

**Руководство администратора
«HRS Целеполагание и оценка»**

ОГЛАВЛЕНИЕ

[1.Термины и сокращения3](#)

[2.Введение4](#)

[3.Общие сведения о системе5](#)

[4.Описание архитектуры ИС6](#)

[4.1Графическая схема развертывания ИС6](#)

[4.1.1 Прикладные сервисы HRS6](#)

[4.1.2 Инфраструктурные и другие составляющие HRS7](#)

[4.2Логическая схема8](#)

[4.3Реестр компонентов ИС9](#)

[4.4Реестр оборудования11](#)

[4.5Связь ИС с внешним окружением11](#)

[5.Подготовка к развертыванию ИС12](#)

[5.1Общие правила подачи заявки на создание сетевых правил12](#)

[5.2Заявка на получение доступа к виртуальным машинам с правами администратора12](#)

[5.3Заявка на получение доступа к ИС12](#)

[5.4Заявка для связи ИС со сторонними сервисами12](#)

[6.Развертывание и установка ИС12](#)

[6.1Перечень и состав работ перед установкой ИС13](#)

[6.2Настройка realm Keycloak14](#)

[6.3Перечень и состав работ для установки ИС15](#)

[7.Дополнительные настройки ИС16](#)

[7.1Настройка подключения к AD16](#)

[8.Администрирование ИС16](#)

[8.1Работа с учетными записями16](#)

[8.2Резервное копирование и восстановление системы16](#)

[8.3Работа с сертификатами16](#)

[8.4Операции с персональными данными17](#)

[9.Приложение 1. Схема сетевого взаимодействия18](#)

[10.Приложение 2. Настройки конфигурационных файлов и лог-файлов22](#)

[10.1Общие параметры22](#)

[10.1.1 Журналирование и телеметрия .NET-сервисов22](#)

[10.1.2 Лог-файлы контейнеров23](#)

[10.1.3 Аутентификация JWT/OIDC24](#)

[10.1.4 Авторизация через Users24](#)

[10.1.5 RabbitMQ25](#)

[10.1.6 TLS и ограничения Kestrel25](#)

[10.2Frontend-сервисы25](#)

[10.2.1 web-ui-trainingmanagement26](#)

[10.2.2 web-ui-main27](#)

[10.2.3 web-ui-admin28](#)

[10.2.4 web-ui-goals29](#)

[10.2.5 ui-docs30](#)

[10.3Backend-сервисы30](#)

[10.3.1 gateway31](#)

[10.3.2 training-management32](#)

[10.3.3 goals32](#)

[10.3.4 users34](#)

[10.3.5 reports35](#)

[10.3.6 notifications35](#)

[10.3.7 scheduler37](#)

[10.3.8 sap-pi-ait-integration-adapter38](#)

[10.4Инфраструктурные параметры Docker Compose40](#)

[10.4.1 Локальная разработка40](#)

[10.4.2 Общие переменные развертывания40](#)

[10.4.3 Версии образов40](#)

[10.4.4 Строки подключения и секреты41](#)

[10.5Источники42](#)

1. ТЕРМИНЫ И СОКРАЩЕНИЯ

№	Термин	Определение
1.	AD	Active Directory
2.	БД	База данных
3.	ИБ	Информационная безопасность
4.	ИС	Информационная система
5.	ИТ	Информационные технологии
6.	ПО	Программное обеспечение
7.	РП	Руководитель проекта
8.	СЭД	Система электронного документооборота
9.	Роль	Совокупность полномочий, которые предоставляются пользователю в ИС для выполнения должностных обязанностей.
10.	УЗ	Учетная запись

2. ВВЕДЕНИЕ

Настоящий документ предназначен для работников эксплуатирующей организации, в должностные обязанности которых входят функции по администрированию ИС HRS.

3. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ

Полное наименование – Кадровые сервисы.

Условное обозначение – HRS.

«HRS Целеполагание и оценка» разработана для автоматизации процессов постановки целей, отслеживания ключевых результатов и оценки эффективности работников.

В программе «HRS Целеполагание и оценка» реализованы возможности для:

- оценки ключевых показателей эффективности (далее - КПЭ) работников компании,
- включения в процесс постановки и оценки целей дополнительных участников, выстраивания кросс-взаимодействия,
- контроля за соблюдением сроков и полноты заполнения карт целей с помощью построения отчетов,
- создания справочника корпоративных целей для каскадирования и отслеживания.

Автоматизируемые в модуле бизнес-процессы описаны в соответствующих разделах Функционально-технических требований.

