ЛИИС Офис Эксплуатация

Руководство системного администратора

Версия 0.2.2

13 сентября 2026

Документация системы учёта оборудования и планирования технического обслуживания.
simplecare.liis.su

Содержание

Общие сведения о системе

Структура системы

Требования к серверу

Настройка системы

- Установка через Portainer
- Переменные стека
- Работа за HTTPS
- Резервное копирование и восстановление
- Обновление версии

Проверка системы

Дополнительные возможности

Сообщения системному администратору

Общие сведения о системе

«ЛИИС Офис Эксплуатация» — веб-система календарного обслуживания оборудования. Поставляется как набор готовых Docker-образов и файл описания стека для Docker Compose или Portainer. Backend — Python/FastAPI, база данных — MySQL 8, веб-интерфейс — React, раздаётся Nginx. Backend отдаёт метрики в формате Prometheus на `/metrics` внутри сети стека — подключение сборщика метрик (Prometheus, VictoriaMetrics и т. п.) выполняется силами заказчика.

Наружу публикуется один порт веб-интерфейса (по умолчанию 3000); порты базы данных и backend за пределы сети стека не публикуются. Штатная схема эксплуатации — за обратным прокси заказчика с TLS (см. «Работа за HTTPS»); публикация интерфейса по обычному HTTP допустима только в доверенной сети на время пусконаладки.

Структура системы

Состав сервисов стека:

Сервис	Назначение
`mysql`	MySQL 8.4; данные — в именованном томе `mysql_data`. Образ входит в поставку.
`migrate`	Одноразовый сервис: миграции схемы (Alembic) и создание встроенной учётной записи `superadmin`. Одноразовый пароль `BOOTSTRAP_SUPERADMIN_PASSWORD` доступен только ему. Штатно завершается со статусом `exited (0)`.
`backend`	FastAPI-приложение: бизнес-API `/api/*`, `/health`, `/ready`, `/metrics`. Работает без прав root. Загруженные документы — в томе `uploads_data`.
`frontend`	Nginx без прав root (порт 8080 внутри контейнера): веб-интерфейс, проксирование `/api/*`, `/health` и `/ready` на backend, статика руководств `/manuals/*.docx`.

Все контейнеры стека запускаются без права повышения привилегий (`no-new-privileges`) и, кроме MySQL (его штатная инициализация их использует), без Linux capabilities (`cap_drop: [ALL]`) — это задано в `compose.yaml`, отдельной настройки не требует, и при переносе стека в другой файл описания эти строки убирать не следует.

Доступ к бизнес-API требует аутентификации (session cookie); весь ввод-вывод времени — UTC внутри, московское время на экранах. Документы и база образуют одну логическую резервную копию. База данных после установки пустая: справочники и оборудование заводит администратор системы.

Требования к серверу

Система поставляется набором Docker-образов и запускается через Docker Compose или Portainer. Для развёртывания достаточно виртуальной машины Linux x86_64 (образы поставляются для `linux/amd64`) с Docker Engine 24+ и Compose v2.20+ либо Portainer CE 2.19+

на Standalone Docker Engine (окружение Docker, не Swarm: в Swarm-стеке порядок запуска сервисов и политика перезапуска не действуют). Отдельные серверы под СУБД не требуются — MySQL входит в состав стека. Доступ сервера в интернет и к реестрам образов не требуется: все три образа (backend, frontend, MySQL) входят в поставку.

Профиль	CPU	Память	Диск
Минимум (пилот, до 10 пользователей)	1 vCPU	2 ГБ	20 ГБ
Комфортно (рабочая инсталляция)	2 vCPU	4 ГБ	от 40 ГБ

Ориентиры фактического потребления: MySQL — до 0,5 ГБ памяти, backend — около 0,1 ГБ, frontend — десятки мегабайт; остальное — запас на пики и рост данных. Диск планируйте от объёма загружаемых документов: один файл может занимать до 50 МБ, документы хранятся в томе `uploads_data`, база — в томе `mysql_data`; оба тома должны попадать в резервное копирование.

Сетевые требования: один входящий HTTP-порт веб-интерфейса (за обратным прокси с TLS либо выбранный порт); исходящий доступ к SMTP-серверу (обычно порт 465 или 587), если включена отправка почты; исходящий HTTPS до сервера ADFS, если включён корпоративный вход. Порты backend и базы данных наружу не публикуются. Пользователям достаточно актуального браузера (Chrome, Edge, Firefox, Safari); интерфейс адаптирован и для телефона.

