

# Инструкция для подключения аудитора к демонстрационному окружению МБорд

## 1. Подключение к серверу

Доступ осуществляется по SSH к серверу, на котором развёрнуто демонстрационное окружение.

- IP: 84.38.182.23
- Порт: 22
- Логин: auditor
- Пароль: предоставляется отдельно

```
ssh auditor@84.38.182.23
```

## 2. Архитектура окружения

Окружение построено на базе Docker Compose и состоит из 4 контейнеров:

Контейнер	Назначение	Порт
metrikor-audit-db	TimescaleDB (PostgreSQL 15 + TimescaleDB 2.14) — хранение данных	5432 (внутренний)
metrikor-audit-redis	Redis 7 — кэширование и pub/sub	6379 (внутренний)
metrikor-audit-backend	FastAPI (Python 3.11) — REST API, ETL pipeline, автогенерация данных	8200 (localhost)
metrikor-audit-frontend	React 19 (nginx) — веб-интерфейс	8201 (localhost)

Входящий HTTPS-трафик проходит через nginx на хосте (порт 443), который проксирует запросы:

- /api/\*, /health, /ready → backend (127.0.0.1:8200)
- остальное → frontend (127.0.0.1:8201)

SSL-сертификат выпущен через Let's Encrypt (автообновление через certbot).

## 3. Точки доступа к приложению

URL	Назначение
https://demo.metrikor.ru	Веб-интерфейс системы мониторинга
https://demo.metrikor.ru/api/v1/docs	Документация REST API (Swagger UI)
https://demo.metrikor.ru/health	Health check endpoint

## 4. Учётные записи для проверки

В системе предустановлены 4 учётные записи с различными ролями:

Роль	Логин	Пароль	Описание
Администратор	admin@metrikor.ru	Demo2026!admin	Полный доступ: управление пользователями, контроллерами
Инженер	engineer@metrikor.ru	Demo2026!eng	Просмотр + управление контроллерами
Оператор	operator@metrikor.ru	Demo2026!oper	Просмотр данных, подтверждение алертов
Наблюдатель	viewer@metrikor.ru	Demo2026!view	Только просмотр данных (read-only)

## 5. Расположение исходного кода

Исходный код размещён в GitLab:

Репозиторий	URL	Содержимое
backend	https://gitlab.pdd.dev/metrikor-audit/backend	FastAPI backend, ETL pipeline, REST API
frontend	https://gitlab.pdd.dev/metrikor-audit/frontend	React SPA, веб-интерфейс
deploy	https://gitlab.pdd.dev/metrikor-audit/deploy	Docker Compose, конфигурации окружения
arch	https://gitlab.pdd.dev/metrikor-audit/arch	Архитектурная документация, регистрационные документы

Каждый репозиторий содержит один коммит с полным исходным кодом версии 1.0.0.

## 6. Проверка работоспособности

### 6.1 Проверка состояния контейнеров

```
docker ps --format "table {{.Names}}\t{{.Status}}" | grep audit
```

Ожидаемый результат — все 4 контейнера в статусе `Up ... (healthy)`:

```
metrikor-audit-frontend    Up X hours (healthy)
metrikor-audit-backend    Up X hours (healthy)
metrikor-audit-redis      Up X hours (healthy)
metrikor-audit-db        Up X hours (healthy)
```

## 6.2 Проверка health endpoint

```
curl -s https://demo.metrikor.ru/health
```

Ожидаемый результат:

```
{"status":"healthy","timestamp":"...","service":"metrikor-backend","version":"1.0.0"}
```

## 6.3 Проверка API документации

Открыть в браузере: <https://demo.metrikor.ru/api/v1/docs>

Ожидаемый результат — открывается Swagger UI с описанием всех REST API endpoints.

## 6.4 Проверка логов

```
docker logs metrikor-audit-backend --tail 20
docker logs metrikor-audit-db --tail 10
```

Ожидаемый результат — нет ошибок уровня ERROR, сервисы работают штатно.

# 7. Проверка приложения (функциональные тесты)

## Тест 1 — Доступность веб-интерфейса

1. Открыть в браузере <https://demo.metrikor.ru>
2. Ожидаемый результат: отображается страница авторизации

## Тест 2 — Авторизация

1. Ввести логин `admin@metrikor.ru` и пароль `Demo2026!admin`
2. Нажать «Войти»
3. Ожидаемый результат: открывается дашборд с карточками датчиков

## Тест 3 — Дашборд

1. После входа проверить наличие карточек датчиков с текущими значениями
2. Проверить статистику: онлайн/оффлайн датчики, активные алерты
3. Ожидаемый результат: отображаются данные датчиков, сгруппированные по контроллерам

## Тест 4 — Детальная страница датчика

1. Нажать на любую карточку датчика на дашборде
2. Ожидаемый результат: открывается страница с текущим значением, единицей измерения, графиком истории, пороговыми значениями

## Тест 5 — Алерты

1. Перейти в раздел «Алерты» в боковом меню
2. Ожидаемый результат: отображается список алертов с фильтрацией по статусу

## Тест 6 — Контроллеры

1. Перейти в раздел «Контроллеры» в боковом меню
2. Ожидаемый результат: отображается список контроллеров с их статусами

## Тест 7 — Пользователи

1. Перейти в раздел «Пользователи» в боковом меню
2. Ожидаемый результат: отображается список пользователей с ролями

## Тест 8 — Разграничение доступа (RBAC)

1. Выйти из системы
2. Войти под учётной записью `viewer@metrikor.ru` / `Demo2026!view`
3. Ожидаемый результат: в боковом меню отсутствуют пункты «Контроллеры» и «Пользователи» (доступны только администратору)

## Тест 9 — REST API

```
curl -s https://demo.metrikor.ru/health
```

Ожидаемый результат: HTTP 200, JSON с version 1.0.0.

## 8. Структура Docker Compose

Конфигурация окружения: `/opt/metrikor-audit/docker-compose.audit.yml`

```
cat /opt/metrikor-audit/docker-compose.audit.yml
```

## Основные директории на сервере

Путь	Назначение
/opt/metrikor-audit/	Корневая директория окружения
/opt/metrikor-audit/backend/	Исходный код backend
/opt/metrikor-audit/frontend/	Исходный код frontend
/opt/metrikor-audit/docker-compose.audit.yml	Docker Compose конфигурация
/opt/metrikor-audit/.env.audit	Переменные окружения

## Docker volumes

Volume	Назначение
metrikor_audit_pgdata	Данные TimescaleDB
metrikor_audit_csv_watch	CSV файлы для ETL pipeline
metrikor_audit_backend_logs	Логи backend сервиса

## 9. Контактная информация

По вопросам подключения и проверки:

- **Организация:** ООО «Метрикор»
- **Email:** info@metrikor.ru
- **Ответственный:** Касинский Роман Алексеевич