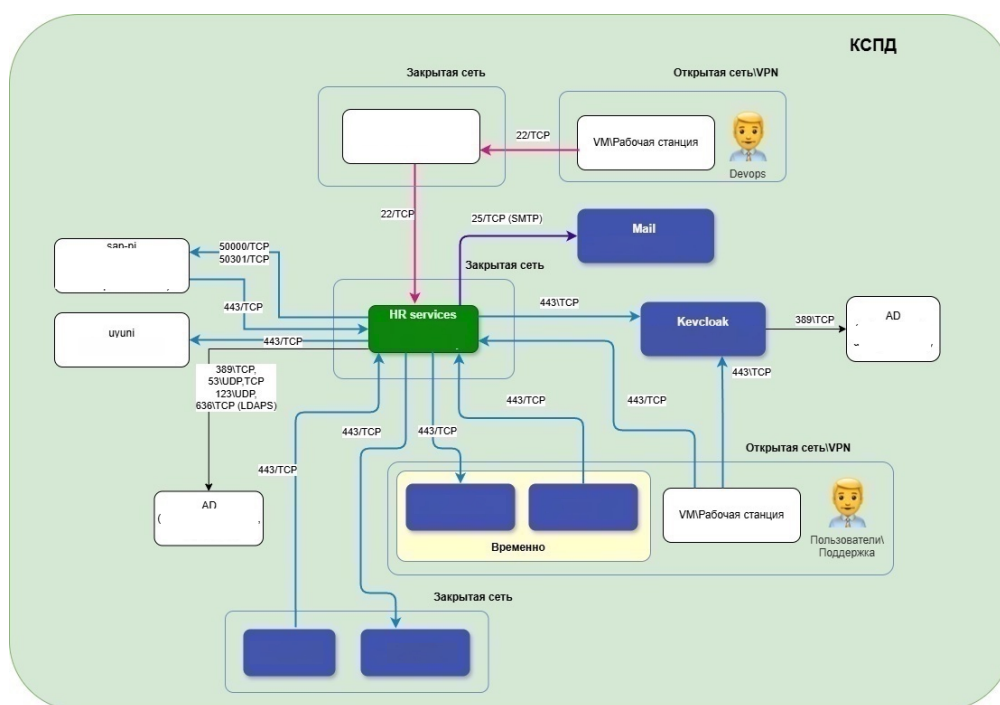
4. ОПИСАНИЕ АРХИТЕКТУРЫ ИС

ИС HRS спроектирована по принципам микросервисной архитектуры. Каждый компонент ИС в общем случае представляет собой изолированное приложение, запущенное в контейнере, под управлением Kubernetes.

Компоненты можно разделить на инфраструктурные и функциональные. В п. 4.1 содержится информация по серверам для каждого контура ИС – prod, stage.

4.1 ГРАФИЧЕСКАЯ СХЕМА РАЗВЕРТЫВАНИЯ ИС

Графическая схема развертывания ИС (сетевая схема целевой продуктивной среды) представлена на следующем рисунке.



а.

Рисунок 1. Графическая схема развертывания ИС

Информация о подготовке к развертыванию ИС, а также конфигурациях прикладных и инфраструктурных сервисов размещена на портале по адресу: docs.hrs.alrosa.ru в разделе Разработка/Docker и Git.

4.1.1 ПРИКЛАДНЫЕ СЕРВИСЫ HRS

Основные сервисы HRS представлены в следующей таблице:

Название репозитория	Ссылка	Описание
alrosa.hrs.gateway	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.gateway	Доменный сервис для идентификации пользователей
alrosa.hrs.goals	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.goals	Доменный сервис back-end

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

		«Целься»
alrosa.hrs.notifications	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.notifications	Доменный сервис back-end уведомлений.
alrosa.hrs.trainningmanagement	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.trainingmanagment	Доменный сервис back-end «Обучение».
alrosa.hrs.users	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.users	Доменный сервис back-end пользователей и организационной структуры.
alrosa.hrs.ui-admin	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-admin	Сервис front-end для системного администратора.
alrosa.hrs.ui-goals	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-goals	Сервис front-end «Целься».
alrosa.hrs.ui-main	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-main	Сервис front-end главной страницы с выбором модулей.
alrosa.hrs.ui-trainingmanagement	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-trainingmanagement	Сервис front-end «Обучение».
alrosa.hrs.docs	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.docs	Портал с документацией к ИС.
alrosa.hrs.reports	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.reports	Сервис back-end для отчетов.

