

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

HR:ERP

Инструкция по установке и эксплуатации программного обеспечения

Автоматизированная система управления персоналом предприятия

1. Общие сведения

Программное обеспечение "HR:ERP" поставляется по модели SaaS: развертывание и сопровождение выполняются правообладателем на собственной серверной инфраструктуре (выделенные серверы на базе дата-центров Selectel на территории РФ). Пользователю для работы достаточно современного веб-браузера; установка ПО на стороне пользователя не требуется.

Настоящая инструкция описывает развертывание серверной части ПО "HR:ERP" уполномоченным техническим специалистом. Приложение поставляется в виде контейнерного образа Docker и разворачивается совместно с внешними сервисами - PostgreSQL, Redis, S3-совместимое хранилище MinIO. Для промышленной эксплуатации применяется оркестрация в среде Kubernetes.

2. Требования к окружению

2.1. Аппаратные требования

- процессор: от 4 vCPU (рекомендуется 8 vCPU);
- оперативная память: от 8 ГБ (рекомендуется 16 ГБ);
- дисковое пространство: от 50 ГБ SSD;
- стабильное сетевое подключение.

2.2. Программные требования

- операционная система: Linux (семейство Debian/Ubuntu);
- среда контейнеризации: Docker и Docker Compose (для одиночного развертывания) либо Kubernetes (для промышленной эксплуатации);
- СУБД PostgreSQL 15+ с расширениями pg_trgm, hstore, pgcrypto, plpgsql;
- хранилище Redis 6+;
- S3-совместимое объектное хранилище (для файлов и вложений);
- веб-сервер Nginx (терминация TLS, обратное проксирование).

3. Подготовка внешних зависимостей

3.1. PostgreSQL

Назначение: основная база данных приложения. Создайте базу данных и пользователя-владельца, после чего включите необходимые расширения:

```
sudo -u postgres createuser --pwprompt hrerp
sudo -u postgres createdb --owner hrerp hrerp_production
sudo -u postgres psql -d hrerp_production -c "CREATE EXTENSION IF NOT EXISTS plpgsql;"
sudo -u postgres psql -d hrerp_production -c "CREATE EXTENSION IF NOT EXISTS pgcrypto;"
sudo -u postgres psql -d hrerp_production -c "CREATE EXTENSION IF NOT EXISTS pg_trgm;"
sudo -u postgres psql -d hrerp_production -c "CREATE EXTENSION IF NOT EXISTS hstore;"
```

Строка подключения в формате

postgres://<ПОЛЬЗОВАТЕЛЬ>:<ПАРОЛЬ>@<ХОСТ>:5432/<БД> задается в переменной окружения DATABASE_URL.

3.2. Redis

Назначение: кэш, пользовательские сессии, очереди фоновых заданий (Sidekiq) и канал реального времени (Action Cable). Установите Redis и задайте пароль; строка подключения формата redis://<пароль>@<хост>:6379/1 указывается в переменной REDIS_URL.

3.3. S3-совместимое хранилище

Назначение: хранение файлов и вложений (Active Storage / Shrine). Используется S3-совместимое объектное хранилище (например, объектное хранилище Selectel или MinIO). В конфигурации приложения задаются эндпоинт, ключи доступа, имя бакета и регион.

4. Переменные окружения

Конфигурация приложения задается через переменные окружения. Основные параметры:

Переменная	Назначение
RAILS_ENV	Режим работы приложения (production)
SECRET_KEY_BASE	Секретный ключ приложения для подписи сессий
DATABASE_URL	Строка подключения к PostgreSQL
REDIS_URL	Строка подключения к Redis
S3_ENDPOINT, S3_BUCKET, S3_REGION	Параметры S3-совместимого хранилища

Переменная	Назначение
S3_ACCESS_KEY_ID, S3_SECRET_ACCESS_KEY	Учетные данные доступа к хранилищу
OIDC_* / KEYCLOAK_*	Параметры федеративной аутентификации (OpenID Connect / Keycloak)
GOOGLE_CLIENT_ID, GOOGLE_CLIENT_SECRET	Учетные данные интеграции с Google Workspace
SENTRY_DSN	Адрес сбора ошибок (мониторинг Sentry)
Параметры интеграций	Учетные данные JIRA, Pyrus, Float.rp, HeadHunter (hh.ru), Slack и др. (по необходимости)

5. Установка через Docker

Для одиночного развертывания используется файл docker-compose, описывающий приложение и внешние сервисы. Пример структуры (фрагмент):

```
services:
  postgres:
    image: postgres:15-alpine
    environment:
      POSTGRES_USER: hrerp
      POSTGRES_PASSWORD: <пароль>
      POSTGRES_DB: hrerp_production
    ports: ["5432:5432"]
  redis:
    image: redis:7-alpine
    ports: ["6379:6379"]
  web:
    image: <реестр>/hrerp:<версия>
    env_file: .env
    depends_on: [postgres, redis]
    command: bundle exec puma -C config/puma.rb
    ports: ["3000:3000"]
  sidekiq:
    image: <реестр>/hrerp:<версия>
    env_file: .env
    depends_on: [postgres, redis]
    command: bundle exec sidekiq
```

Запуск:

```
docker compose up -d
```

6. Инициализация базы данных

После запуска контейнера приложения выполните миграции схемы базы данных:

```
docker compose exec web bin/rails db:prepare  
docker compose exec web bin/rails db:migrate
```

Новые версии приложения могут изменять структуру и состав таблиц в базе. Приложение не создает и не удаляет БД, поэтому права администратора СУБД не требуются - достаточно прав владельца базы данных.

7. Веб-сервер и HTTPS

Перед приложением размещается Nginx, выполняющий терминацию TLS и обратное проксирование на сервер приложений Puma (порт 3000). Все клиент-серверное взаимодействие осуществляется по протоколу HTTPS.

8. Промышленное развертывание (Kubernetes)

Для промышленной эксплуатации приложение развертывается в кластере Kubernetes. Сборка контейнерных образов и развертывание автоматизированы средствами GitLab CI/CD (сборка образов выполняется средством Kaniko). Предусмотрены среды разработки, тестирования, предпросмотра, демонстрации и промышленной эксплуатации, а также возможность отката (rollback) на предыдущую версию.

9. Проверка работоспособности

1. Убедитесь, что контейнеры приложения, фонового обработчика (Sidekiq), PostgreSQL и Redis запущены.
2. Откройте адрес приложения в браузере - должна отобразиться страница авторизации.
3. Выполните вход с учетными данными администратора и проверьте доступность основных модулей.
4. Проверьте поступление событий в систему мониторинга (Sentry) и метрик приложения.

10. Поддержка

По вопросам установки и эксплуатации обращайтесь в службу технической поддержки:
телефон 8-958-111-9263, электронная почта support@cifraprom.ru.