7(4712) 26-99-00
www.dcatk.pro
info@dcatk.pro

Инструкция по установке экземпляра программного обеспечения, предоставленного для проведения экспертной проверки

Инструкция для одного Linux-сервера с Docker Compose и nginx на хосте. Compose запускает веб-интерфейс, API, отдельный воркер сборки записей, PostgreSQL, Redis и S3. nginx принимает HTTPS и проксирует запросы к сервисам, чьи порты на хосте опубликованы только на 127.0.0.1. PostgreSQL и Redis не публикуют порты на хосте.

1. Подготовить сервер

Нужны Docker Engine с плагином docker compose, nginx, DNS-запись для сервера и действующий TLS-сертификат. Откройте снаружи HTTPS-порт 443. Клонируйте репозиторий и выполняйте команды из его корня.

Пример ниже использует домен meeting.example.com; замените его своим. Установите конфигурацию nginx (nginx/ai-meeting-room.conf) в контекст http (например, через sites-available/sites-enabled на Debian/Ubuntu), исправьте server_name, ssl_certificate и ssl_certificate_key. Файл содержит только сервер на порту 443: получение сертификата и перенаправление HTTP на HTTPS настройте отдельно, если они нужны.

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

2. Заполнить настройки

```
cp deploy/compose.env.example .env
chmod 600 .env
```

В корневом .env заполните все поля из шаблона:

Переменная	Значение
PUBLIC_ORIGIN	Внешний адрес, например https://meeting.example.com, без завершающего /
POSTGRES_PASSWORD	Пароль пользователя PostgreSQL meeting
DATABASE_URL	postgresql://meeting:<тот_же_пароль>@postgres:5432/meeting_room?schema=public
S3_ACCESS_KEY, S3_SECRET_KEY	Учётные данные S3-хранилища (RustFS); задайте отдельные надёжные значения
STT_TASK_URL	Адрес LinTO task API, доступный из контейнера backend, без суффикса /api; пустое значение включает тестовую транскрипцию
STT_TASK_TOKEN, STT_TASK_TOKEN_HEADER	Токен и имя заголовка авторизации STT; по умолчанию заголовок Authorization

Переменная	Значение
STT_TASK_LANGUAGE, STT_TASK_PROVIDER_VERSION	Необязательные язык и версия STT-провайдера
LLM_BASE_URL	Адрес OpenAI-совместимого API LiteLLM, например https://gateway.example.com/v1, доступный из контейнера backend; пустое значение включает тестовое резюме
LLM_MODEL, LLM_API_KEY	Имя модели в LiteLLM и ключ доступа к шлюзу
JWT_ACCESS_SECRET, JWT_REFRESH_SECRET	Два разных длинных случайных секрета
INTEGRATION_ENCRYPTION_KEY, DEVICE_TOKEN_SECRET, VOICE_PROCESSING_WEBHOOK_SECRET	Еще три независимых случайных секрета

Для генерации секретов можно использовать openssl rand -hex 32 отдельно для каждого поля. Если пароль БД содержит специальные символы URL, закодируйте их в DATABASE_URL. Для контейнеров нужен адрес БД postgres, а не localhost. Файл backend/.env.example предназначен для запуска API без Compose; значения для текущего Compose-стека берутся из корневого .env.

Проверьте подстановку переменных, не выводя конфигурацию с секретами в журнал:

```
docker compose config -q
```

3. Запустить стек

```
docker compose up -d postgres redis rustfs
docker compose build backend recording-worker frontend
docker compose run --rm --no-deps backend npm run prisma:deploy
docker compose up -d --no-build backend recording-worker frontend
docker compose ps -a
docker compose logs --tail=100 backend recording-worker frontend
```

При старте API автоматически создаёт бакет ai-meeting-room, если его ещё нет. Миграции применяются до запуска API; команды запуска контейнеров backend и recording-worker также проверяют их повторно. Убедитесь, что оба процесса работают и миграции завершились без ошибок.

Откройте https://meeting.example.com/setup и создайте первого системного администратора. Этот маршрут доступен только пока администратора нет. Проверьте веб-интерфейс по /, API через https://meeting.example.com/api/setup/status и Swagger через /api/docs. Для проверки ответа API:

```
curl -fsS https://meeting.example.com/api/setup/status
```

После входа проверьте создание конференции и загрузку/воспроизведение записи: это подтверждает работу PostgreSQL, Redis, воркера, S3 и nginx-маршрута /ai-meeting-room/ для подписанных файлов.

4. Обновлять и обслуживать

Перед обновлением сохраните резервные копии БД, объектов S3 и данных загрузки записей. Для восстановления фоновых заданий сохраните также Redis. Docker-тома называются postgres_data, rustfs_data, recording_uploads и redis_data внутри Compose-проекта. Не используйте docker compose down -v при обычном обновлении: эта команда удаляет тома.

При обновлении старой БД до миграции, удаляющей роль OWNER, выполните запрос SELECT count(*) FROM "User" WHERE role::text = 'OWNER';. Если результат больше нуля, сначала переведите этих пользователей на поддерживаемую роль. Изменение POSTGRES_PASSWORD в .env не меняет пароль в уже созданном PostgreSQL-томе; пароль в БД и DATABASE_URL должен совпадать.

```
docker compose build backend recording-worker frontend
docker compose stop frontend backend recording-worker
docker compose run --rm --no-deps backend npm run prisma:deploy
docker compose up -d --no-build backend recording-worker frontend
docker compose ps -a
```

Команда run применяет миграции до перезапуска API. При изменении конфигурации nginx дополнительно выполните sudo nginx -t && sudo systemctl reload nginx. Для диагностики:

```
docker compose logs --tail=200 backend recording-worker
docker compose ps -a
```

Дополнительные компоненты

Room Agent устанавливается отдельно на устройство в переговорной. В его Local Admin укажите URL API https://meeting.example.com/api, затем выполните привязку устройства в панели администратора.

Keycloak, сервис диаризации, NATS, внешний STT, LiteLLM и vox2pg в этом Compose-файле не запускаются. Адреса и ключи STT и LiteLLM задаются в корневом .env и передаются контейнеру backend, где работает воркер обработки. Используйте адреса, доступные из контейнера: localhost внутри него не указывает на хостовый сервис. При пустых STT_TASK_URL и LLM_BASE_URL используются тестовые провайдеры. Для остальных внешних служб также нужны отдельное развёртывание и соответствующие

настройки API. Если включаете vox2pg, отдельно примонтируйте закрытый RSA-ключ только для чтения по пути VOICE_PROCESSING_PRIVATE_KEY_PATH: текущий Compose-файл ключ не монтирует.