4.1.2 ИНФРАСТРУКТУРНЫЕ И ДРУГИЕ СОСТАВЛЯЮЩИЕ HRS

Основные составляющие инфраструктуры представлены в следующей таблице:

Название репозитория	Ссылка	Описание
alrosa.hrs.deploy	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.deploy	Репозиторий с конфигурацией развёртывания Dev и Prod окружений HRS в docker-compose.
alrosa.hrs.argocd	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.argocd	Репозиторий, содержащий в себе манифесты с Apps для ArgoCD, а также различными

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

		конфигурациями самого ArgoCD (используется при создании тестового стенда).
alrosa.hrs-helm-charts	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.helm-charts	Репозиторий с helm-чартами сервисов HRS, а также некоторыми инфраструктурными сервисами.
alrosa.hrs.autotests	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.autotests	Репозиторий с автотестами, написанными на Python. Используется для автоматического тестирования новых версий HRS-сервисов.

Используется следующий набор инфраструктурных библиотек:

Название репозитория	Ссылка	Описание
alrosa.hrs.infrastructure	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.infrastructure	Набор инфраструктурных библиотек, используемых для back-end.
alrosa.hrs.ui-core	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-core	Набор инфраструктурных библиотек, используемых для front-end.
alrosa.hrs.ui-feature	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-feature	Набор инфраструктурных библиотек, используемых для back-end.
alrosa.hrs.ui-kit	https://gitlab-ext.alrosa.ru/alrosa-crt/hrs/alrosa.hrs.ui-kit	Общий UI-пакет для работы приложений.

4.2 ЛОГИЧЕСКАЯ СХЕМА

На схеме архитектуры представлен целевой вариант проектирования ИС.

Исходник схемы в более высоком качестве размещен по адресу:

<https://confluence.internal/pages/viewpage.action?pageId=384239937>

оптимально < 3 сек
800 пользователей в пике

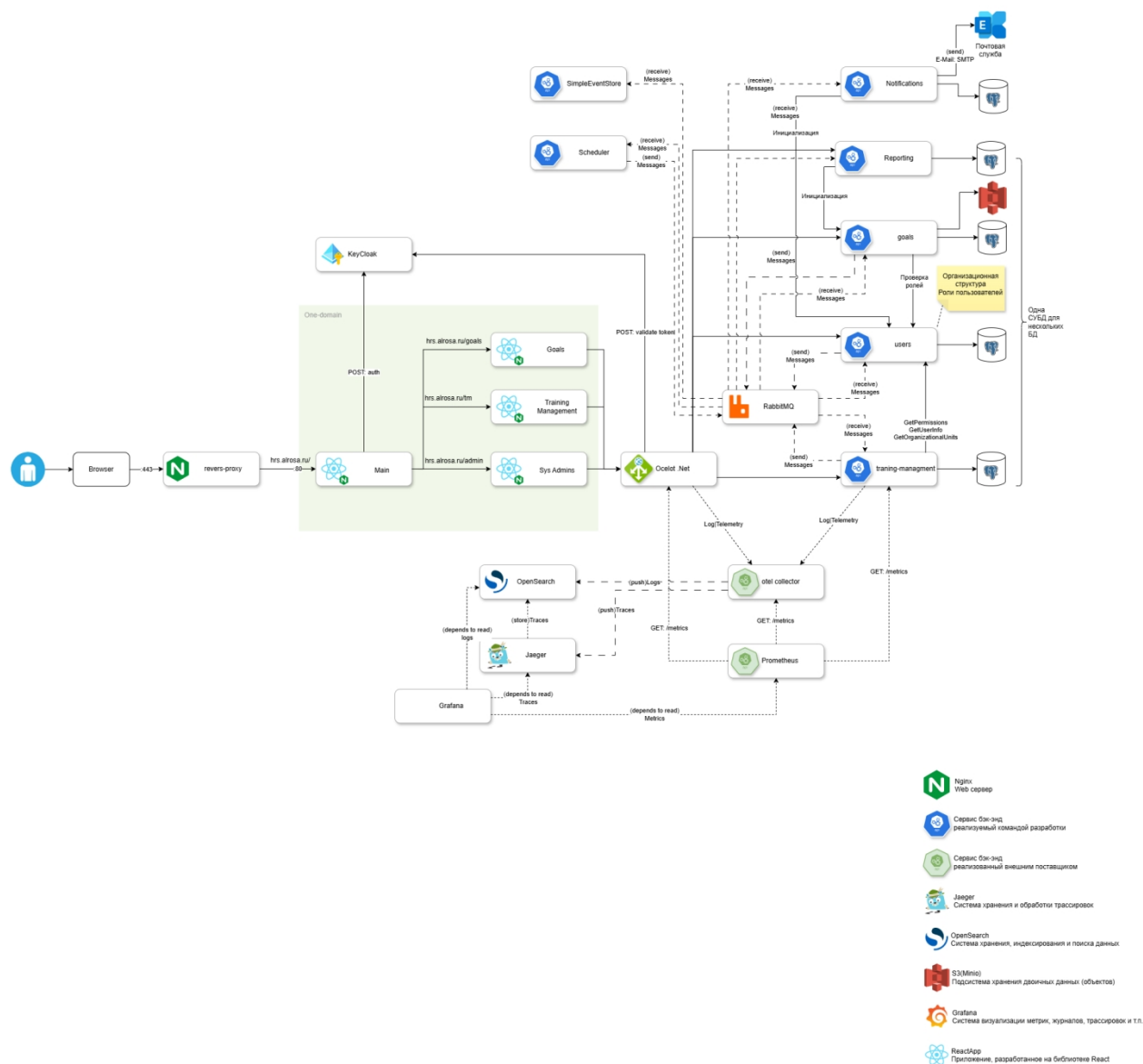


Рисунок 2. Логическая схема архитектуры ИС

4.3 РЕЕСТР КОМПОНЕНТОВ ИС

Для полноценной работы ИС необходима инфраструктура, представленная в таблице:

Наименование компонента	Используемое ПО и техническая характеристика	Версия
Подсистема хранения реляционных данных	PostgreSQL	v.17.2
Репозиторий deb-пакетов	sonatype/nexus3 Sonatype Nexus — программный продукт, который представляет собой менеджер репозитория для хранения артефактов. Поддерживаются такие форматы артефактов, как Maven, образы Docker, Python PyPI,	v.3.64.0

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

	RubyGems, npm, nuget, deb и другие	
Подсистема единого входа	Keycloak Keycloak — это система идентификации и управления доступом, которая позволяет управлять идентификацией пользователей, контролировать доступ к приложениям и данным, обеспечивая единую точку входа (SSO).	v.23
Подсистема хранения объектов	MinIO Console Сервер хранения объектов с открытым исходным кодом, совместимый с облачным хранилищем Amazon S3.	RELEASE.2025-02-28T09-55-16Z-cpuv1
Подсистема передачи сообщений	RabbitMQ Программный брокер сообщений на основе стандарта AMQP	3-management-alpine
Подсистема непрерывной доставки	GitLab GitLab — платформа совместной разработки ПО, которая сочетает в себе систему управления версиями (Git), баг-трекер, CI/CD	Не ниже v.17.0.1
	GitLab Runner GitLab Runner — это веб-приложение (агент), предназначенное для запуска и автоматического выполнения процессов CI/CD в GitLab.	Не ниже alpine-v.16.7.0
Подсистема сбора логов	OpenSearch OpenSearch — ПО, состоящее из поисковой системы и панели визуализации данных для этой поисковой системы (система хранения, индексирования и поиска данных).	v.2.18.0
Система мониторинга	Prometheus	v.3.0.1
Система хранения и обработки трассировок	Jaeger AIO	v.1.52
Система визуализации метрик, журналов, трассировок и т.п.	Grafana	v.11.0.4
OpenTelemetry	open-telemetry / opentelemetry-collector-contrib Инструмент для мониторинга и трассировки, который интегрируется с системой Gateway для сбора метрик и логов.	v.0.122.0
Web-портал	Главная страница с перечнем модулей	-
Информационный портал	Docs Информационный портал для ведения документации о системе	-

Также в ИС используется **Ocelot** — библиотека для API Gateway, которая поддерживает маршрутизацию, балансировку нагрузки, кэширование, аутентификацию и другие функции. Всю инфраструктуру можно развернуть вместе с прикладными сервисами, но данный

вариант не рекомендуется для производственного развертывания.

4.4 РЕЕСТР ОБОРУДОВАНИЯ

В таблице представлена информация о:

- перечне оборудования, которое используется в ИС,
- требуемых характеристиках,
- данных по сетевым взаимодействиям кластера DM с окружением в КСПД.

Решение работает в виртуальной среде.

Используются два сервера – Stage и Prod:

Имя сервера	IP-адрес	ОС	CPU	RAM	SSD	Среда
		Astra Linux 1.7.x	6	16	150GB	stage
		Astra Linux 1.7.x	8	32	150GB	prod

4.5 СВЯЗЬ ИС С ВНЕШНИМ ОКРУЖЕНИЕМ

В системе используется настроенная интеграция:

- системой «Босс-Кадровик» для получения актуального статуса сотрудников,
- почтовой программой для получения уведомлений о событиях в ИС.