Настройка системы

Установка через Portainer

Поставка — каталог `simple-office-care-<версия>` с файлами `images.tar.gz` (три образа: backend, frontend, MySQL), `compose.yaml`, `.env.example`, `RELEASE.txt`, `SHA256SUMS` и этим руководством. Сверьте контрольные суммы: `sha256sum -c SHA256SUMS` — это защита от повреждения при передаче; подлинность поставки подтверждайте сверкой значений из `SHA256SUMS` с поставщиком по независимому каналу. Установка не требует доступа сервера в интернет.

1. Загрузите образы на сервер: **Images** → **Import** в Portainer либо `docker load -i images.tar.gz` на хосте. В списке образов появятся `simple-office-care/backend:<версия>`, `simple-office-care/frontend:<версия>` и `mysql:8.4.<N>` (точный тег указан в `RELEASE.txt`).
2. **Stacks** → **Add stack** → **Web editor**: вставьте содержимое `compose.yaml` без изменений.
3. В блоке **Environment variables** нажмите «Load variables from .env file» и загрузите заполненную копию `.env.example` либо задайте переменные вручную (см. «Переменные стека»). Обязательные: `MYSQL_PASSWORD`, `MYSQL_ROOT_PASSWORD`, `SETTINGS_ENCRYPTION_KEY`; для первого запуска — `BOOTSTRAP_SUPERADMIN_PASSWORD`. Если система будет доступна не только с этого сервера, сразу задайте `PUBLIC_BASE_URL`, `AUTH_COOKIE_SECURE=true` и `UI_BIND=127.0.0.1` по разделу «Работа за HTTPS» — значения по умолчанию рассчитаны на пусконаладку в доверенной сети.
4. **Deploy the stack**. Сервис `migrate` выполнит миграции, создаст учётную запись `superadmin` и завершится со статусом `exited (0)` — это штатно; `backend` переходит в `healthy`, `frontend` — в `running`. Если `migrate` завершился с другим кодом — откройте его журнал (Containers

→ `<стек>-migrate-1` → Logs): типичная причина при первом запуске — не задан `BOOTSTRAP_SUPERADMIN_PASSWORD`.

5. Откройте `http://<сервер>:3000`, войдите под `superadmin` с одноразовым паролем — система потребует сменить его. После смены удалите `BOOTSTRAP_SUPERADMIN_PASSWORD` из переменных стека и нажмите **Update the stack** — **не включая** опцию повторного скачивания образов (Re-pull image): образы локальные, в реестрах их нет, и попытка скачать завершит обновление ошибкой. Повторный запуск `migrate` существующую учётную запись не меняет.

После загрузки переменных в Portainer заполненный файл `.env` на сервере больше не нужен: удалите его или храните с правами `600` вне общедоступных каталогов. Значения переменных стека видны каждому, у кого есть доступ к Portainer или к `docker inspect` на этом сервере.

Стек не запустится, если не заданы обязательные переменные: Portainer покажет сообщение вида `required variable MYSQL_PASSWORD is missing` — заполните переменную и повторите развёртывание.

Без Portainer та же поставка разворачивается командами Docker Compose на хосте: `docker load -i images.tar.gz`, затем в каталоге с `compose.yaml` и заполненным `.env` — `docker compose up -d`.

Переменные стека

Переменная	Обязательна	Назначение
`MYSQL_PASSWORD`, `MYSQL_ROOT_PASSWORD`	да	Пароли БД: латиница и цифры, 24+ символов. Задаются один раз при создании стека; смена после инициализации требует смены пароля в самой MySQL.
`SETTINGS_ENCRYPTION_KEY`	да	Ключ Fernet для секретов раздела «Система». Восстановление копии БД требует того же ключа.
`BOOTSTRAP_SUPERADMIN_PASSWORD`	при первом запуске	Одноразовый пароль `superadmin`; без него первый запуск завершится ошибкой сервиса `migrate`. Удалить после смены пароля.
`PUBLIC_BASE_URL`	рекомендуется	Внешний адрес интерфейса без хвостового слэша (`https://care.example.com`): ссылки в письмах.
`AUTH_COOKIE_SECURE`	за HTTPS — `true`	Флаг Secure у cookie сессии. При `true` вход по обычному `http://` невозможен.
`UI_BIND`, `UI_PORT`	нет	Интерфейс публикуется на `UI_BIND:UI_PORT` (по умолчанию `0.0.0.0:3000`); за обратным прокси задайте `UI_BIND=127.0.0.1`.
`AUTH_SESSION_TTL_MINUTES`	нет	Срок жизни сессии, по умолчанию 480.
`OIDC_REDIRECT_URI`, `OIDC_POST_LOGIN_URL`	при корпоративном входе	`https://<имя>/api/auth/oidc/callback` и `https://<имя>/`, где `<имя>` — хост из `PUBLIC_BASE_URL`. Без корпоративного входа не задаются.
`SMTP_*`, `NOTIFICATION_RECIPIENT`	нет	Почта; те же параметры доступны в разделе «Система» и имеют приоритет.

