — `http://<адрес_сервера>:8095/`) и войдите под учётной записью администратора из переменных `BI_ADMIN_USER` / `BI_ADMIN_PASSWORD`. После первого входа рекомендуется сменить пароль администратора: «Администрирование» → «Пользователи», действие «Сбросить пароль».

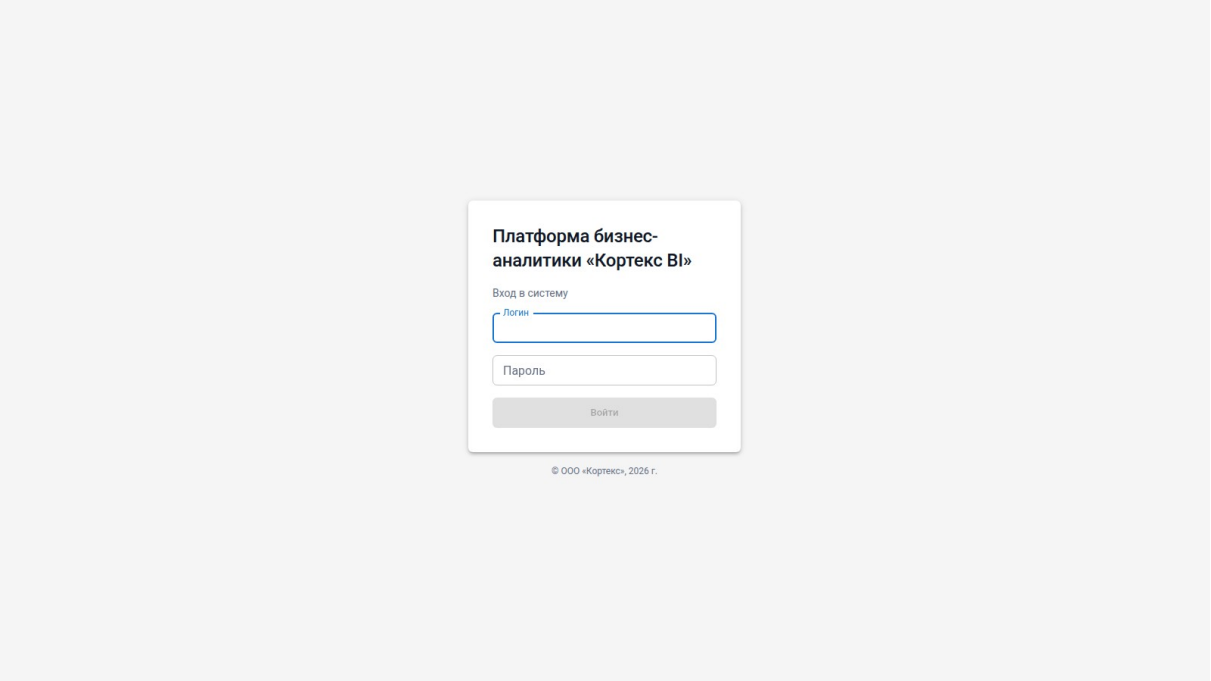


Рисунок 1 — Страница входа в систему

5.6. Шаг 6. Первичная настройка

Подключение источника данных

Для каждого источника данных рекомендуется создать в СУБД-источнике отдельную роль с правом только на чтение. Пример для PostgreSQL (выполняет администратор источника):

```
-- ============================================
-- Роль только для чтения для «Кортекс BI»
-- ============================================
CREATE ROLE bi_source_reader LOGIN PASSWORD 'замените_на_свой_пароль';
ALTER ROLE bi_source_reader SET default_transaction_read_only = on;

GRANT CONNECT ON DATABASE <база_источника> TO bi_source_reader;
\c <база_источника>
GRANT USAGE ON SCHEMA public TO bi_source_reader;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO bi_source_reader;
ALTER DEFAULT PRIVILEGES IN SCHEMA public
  GRANT SELECT ON TABLES TO bi_source_reader;
```

Регистрация источника в системе:

1. Войдите под учётной записью администратора.
2. Откройте раздел «Администрирование» → «Источники данных» и нажмите кнопку добавления источника.
3. Укажите наименование, хост, порт, имя базы данных, пользователя, пароль и режим SSL (`sslmode`).
4. Выполните проверку подключения и сохраните источник.
5. Убедитесь, что в карточке источника отображается список таблиц и колонок.

Пароль подключения хранится в БД `shud_bi` в зашифрованном виде (ключ `BI_SECRET_KEY`).

место для снимка экрана: Администрирование → Источники данных, форма добавления источника PostgreSQL с успешной проверкой подключения

Рисунок 2 — Регистрация источника данных

Настройка интерфейса

В разделе «Администрирование» → «Интерфейс» задаются название системы, цвета и логотипы. В разделах «Палитры» и «Темы дашбордов» настраиваются цветовые палитры виджетов и оформление дашбордов. В разделе «Пользователи» создаются учётные записи пользователей и назначаются права администратора.