5. ПОДГОТОВКА К РАЗВЕРТЫВАНИЮ ИС

Оформите все необходимые заявки в HD (<https://helpdesk.alrosa.ru>)

- на получение доступа к виртуальным машинам;
- на получение прав администратора;
- на создание сетевых правил.

5.1 ОБЩИЕ ПРАВИЛА ПОДАЧИ ЗАЯВКИ НА СОЗДАНИЕ СЕТЕВЫХ ПРАВИЛ

Для создания сетевого правила, необходимо написать заявку в HD

<https://helpdesk.alrosa.ru/portal/sc?item=168320507006929937>

- В тексте заявки укажите: «В рамках проекта HRS прошу согласовать открытие портов, согласно приложенному шаблону схемы межсетевого взаимодействия для ...».
- Если при открытии порта есть риск утечки чувствительных данных, то более подробно опишите причину открытия этого порта и как он будет использоваться, желательно - с указанием названия сервисов или стороннего ПО.
- К заявке **ОБЯЗАТЕЛЬНО** приложите шаблон схемы сетевого взаимодействия.
- В получателя услуги добавьте РП (для отслеживания статуса) и владельца продукта для продуктивной и тестовой среды.

5.2 ЗАЯВКА НА ПОЛУЧЕНИЕ ДОСТУПА К ВИРТУАЛЬНЫМ МАШИНАМ С ПРАВАМИ АДМИНИСТРАТОРА

Для получения доступа администраторов к серверам:

- создайте заявку, согласно правилам;
- Текст заявки:

Для реализации проекта (*наименование и код проекта*) выделить мощности

Production среды в разбивке по виртуальным машинам:

(*укажите список виртуальных машин администраторов*).

Пример с перечнем необходимых доступов для получения прав администратора в ИС представлен в таблице в п.

+ дополнительное дисковое хранилище в объеме – 6 ТБ HDD и 1 ТБ SSD, которое будут подключаться к сервисам Kubernetes

В соответствии с решением <И.О.Фамилия>, мощности должны быть развернуты в КСПД.

5.3 ЗАЯВКА НА ПОЛУЧЕНИЕ ДОСТУПА К ИС

Для получения доступа к ИС необходимо открыть порт 443.

Создайте заявку для открытия порта и приложите таблицу со списком пользователей, для которых необходим доступ к стенду.

Пример таблицы со схемой сетевого взаимодействия, которую необходимо приложить к заявке, приведен в Приложении 1.

5.4 ЗАЯВКА ДЛЯ СВЯЗИ ИС СО СТОРОННИМИ СЕРВИСАМИ

Для развертывания кластера и его корректной работы (связи ИС со сторонними сервисами) создайте заявку и приложите таблицу сетевого взаимодействия, которая приведена в Приложении 1.

6. РАЗВЕРТЫВАНИЕ И УСТАНОВКА ИС

Раздел содержит описание перечня подготовительных работ и шагов по развертыванию ИС.

6.1 ПЕРЕЧЕНЬ И СОСТАВ РАБОТ ПЕРЕД УСТАНОВКОЙ ИС

В разделе описывается перечень работ и условий, которые необходимо выполнить до начала развертывания и установки ИС.

Данная инструкция подразумевает заранее установленные на сервере Docker и Docker-Compose.

Для полноценной работы системы необходима следующая инфраструктура:

🔑 Keycloak
🔑 Postgres
🔑 RabbitMq
Prometheus
OpenSearch
Grafana
Jaeger

* Знаком 🔑 помечены критически важные сервисы, без которых ИС не будет работать. Всю инфраструктуру можно развернуть вместе с прикладными сервисами, но данный вариант не рекомендуется для производственного развёртывания.

Настройка Keycloak

1. Донастроить keycloak - создать БД:

- `sudo docker exec -it services-postgres-1 psql -U postgres -c "CREATE USER keycloak WITH PASSWORD 'password';"`
- `sudo docker exec -it services-postgres-1 psql -U postgres -c "CREATE DATABASE keycloak;"`
- `sudo docker exec -it services-postgres-1 psql -U postgres -c "ALTER DATABASE keycloak OWNER TO keycloak;"`
- sql лежит в Configs\postgres-scripts