Порты БД и backend наружу не публикуются. Метрики backend (`/metrics`, формат Prometheus) доступны только контейнерам внутри сети стека по адресу `http://backend:8000/metrics` — с хоста этот адрес не открывается. Чтобы их собирать, подключите свой сборщик к сети `<имя стека>_default` (например, `docker network connect <имя стека>_default <контейнер сборщика>`).

Работа за HTTPS

Система не терминирует TLS: перед ней ставится обратный прокси заказчика (nginx, Traefik, Caddy), который проксирует весь трафик на `UI_BIND:UI_PORT`. Требования: `PUBLIC_BASE_URL=https://<имя>`, `AUTH_COOKIE_SECURE=true`, `UI_BIND=127.0.0.1` (интерфейс не виден мимо прокси), лимит тела запроса на прокси не меньше 51 МБ (загрузка документов до 50 МБ), таймаут проксирования не меньше 60 с. Для корпоративного входа (ADFS/OIDC) задайте `OIDC_REDIRECT_URI=https://<имя>/api/auth/oidc/callback` и `OIDC_POST_LOGIN_URL=https://<имя>/` и зарегистрируйте тот же redirect URI у провайдера. Пока `AUTH_COOKIE_SECURE=false`, а интерфейс опубликован на `0.0.0.0`, сессии пользователей передаются по сети открытым текстом — такой режим допустим только на время пусконаладки.

Резервное копирование и восстановление

Копия состоит из дампа MySQL и архива тома документов, снятых совместно при остановленном backend (на это время система недоступна — планируйте окно). Команды выполняются в `bash` на хосте с Docker (`<стек>` — имя стека в Portainer, тома называются `<стек>_mysql_data` и `<стек>_uploads_data`; `<версия>` — тег образов из `RELEASE.txt`). Пароль БД вводится по запросу без отображения символов (это нормально) и передаётся в контейнер по имени переменной (`-e MYSQL_PWD` без значения) — так он не попадает ни в историю команд, ни в список процессов; `umask 077` создаёт файлы копии сразу с правами `0600`. Дамп снимается с `--databases … --add-drop-database`, чтобы восстановление приводило базу к состоянию копии целиком (таблицы более новой версии не остаются); `--no-tablespaces` обязателен — у пользователя `toir` нет привилегии PROCESS. Архив тома документов снимает служебный контейнер из образа backend, входящего в поставку (других образов на сервере может не быть): `tar` внутри контейнера пишет в файл на хосте, затем файл сжимается — без конвейера через `|`, чтобы падение контейнера (например, опечатка в `<версия>`) не оставило правдоподобный, но пустой архив.

```
umask 077
read -rsp 'Пароль MySQL (MYSQL_PASSWORD): ' MYSQL_PWD; echo; export MYSQL_PWD
docker stop <стек>-backend-1
docker exec -e MYSQL_PWD <стек>-mysql-1 mysqldump -utoir --single-transaction --routines --no-
tablespaces --databases toir --add-drop-database > backup-<дата>.sql
docker run --rm -v <стек>_uploads_data:/data:ro simple-office-care/backend:<версия> tar cf - -C
/data . > uploads-<дата>.tar && gzip uploads-<дата>.tar
sha256sum backup-<дата>.sql uploads-<дата>.tar.gz > backup-<дата>.sha256
docker start <стек>-backend-1
unset MYSQL_PWD
```

Перед тем как убрать копию на хранение, убедитесь, что `backup-<дата>.sql` непустой (`ls -l`), а архив читается: `tar tzf uploads-<дата>.tar.gz | head`.