6. Диагностика

В случае проблем выполните:

```
# ============================================
# Журналы сервисов (последние 50 строк)
# ============================================
# 1. Серверное приложение
docker logs shud-bi-backend --tail 50
# 2. Веб-сервер клиентского приложения
docker logs shud-bi-frontend --tail 50
# 3. Встроенная БД (вариант А)
docker logs shud-bi-db --tail 50

# ============================================
# Проверить, что контейнеры запущены
# ============================================
docker ps

# ============================================
# Процессы внутри контейнеров
# ============================================
docker top shud-bi-backend
docker top shud-bi-frontend
docker top shud-bi-db

# ============================================
# Состояние подключения к БД через API
# ============================================
curl -s http://localhost:8095/api/health
```

Администратор системы также может просмотреть состояние подключений к БД, версию и время работы серверного приложения в веб-интерфейсе: «Администрирование» → «Диагностика».

Наиболее частые проблемы:

Симптом	Причина	Решение
Контейнер shud-bi-backend постоянно перезапускается, в журнале `"msg":"fatal","err":"BI_WRITER_DSN is required"` или `"BI_JWT_SECRET is required"`	Не заданы обязательные переменные в `.env`	Заполнить `.env` и выполнить `docker compose up -d --force-recreate shud-bi-backend`
shud-bi-backend перезапускается, в журнале ошибка подключения (`connection refused`, `password authentication failed`)	БД ещё инициализируется либо пароль в `BI_WRITER_DSN` не совпадает с паролем пользователя БД	При первом запуске дождаться инициализации БД; сверить пароль в `BI_WRITER_DSN` и `POSTGRES_PASSWORD`
В журнале shud-bi-backend `BI_SECRET_KEY must be 32 bytes`	Ключ шифрования имеет неверную длину	Получить ключ командой `openssl rand -base64 32`, указать в `.env`, пересоздать контейнер
Контейнер shud-bi-db не запускается, в журнале сообщение об отсутствии пароля суперпользователя	Не задана переменная `POSTGRES_PASSWORD`	Задать `POSTGRES_PASSWORD` в `.env` и повторить запуск
Изменение `POSTGRES_PASSWORD` в `.env` не действует	Пароль применяется только при первой инициализации тома БД	Сменить пароль командой `ALTER ROLE` в БД и указать его в `BI_WRITER_DSN`
`docker compose pull` завершается ошибкой авторизации	Не выполнен вход в реестр образов	Выполнить `docker login <реестр_образов>` с учётными данными правообладателя
`/api/health` возвращает HTTP 503 `{"status":"db down"}`	Недоступна БД `shud_bi`	Проверить состояние shud-bi-db (`docker compose ps`) или доступность внешнего сервера БД
Браузер получает ошибку 502 на запросы `/api/`	Серверное приложение не запущено	Проверить `docker logs shud-bi-backend --tail 50`
При сохранении источника данных сообщение «BI_SECRET_KEY не настроен — регистрация источников с паролем невозможна»	Не задан ключ шифрования	Задать `BI_SECRET_KEY` в `.env`, выполнить `docker compose up -d --force-recreate shud-bi-backend`
Изменения в `.env` не применяются после `docker restart`	Команда `restart` не перечитывает файл `.env`	Пересоздать контейнер: `docker compose up -d --force-recreate shud-bi-backend`
Не выполняется вход с	Учётная запись	Войти с прежним паролем или

Симптом	Причина	Решение
паролем из `BI_ADMIN_PASSWORD`	администратора уже создана ранее с другим паролем	сбросить пароль из раздела «Администрирование» → «Пользователи» другим администратором
Ошибка 413 при импорте слоя регионов (GeoJSON)	Размер файла превышает ограничение веб-сервера	Увеличить `client_max_body_size` во внешнем nginx (раздел 7); файл GeoJSON рекомендуется предварительно упрощать до размера не более 1 МБ

7. Настройка Nginx с HTTPS

Для публикации системы по доменному имени с шифрованием трафика на хосте устанавливается веб-сервер nginx, который принимает подключения на порт 443 и передаёт их на порт сервиса `shud-bi-frontend`. Пример конфигурации (`/etc/nginx/conf.d/kortex-bi.conf`):

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

# ============================================
# 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;
    ssl_protocols       TLSv1.2 TLSv1.3;

    # Максимальный размер запроса (импорт слоёв регионов GeoJSON,
    # загрузка изображений для оформления)
    client_max_body_size 64m;