2. Залогиниться в keycloak под admin- <https://hrs.internal:8443/>

Настройка Postgres

1. Запустить postgres с помощью команды:

```
sudo docker compose --env-file /opt/hrs/alrosa.hrs.deploy/Docker/Dev/.all_envs -f  
/opt/hrs/alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml -f  
/opt/hrs/alrosa.hrs.deploy/Docker/Dev/Infrastructure/docker-compose.yml up postgres
```

2. Создать БД для keycloak с помощью команды из файла, размещенного по адресу:

Docker/Prod/Configs/postgres-scripts/keycloak-init.sql (для продуктивной среды)

Настройка RabbitMq

1. Необходимо задать администратора при первом запуске. За это отвечают переменные среды:

RABBIT_MQ_ADMIN=guest

```
RABBIT_MQ_ADMIN_PASSWORD=guest
```

2. Настроить пользователя и vHost для сервисов. Настройка будет описана с указанием переменных среды.

```
RABBIT_MQ_HRS_VHOST=develop
```

```
RABBIT_MQ_HRS_USER=develop
```

```
RABBIT_MQ_HRS_USER_PASSWORD=develop
```

2.1 Создание vhost

```
curl -i -u ${RABBIT_MQ_ADMIN}:${RABBIT_MQ_ADMIN_PASSWORD} -H "content-type:application/json" \
-XPUT http://localhost:15672/api/vhosts/${RABBIT_MQ_HRS_VHOST}
```

2.2. Создание пользователя

```
curl -i -u ${RABBIT_MQ_ADMIN}:${RABBIT_MQ_ADMIN_PASSWORD} -H "content-type:application/json" \
-XPUT http://localhost:15672/api/users/${RABBIT_MQ_HRS_USER} \
-d '{"password":"${RABBIT_MQ_HRS_USER_PASSWORD}","tags":"management"}'
```

2.3 Предоставление прав пользователю

```
curl -i -u ${RABBIT_MQ_ADMIN}:${RABBIT_MQ_ADMIN_PASSWORD} -H "content-type:application/json" \
-XPUT
http://localhost:15672/api/permissions/${RABBIT_MQ_HRS_VHOST}/${RABBIT_MQ_HRS_USER} \
-d '{"configure":".*","write":".*","read":".*"}'

curl -i -u ${RABBIT_MQ_ADMIN}:${RABBIT_MQ_ADMIN_PASSWORD} -H "content-type:application/json" \
-XPUT http://localhost:15672/api/topic-
permissions/${RABBIT_MQ_HRS_VHOST}/${RABBIT_MQ_HRS_USER} \
-d '{"exchange":"amq.topic","write":"^a","read":".*"}'
```

После запуска сервисов проверить, что соединения установлены, выполнив следующий запрос:

```
curl -i -u ${RABBIT_MQ_ADMIN}:${RABBIT_MQ_ADMIN_PASSWORD} -H "content-type:application/json" \
-XGET http://localhost:15672/api/vhosts/${RABBIT_MQ_HRS_VHOST}/connections
```

* Документация по командам curl в также содержится в docs.

**** Корректная настройка rabbit необходима для отображения у пользователя периодов КПЭ.**

6.2 НАСТРОЙКА REALM KEYCLOAK

1. Создать Realm с наименованием 'HRServices'.
2. Создать клиента с наименованием 'hrs-web-ui'
3. Включить для клиента стандартный поток аутентификации с поддержкой 'Authorization Code Flow'
4. В Valid redirect URIs указать `https://<domain_name>/*`

5. В Valid post logout redirect URIs указать '+'
 6. В Web origins указать '+'
 7. Завести всех необходимых пользователей в Realm
- * Дополнительно настроить интеграции с AD или другим провайдером.

6.3 ПЕРЕЧЕНЬ И СОСТАВ РАБОТ ДЛЯ УСТАНОВКИ ИС

Шаги развертывания ИС:

1. Скопировать из гитлаб исходники проекта alrosa.hrs.deploy
 2. Сгенерировать сертификаты и положить их в /opt/hrs/ssl
 3. Выполнить настройку функциональных сервисов (подробнее – в Приложении 2).
 4. Настроить корректно .all_envs
 - Корректно настроить PATH_TO_REPO
 - И правильный (абсолютный) путь до сертификатов.
 5. Авторизоваться на docker.devops.alrosa.ru
 6. При необходимости закомментировать дополнительные сервисы - (opensearch, pgadmin, Grafana и др. при необходимости) в файлах docker-compose.yml
 7. Если ранее не настроен realm, необходимо выполнить п.6.2.
- Настройка realm keycloak должна быть выполнена до создания пользователей.
8. Запустить все сервисы с помощью команды:

```
sudo docker compose --env-file /opt/hrs/alrosa.hrs.deploy/Docker/Dev/.all_envs -f
/opt/hrs/alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml -f
/opt/hrs/alrosa.hrs.deploy/Docker/Dev/Infrastructure/docker-compose.yml up
```