Храните копии вместе с копией значения `SETTINGS_ENCRYPTION_KEY` (в менеджере секретов, не рядом с дампом): без него система после восстановления не пройдёт проверку готовности (`/ready` отвечает `settings_key_invalid`), а сохранённые OIDC/SMTP-секреты придётся ввести заново. Дамп содержит хэши паролей и зашифрованные секреты — права `0600`, доступ только администратору.

Восстановление — на остановленном backend, в том же порядке. Сначала проверки копии (контрольные суммы, целостность и читаемость архива): **если любая из них сообщила об ошибке — остановитесь и не выполняйте следующие команды**, копия повреждена. Команды сцеплены через `&&`, поэтому при построчном вводе после ошибки они не выполнятся, но проверьте это глазами — текущие документы стираются только после всех проверок:

```
read -rsp 'Пароль MySQL (MYSQL_PASSWORD): ' MYSQL_PWD; echo; export MYSQL_PWD
sha256sum -c backup-<дата>.sha256 && gzip -t uploads-<дата>.tar.gz && tar tzf uploads-
<дата>.tar.gz >/dev/null \
 && docker stop <стек>-backend-1 \
 && docker exec -i -e MYSQL_PWD <стек>-mysql-1 mysql -utoir < backup-<дата>.sql \
 && gzip -dc uploads-<дата>.tar.gz | docker run --rm -i --user 0 -v <стек>_uploads_data:/data
simple-office-care/backend:<версия> sh -c 'rm -rf /data/* /data/.[!.]* && tar xf - -C /data &&
chown -R 10001:10001 /data' \
 && docker start <стек>-backend-1
unset MYSQL_PWD
```

Восстановление приводит базу и документы к состоянию копии полностью; частичное восстановление поверх более новой схемы не поддерживается. Если импорт дампа отвечает `Access denied … to database 'toir'`, повторите команду импорта от имени `root` (`mysql -uroot` с паролем `MYSQL_ROOT_PASSWORD`, введённым тем же способом через `read -rsp` в `MYSQL_PWD`), а затем **обязательно выполните оставшиеся две команды цепочки** — восстановление тома документов и `docker start <стек>-backend-1`: иначе база будет из копии, а документы — прежние. После восстановления проверьте `http://<сервер>:3000/ready` (ожидается `200`) и вход. Регулярность копий и место хранения — по политике заказчика; рекомендуемый минимум — ежедневно с хранением 14 дней.

Обновление версии

1. Снимите резервную копию (раздел выше) и сверьте контрольные суммы новой поставки.
2. Загрузите новые образы (`docker load -i images.tar.gz` или Portainer → Images → Import).
3. В редакторе стека замените `compose.yaml` содержимым из новой поставки (в нём новые теги образов), переменные стека сохраняются; нажмите **Update the stack**, не включая опцию повторного скачивания образов (Re-pull image) — образы уже загружены локально.
4. `migrate` применит миграции схемы автоматически и идемпотентно; после `healthy` у backend проверьте `/ready` и вход.

Миграции схемы выполняются только вперёд: обратной миграции нет. Откат — это восстановление резервной копии, снятой на шаге 1, и возврат прежнего `compose.yaml` с прежними тегами (образы предыдущей версии остаются на сервере, пока их не удалили). Не запускайте новую версию поверх БД, восстановленной из копии более новой версии.

Системная конфигурация (раздел «Система» в интерфейсе, доступен только роли «Системный администратор»):

