Инструкция по установке и эксплуатации

Продукт поставляется в виде набора образов Docker и файла запуска. Установка не требует доступа к сети разработчика и не содержит обращений «домой».

1. Состав поставки

Файл	Назначение
`kii-images.tar`	Образы: приложение `kii`, PostgreSQL 16, Redis 7
`docker-compose.yml`	Описание запуска четырёх контейнеров: `app`, `queue`, `db`, `redis`
`.env.example`	Шаблон настроек
`tools/docker-compose`	Плагин Docker Compose для серверов, где он не установлен (лицензия Apache-2.0)
`SHA256SUMS`	Контрольные суммы файлов поставки
`docs/`	Документация

2. Требования к серверу

Параметр	Значение
ОС	Linux общего назначения с поддержкой Docker (в том числе доверенные ОС)
Docker Engine	24 и выше, плагин Docker Compose v2
Процессор	от 2 ядер
Память	от 4 ГБ
Диск	от 20 ГБ (образы около 2 ГБ, остальное под данные и загружаемые файлы)
Сеть	Входящий порт веб-интерфейса (по умолчанию 8080). Исходящий доступ нужен только к выбранным провайдерам моделей (Яндекс, GigaChat)

Если используются только локальные модели внутри вашей сети, исходящий доступ в интернет не нужен.

3. Установка

1. Скопируйте архив на сервер, распакуйте и перейдите в каталог комплекта:

```
tar xzf kii-1.0.tar.gz && cd kii-1.0
sha256sum -c SHA256SUMS
```

2. Если команда `docker compose version` не работает, установите плагин из комплекта:

```
mkdir -p /usr/local/lib/docker/cli-plugins
cp tools/docker-compose /usr/local/lib/docker/cli-plugins/docker-compose
chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
docker compose version
```

   Если на сервере запрещена запись в `/usr/local` (например, при усиленной защите ОС), используйте каталог комплекта:

```
mkdir -p ~/.docker-kii/cli-plugins && cp tools/docker-compose ~/.docker-kii/cli-plugins/
export DOCKER_CONFIG=~/.docker-kii
docker compose version
```

   Если запись запрещена и в домашний каталог, подойдёт любой каталог с правом записи (например `/var/tmp`). Переменную `DOCKER_CONFIG` нужно задавать в каждой новой сессии. Распаковывайте комплект тоже в каталог с правом записи.
3. Загрузите образы:

```
docker load -i kii-images.tar
```

4. Создайте файл настроек:

```
cp .env.example .env
```

5. Сгенерируйте ключ приложения и впишите его в `.env` (значение целиком, включая `base64:`):

```
docker run --rm kii:1.0 php artisan key:generate --show
```

6. В `.env` задайте пароль базы данных (`DB_PASSWORD`), адрес системы (`APP_URL`), порт (`HTTP_PORT`) и учётную запись первого пользователя — суперадминистратора, который затем создаёт подразделения (`ADMIN_EMAIL`, `ADMIN_INITIAL_PASSWORD`).
7. Запустите:

```
docker compose up -d
```

   При первом запуске приложение само создаёт структуру базы данных и начальные данные. Это занимает до минуты.
8. Откройте адрес из `APP_URL` и войдите под суперадминистратором и создайте подразделения (раздел «Подразделения»).

Если `ADMIN_INITIAL_PASSWORD` не задан, пароль генерируется и выводится в журнал контейнера один раз:

```
docker compose logs app | grep "Начальный пароль"
```

Смените пароль после первого входа.

Ключ приложения `APP_KEY` нужно сохранить отдельно. Без него расшифровка сессий и защищённых данных невозможна.

4. Доступ по HTTPS

Контейнер `app` слушает HTTP на порту 80 (проброшен на `HTTP_PORT`). Для HTTPS разместите перед ним обратный прокси вашей организации (nginx, Apache, Traefik и подобные) и укажите в `APP_URL` адрес с `https://`. Прокси должен передавать заголовки `X-Forwarded-For`, `X-Forwarded-Host`, `X-Forwarded-Port`, `X-Forwarded-Proto`.

5. Подключение моделей

После установки провайдеров и моделей в системе нет: пользователям недоступна ни одна модель, пока суперадминистратор не подключит провайдера и не добавит конкретную модель. Всё делается в веб-интерфейсе: раздел «AI Модели» (`/admin/ai-models`).

Порядок действий

1. Вкладка провайдеров, «+ Добавить провайдера»: укажите название, тип подключения и параметры.