*Несмотря на то, что сервисы разнесены по 2 compose, они связаны, поэтому поднятие обоих (2 compose)

9. Проверить журнал (записи) на предмет отсутствия ошибок и непредвиденных остановок сервисов.
10. Если какие-то сервисы не запустились, поднять отдельно с контролем логов:
 - `sudo docker compose --env-file /opt/hrs/alrosa.hrs.deploy/Docker/Dev/.all_envs -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Infrastructure/docker-compose.yml up minio`
 - `sudo docker compose --env-file /opt/hrs/alrosa.hrs.deploy/Docker/Dev/.all_envs -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Infrastructure/docker-compose.yml up rabbit-mq`
 - `sudo docker compose --env-file /opt/hrs/alrosa.hrs.deploy/Docker/Dev/.all_envs -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml -f /opt/hrs/alrosa.hrs.deploy/Docker/Dev/Infrastructure/docker-compose.yml up --force-recreate rabbit-mq`
11. Убедиться, что все очереди rabbit созданы
 - <http://hrs.internal:15672/#/queues>

- очереди сервисов - notifications, users, goals, integrations

12. Создать пользователя администратора – войти в ИС через интерфейс с УЗ администратора.

Прописать администратора - системного админа.

- <http://hrs.internal:10010/swagger/index.html>
- /internal/api/v1/InternalUsers/{login}/sys-admin-assign

13. Завести организацию и орг.структуру (через UI, блок администрирования).

14. Создать год и периоды:

- <http://hrs.internal:10011/swagger/index.html>
- /internal/api/v1/InternalPeriod/createForYearByQuarter/{tenantId}/{year}
- /internal/api/v1/InternalPeriod/createForYear/{tenantId}/{year}
- (tenantid взять из ui)

15. Убедиться в системе, что созданы карты КПЭ на периоды.

7. ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ ИС

Раздел содержит описание дополнительных настроек ИС.

7.1 НАСТРОЙКА ПОДКЛЮЧЕНИЯ К AD

Интеграция с AD используется для авторизации пользователей в ИС, выполнена через настройку в KeyCloak (<https://keycloak.alrosa.ru>, Realm hrsservices) подключения к контроллеру домена .

Сервера ИС введены в домен, администрирование осуществляется с использованием служебных УЗ.

8. АДМИНИСТРИРОВАНИЕ ИС

8.1 РАБОТА С УЧЕТНЫМИ ЗАПИСЯМИ

Подробное пошаговое описание порядка действий по созданию, настройке и изменению полномочий, а также удалению пользователей ИС содержится в Регламенте предоставления доступа к ИС.

Управление (заведение, активация, блокирование и уничтожение) учетными записями пользователей, в том числе внешних пользователей осуществляется по заявке в соответствии с порядком, описанным в Регламенте предоставления доступа к ИС.

8.2 РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ СИСТЕМЫ

Порядок действий администраторов по резервному копированию и восстановлению системы из резервной копии описан в Регламенте процесса «Управление доступностью и непрерывностью» и Плана аварийного восстановления и сценариях обеспечения доступности ИС «HRS Целеполагание и оценка».

Периодичность бекапирования - 1 раз в сутки в 22:00 по МСК, срок хранения резервных копий 8 дней.

8.3 РАБОТА С СЕРТИФИКАТАМИ

Для обеспечения работы ИС по защищенному протоколу https необходимо использование сертификатов:

- сертификат hrs.alrosa.ru,
- wildcard сертификат *.hrs.alrosa.ru.

8.4 ОПЕРАЦИИ С ПЕРСОНАЛЬНЫМИ ДАННЫМИ

В целях исполнения требования ИБ к ИСПДн системный администратор выполняет необходимые операции с ПДн пользователя по заявке ITSM.

Системный администратор действует по заявке ITSM и при этом не оценивает правовую сторону запроса, не контактирует с субъектом, не принимает решений о сроках и объемах обработки, выполняет только технические аспекты при выполнении запроса субъекта.

Список операций, который системный администратор выполняет по заявке в ITSM:

- операция «Поиск и идентификация данных субъекта», выполняемая по идентификаторам из запроса субъекта: ФИО, СНИЛС, email, номер договора и т.д.
- операция «Корректировка (уточнение) данных»: внесение изменение в записи по предоставленному в заявке списку, обновление дат актуализации, фиксация истории изменений.
- операция «Уничтожение данных»: удаление записей субъекта из баз данных ИС, очистка индексов и кэшей, уничтожение данных в резервных копиях, применение методов необратимого стирания.