- **Корпоративный вход (ADFS/OIDC)**: адреса провайдера (только `https://`), идентификатор клиента и секрет. Секреты после сохранения не показываются — только статус «Настроено/Не настроено»; заменить можно, увидеть заново — нельзя; кнопка «Очистить» полностью снимает сохранённый секрет. Кнопка «Корпоративная учётная запись» на странице входа появляется только когда корпоративный вход включён и заполнены все адреса провайдера и идентификатор клиента; до этого пользователи видят только локальный вход. Корпоративные учётные записи создаются при первом входе (JIT), регулярной синхронизации с AD/LDAP нет. Адреса возврата задаются переменными стека `OIDC_REDIRECT_URI` и `OIDC_POST_LOGIN_URL` (см. «Работа за HTTPS»).
- **Сопоставление групп каталога** (AD-mappings): группа каталога → роль системы. Пользователи корпоративного входа получают роли по своим группам при каждом входе; системная роль через группы не выдаётся. Удаление сопоставления означает, что пользователи этой группы при следующем входе потеряют доступ.
- **Почта (SMTP)**: сервер, порт, отправитель, учётные данные; включается отдельным флагом. По умолчанию отправка выключена. Настройки из интерфейса имеют приоритет над переменными стека (`SMTP_ENABLED`, `SMTP_HOST`, `SMTP_PORT`, `SMTP_USERNAME`, `SMTP_PASSWORD=<пароль приложения>`, `SMTP_SENDER`); неполная конфигурация отключает отправку. Пароль приложения не должен попадать в журналы, скриншоты и системы контроля версий. Здесь же задаётся **публичный адрес системы** — адрес интерфейса (например, `https://care.example.com`), по которому строится кнопка «Открыть задание» в письмах; если адрес не задан ни в интерфейсе, ни в переменной стека `PUBLIC_BASE_URL`, письма уходят без ссылки. Значение из интерфейса приоритетнее переменной независимо от флага отправки.
- **Политика паролей**: минимальная длина (по умолчанию 12) действует на все пути смены и выдачи паролей, включая административный сброс.

Секреты системных настроек хранятся в базе в зашифрованном виде; ключ шифрования — `SETTINGS_ENCRYPTION_KEY`. Потеря ключа не роняет систему, но делает сохранённые секреты нечитаемыми — их придётся ввести заново.

Проверка системы

- `GET http://<сервер>:3000/health` — процесс backend жив (запрос проксируется через интерфейс).
- `GET http://<сервер>:3000/ready` — готовность: доступность БД, наличие встроенной учётной записи и корректный ключ шифрования; именно этот запрос используется как healthcheck контейнера backend. Ошибки дают 503 с кодами: `identity_not_provisioned` (bootstrap не выполнен — проверьте журнал сервиса `migrate` и `BOOTSTRAP_SUPERADMIN_PASSWORD`), `settings_key_invalid` (неверный или отсутствующий `SETTINGS_ENCRYPTION_KEY`).
- Метрики — `/metrics` backend в формате Prometheus, доступны внутри сети стека (см. «Переменные стека»); ключевые: `toir_http_requests_total` (HTTP), `toir_auth_login_attempts_total{provider,result}` (попытки входа),

`toir_auth_session_events_total{event}` (выдача/отзыв сессий), счётчики планировщика и уведомлений.

- `sh scripts/smoke.sh` — анонимная дымовая проверка на dev-стенде ЛИИС (в поставку не входит): готовность БД, метрики, интерфейс, доступность руководств Word и ассерты 401 на защищённых `/api/*`. Скрипт строго read-only: не входит в систему, не создаёт записей, не шлёт почту.
- Журнал действий пользователей — раздел «Аудит» в интерфейсе; технические журналы контейнеров (Portainer → Containers → Logs либо `docker logs`) — только диагностика: тексты ошибок и значения полей в них не пишутся, для сопоставления используйте «Код обращения» (request ID), который видит пользователь.

После перезагрузки сервера контейнеры поднимаются политикой перезапуска Docker одновременно, без порядка «БД → backend → интерфейс»: первые одну-две минуты backend может показывать `unhealthy`, а интерфейс — отдавать ошибку, пока MySQL не завершит запуск. Это штатное самовосстановление, вмешательство не требуется; сервис `migrate` при перезагрузке повторно не выполняется (и не должен). Если через пять минут `/ready` не отвечает `200` — смотрите журналы контейнеров (раздел «Сообщения системному администратору»).

Дополнительные возможности

- **Демонстрационные данные** (dev-стенд ЛИИС): отдельный загрузчик `scripts/load_demo_data.py` наполняет стенд через публичный API под сервисной учётной записью (создайте её с ролью «Системный администратор», смените временный пароль, рабочий пароль передайте только через переменную окружения `DEMO_LOADER_PASSWORD`). Загрузчик идемпотентен: повторный запуск не дублирует данные. В поставку клиенту демонстрационные данные не входят.
- **Аварийное отключение аутентификации**: `AUTH_ENABLED=false` у сервиса backend включает синтетического пользователя с операционными правами — интерфейс работает без входа, но сам вход, смена пароля и административные разделы недоступны. В поставляемом `compose.yaml` этой переменной нет намеренно: включить режим можно только правкой блока `environment` сервиса `backend` в редакторе стека (`AUTH_ENABLED: "false"`), через «Environment variables» он не подействует. Использовать только для локальной диагностики; на рабочей системе не включать.
- **Swagger**: интерактивная документация API доступна только на dev-стенде ЛИИС по адресу `http://127.0.0.1:8000/docs`; в поставке порт backend наружу не публикуется.