Тип подключения	Что указать
Локальная модель (сервер внутри контура)	Базовый URL сервера, совместимого с протоколом `/v1/chat/completions`, и при необходимости ключ
Яндекс GPT	API-ключ и идентификатор каталога (`folder_id`) Yandex Cloud
GigaChat	Ключ авторизации (Base64) и область доступа (`scope`)

2. Вкладка моделей, «+ Добавить модель»: выберите провайдера и заполните поля.
    - «Название модели» — как модель будет называться у пользователей;
    - «Название для API» — уникальное служебное имя в системе;
    - «Имя модели у провайдера (если отличается)» — идентификатор модели на стороне провайдера;
    - «Контекстное окно (токены)»;
    - стоимость входа и выхода в рублях за 1 млн токенов (для собственного сервера можно оставить нули);

    - отметки о поддержке изображений и видео, активность и «по умолчанию».
3. Отправьте тестовое сообщение в чате, выбрав добавленную модель.

Пока нет ни одной активной модели, чат отвечать не будет. Если нужно ограничить модели для подразделения или пользователя, это делается в карточках подразделения и пользователя.

Локальная модель

1. Убедитесь, что сервер модели доступен с сервера, где установлена система, и отвечает на `GET <адрес>/v1/models`.
2. Если адрес сервера модели не относится к частным сетям (10.x, 172.16–31.x, 192.168.x), localhost, коротким именам и `*.local`, `*.lan`, `*.internal`, добавьте его в `.env`:

   ```
   LOCAL_MODEL_ALLOWED_HOSTS=model.example.org
   ```

   и примените: `docker compose up -d`. Без этого адрес будет отклонён при сохранении.
3. Создайте провайдера типа «Локальная модель» с «Базовым URL», например `http://10.0.0.5:8000/v1`. Ключ укажите, только если сервер требует авторизацию.
4. Добавьте модель этого провайдера; в поле «Имя модели у провайдера» укажите идентификатор из ответа `/v1/models` (например `my-model`).

Можно добавить несколько провайдеров и моделей. Всё, что добавлено администратором, при обновлении системы сохраняется.

Для проверки TLS GigaChat в образ встроены корневой и выпускающий сертификаты УЦ Минцифры.

6. Транскрибация

По умолчанию используется локальная модель распознавания речи. В разделе «Оформление» (`/admin/branding`) выберите локального провайдера и при необходимости имя модели. Сервер модели должен поддерживать `POST /audio/transcriptions`. До выбора провайдера загрузка аудио завершается понятной ошибкой, во внешние сервисы ничего не отправляется.

Внешний движок `speech2text` включается там же и требует ключ `SPEECH2TEXT_API_KEY` в `.env`.

7. Почта

Для восстановления пароля задайте `MAIL_MAILER=smtp` и параметры `MAIL_HOST`, `MAIL_PORT`, `MAIL_USERNAME`, `MAIL_PASSWORD`, `MAIL_FROM_ADDRESS`. По умолчанию письма записываются в журнал контейнера.

8. Эксплуатация

Действие	Команда
Состояние	`docker compose ps`
Журналы	`docker compose logs -f app queue`
Перезапуск	`docker compose restart`
Остановка	`docker compose down` (данные сохраняются в томах)

Резервное копирование

Данные хранятся в трёх томах Docker: `db_data` (база), `app_public_storage` и `app_private_storage` (файлы).

```
# База данных
docker compose exec -T db pg_dump -U kii -Fc kii > kii-$(date +%F).dump
# Файлы
docker run --rm -v <проект>_app_public_storage:/d -v "$PWD":/b alpine tar czf /b/public-$
(date +%F).tgz -C /d .
docker run --rm -v <проект>_app_private_storage:/d -v "$PWD":/b alpine tar czf
/b/private-$(date +%F).tgz -C /d .
```

Имя проекта — имя каталога, в котором лежит `docker-compose.yml`.

Восстановление базы:

```
docker compose exec -T db pg_restore -U kii -d kii --clean --if-exists < kii-ГГГГ-ММ-
ДД.dump
```

Обновление

1. Сделайте резервную копию.
2. Загрузите новые образы: `docker load -i kii-images.tar`.
3. Укажите новую версию в `.env` (`KII_VERSION=...`) и выполните `docker compose up -d`.

Структура базы обновляется автоматически при запуске.

Удаление

```
docker compose down -v
```

Команда удаляет контейнеры и все данные. Выполняйте только после резервного копирования.

9. Диагностика

Симптом	Что проверить
Контейнер app перезапускается	`docker compose logs app`: чаще всего не задан APP_KEY или DB_PASSWORD
Страница недоступна	`docker compose ps`: контейнер app должен быть в состоянии `healthy`
Модель не отвечает	Раздел «AI Модели»: ключ провайдера, исходящий доступ к адресу провайдера
Ошибка адреса локальной модели	Адрес вне внутренней сети: добавьте хост в LOCAL_MODEL_ALLOWED_HOSTS
Транскрибация не стартует	Выбран ли локальный провайдер в «Оформлении», запущен ли контейнер `queue`