    # Все запросы (клиентское приложение и /api/) передаются
    # сервису shud-bi-frontend; он сам проксирует /api/ на бэкенд
    location / {
        proxy_pass http://127.0.0.1:8095;
        proxy_http_version 1.1;
        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;
```

```
        # Длительные аналитические запросы
        proxy_read_timeout 120s;
    }
}
```

Проверка и применение конфигурации:

```
nginx -t
systemctl reload nginx
```

При использовании внешнего nginx рекомендуется закрыть порт `BI_HTTP_PORT` (8095) для подключений извне средствами межсетевого экрана либо публиковать его только на локальном интерфейсе (в `docker-compose.yml`: `"127.0.0.1:${BI_HTTP_PORT:-8095}:80"`).

8. Резервное копирование

Подлежат резервному копированию: БД `shud_bi` (все объекты и настройки системы), том встроенной БД и файл `.env` (содержит ключ `BI_SECRET_KEY`, без которого сохранённые пароли источников данных не могут быть расшифрованы).

```
# ============================================
# Создать каталог для резервных копий
# ============================================
mkdir -p /backup

# ============================================
# 1. Дамп БД shud_bi (вариант А — встроенная БД)
# ============================================
docker exec shud-bi-db pg_dump -U bi_writer -d shud_bi -Fc \
  > /backup/shud_bi_$(date +%Y%m%d).dump

# ============================================
# 2. Архив тома встроенной БД (при остановленной БД)
# ============================================
cd /opt/kortex-bi
docker volume inspect kortex-bi_bi-db-data --format '{{ .Mountpoint }}'
docker compose --profile embedded-db stop shud-bi-db
tar -czf /backup/bi-db-data_$(date +%Y%m%d).tar.gz \
  -C /var/lib/docker/volumes/kortex-bi_bi-db-data/_data .
docker compose --profile embedded-db start shud-bi-db

# ============================================
# 3. Файлы конфигурации
# ============================================
tar -czf /backup/kortex-bi-config_$(date +%Y%m%d).tar.gz \
  -C /opt/kortex-bi docker-compose.yml .env
chmod 600 /backup/kortex-bi-config_*.tar.gz
```

Восстановление БД из дампа:

```
docker exec -i shud-bi-db pg_restore -U bi_writer -d shud_bi --clean --if-exists
\
  < /backup/shud_bi_ГГГГММДД.dump
```

```
docker compose --profile embedded-db restart shud-bi-backend
```

В варианте Б дамп выполняется утилитой `pg_dump` на внешнем сервере БД или с сервера развертывания по сети.

9. Профилактика БД

При снижении производительности выполните:

```
# ========================================
# Оптимизация БД shud_bi (VACUUM ANALYZE)
# ========================================
docker exec shud-bi-db psql -U bi_writer -d shud_bi -c "VACUUM ANALYZE;"
```

Устаревшие записи кэша агрегированных результатов удаляются серверным приложением автоматически по сроку хранения (`BI_CACHE_TTL`) и при достижении предельного объёма (`BI_CACHE_MAX_BYTES`). Принудительная очистка кэша выполняется администратором в веб-интерфейсе: «Администрирование» → «Диагностика».

10. Обновление

Для обновления системы укажите в `docker-compose.yml` номер новой версии образов (сообщает правообладатель) и выполните:

```
cd /opt/kortex-bi

# Рекомендуется предварительно выполнить резервное копирование (раздел 8)

# Вариант А — встроенная БД
docker compose --profile embedded-db pull
docker compose --profile embedded-db up -d

# Вариант Б — внешний PostgreSQL
docker compose pull && docker compose up -d
```

Новые миграции структуры БД применяются серверным приложением автоматически при запуске; данные в томе `bi-db-data` сохраняются. Результат обновления проверяется командами шага 5 (раздел 5.5).

11. Примечания

- Все пароли, адреса и секретные ключи в примерах — заглушки. Обязательно замените их на собственные значения.
- Для получения секретных ключей `BI_JWT_SECRET` и `BI_SECRET_KEY` используйте команду `openssl rand -base64 32`; для пароля БД — `openssl rand -hex 16`.

- Ключ `BI_SECRET_KEY` после регистрации источников данных не изменяется: при его замене сохранённые пароли подключения не будут расшифрованы, и источники потребуется зарегистрировать повторно.
- Файл `.env` содержит секреты: права доступа `600`, копия хранится только в защищённом резервном хранилище.
- Все сервисы запускаются с политикой перезапуска `restart: unless-stopped`: контейнеры автоматически запускаются после перезагрузки сервера и перезапускаются при сбое, если не были остановлены администратором вручную.
- Команда `docker compose down` сохраняет том БД; команда `docker compose down -v` удаляет том со всеми данными системы и без резервной копии не применяется.
- Серверное приложение выполняется в контейнере от имени непривилегированного пользователя; порты серверного приложения и БД извне не публикуются.