Сообщения системному администратору

Сообщение / симптом	Причина	Действие
`required variable … is missing` при развёртывании	Не задана обязательная переменная стека.	Заполните переменную (см. «Переменные стека») и повторите развёртывание.

Сообщение / симптом	Причина	Действие
Deploy/Update the stack завершается ошибкой вида `service "migrate" didn't complete successfully: exit N` (или `dependency failed to start: container <стек>-migrate-1 exited (N)`); `backend` и `frontend` не запущены, интерфейс недоступен	Сервис `migrate` завершился с ошибкой, зависящие от него сервисы не стартовали. Код `2` — при первом запуске не задан `BOOTSTRAP_SUPERADMIN_PASSWORD`; код `1` — ошибка миграций или подключения к БД (неверный `MYSQL_PASSWORD` → `Access denied`, MySQL не успела запуститься, нет места на диске).	При неудачном **первом** Deploy Portainer не сохраняет стек и удаляет созданные контейнеры — ориентируйтесь на код `N` в тексте ошибки, исправьте переменные и разверните стек заново. Если стек уже существовал (ошибка при **Update the stack**), контейнеры остаются: Containers → `<стек>-migrate-1` → Logs (для кода 2 там строка `bootstrap: учётной записи superadmin нет…`); после исправления — **Update the stack**; повторный запуск безопасен — миграции идемпотентны.
Интерфейс открывается, но `backend` в состоянии `unhealthy` дольше пяти минут (обычно после перезагрузки сервера или восстановления копии)	Backend не проходит проверку готовности.	Откройте `http://<сервер>:3000/ready`: в ответе указана причина (см. строки ниже); при `502` backend не отвечает — смотрите его журнал.
`/ready` → 503 `identity_not_provisioned`	Bootstrap встроенной учётной записи не выполнен.	Проверьте журнал сервиса `migrate`; убедитесь, что при первом запуске был задан `BOOTSTRAP_SUPERADMIN_PASSWORD`; задайте переменную и выполните **Update the stack**.
`/ready` → 503 `settings_key_invalid`	`SETTINGS_ENCRYPTION_KEY` изменён или не совпадает с тем, которым зашифрованы настройки (например, после восстановления копии).	Верните ключ, которым шифровались секреты, либо задайте новый и введите OIDC/SMTP-секреты заново.
`migrate` или `backend` в журнале сообщают об ошибке подключения к БД без `Access denied` (например, `Name or service not known`, `Can't connect to MySQL server`)	В `MYSQL_PASSWORD` попали спецсимволы (`@`, `/`, `:`, `#`), и строка подключения разобрана неверно, либо MySQL ещё не запущена.	Задайте пароль только из латиницы и цифр (он входит в строку подключения без экранирования); при смене пароля после инициализации тома — см. «Переменные стека».
Вход выключен, интерфейс отвечает 503 на вход	Включён аварийный режим `AUTH_ENABLED=false` (правкой YAML стека).	Уберите `AUTH_ENABLED` из блока `environment` сервиса `backend` и выполните **Update the stack**.
Письма не отправляются	SMTP выключен или конфигурация неполна.	Проверьте флаг включения и все поля SMTP в разделе «Система» (они приоритетнее переменных стека).

Сообщение / симптом	Причина	Действие
Корпоративный вход возвращает на экран входа с ошибкой	Ошибка конфигурации OIDC (адреса, клиент, секрет, `OIDC_REDIRECT_URI`) либо ротация ключа шифрования посреди входа.	Сверьте настройки провайдера в разделе «Система» и переменные `OIDC_*`; повторите вход; в разделе «Аудит» смотрите события неудачного входа, а при подключённом сборщике метрик — `toir_auth_login_attempts_total{provider="oidc",result="failure"}`.
Пользователь заблокирован после неудачных попыток	Сработала защита от подбора (10 неудач ≈ 15 минут).	Подождать либо сбросить пароль в разделе «Пользователи» — сброс снимает блокировку.
Порт `UI_PORT` занят	На сервере уже слушает другой процесс.	Измените `UI_PORT` в переменных стека и выполните **Update the stack**.