Для выполнения перечисленных операций системный администратор обращается напрямую к БД ИС и с помощью запроса выполняет необходимую операцию.

ПРИЛОЖЕНИЕ 1. СХЕМА СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

Схема сетевого взаимодействия представлена в Таблице:

№	Описание источника *	Логин УЗ (AD)	IP адрес / FQDN источника *	Описание назначения *	IP адрес назначения *	Протокол *	Порт назначения *	Описание *	Комментарий/ задачи/ обоснование *	ответ.
1.	Open Net + VPN	-				TCP	22	SSH	доступ из открытой сети (VM станций DevOps + VPN)	-
2.	"	-				TCP	22	SSH	доступ с jmp к продуктовым серверам	-
3.		-				TCP	22	SSH	доступ с jmp к продуктовым серверам	-
4.		-				TCP	443	HTTPS	-	-
5.		-				TCP	443	HTTPS	-	-
6.		-				TCP /UDP	389,53,123, 636	LDAP, DNS, NTP, LDAPS	Коммуникация с DC	-
7.		-				TCP /UDP	389,53,123, 636	LDAP, DNS, NTP, LDAPS	Коммуникация с DC	-
8.		-				TCP	443	HTTPS	доступ с Git (в закрытой сети)	-
9.		-				TCP	443	HTTPS	доступ с Git (в	-

									закрытой сети)	
10.						TCP	443	HTTPS	работа gitlab-runner	-
11.						TCP	443	HTTPS	работа gitlab-runner	-
12.						TCP	443	HTTPS	доступ к Nexus	-
13.						TCP	443	HTTPS	доступ к Nexus	-
14.						TCP	443	HTTPS	доступ из открытой сети для пользователей и поддержки	-
15.						TCP	443	HTTPS	доступ из открытой сети для пользователей и поддержки	-
16.						TCP	443	HTTPS	доступ к сервису аутентификации	-
17.						TCP	443	HTTPS	доступ к сервису аутентификации	-
18.						TCP	443	HTTPS	доступ к сервису аутентификации	-
19.						TCP	389, 636	LDAP, LDAPS	Коммуникация с DC	-
20.						TCP	25	SMTP	Почта	-
21.						TCP	25	SMTP	Почта	-
22.						TCP/ UDP	8428, 9100	TCP	Мониторинг	-
23.						TCP/	9100	TCP	Мониторинг	-

						UDP				
24.						TCP	50000, 50301	HTTP, HTTPS	Запросы интеграционных данных. Основной протокол https, http запасной.	-
25.						TCP	50000, 50301	HTTP, HTTPS	Запросы интеграционных данных. Основной протокол https, http запасной.	-
26.						TCP	50000, 50301	HTTP, HTTPS	Запросы интеграционных данных. Основной протокол https, http запасной.	
27.						TCP	443	HTTPS	Поток данных, инициированный шиной SAP PI	
28.						TCP	443	HTTPS	Поток данных, инициированный шиной SAP PI	
29.						TCP	443	HTTPS	Поток данных, инициированный шиной SAP PI	

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

9. ПРИЛОЖЕНИЕ 2. НАСТРОЙКИ КОНФИГУРАЦИОННЫХ ФАЙЛОВ И ЛОГ-ФАЙЛОВ

В данном разделе приведены параметры конфигурации прикладных сервисов ИС HRS. Параметры собраны из `appsettings\*.json`, Docker Compose файлов развертывания, шаблонов Nginx и документации `alrosa.hrs.docs`.

Для .NET-сервисов параметры могут задаваться в JSON-конфигурации через `:` и в переменных окружения через `\_\_`. Например, `ConnectionStrings:Default` соответствует `ConnectionStrings\_\_Default`.

9.1 ОБЩИЕ ПАРАМЕТРЫ

9.1.1 ЖУРНАЛИРОВАНИЕ И ТЕЛЕМЕТРИЯ .NET-СЕРВИСОВ

Таблица 1. Параметры журналирования и телеметрии

Параметр	Назначение	Типовое значение
`OTLP_ENDPOINT_URL`	Адрес OpenTelemetry Collector для экспорта телеметрии	`http://\${OTLP_ENDPOINT}`
`Serilog__WriteTo__0__Name`	Вывод журнала в stdout контейнера	`Console`
`Serilog__WriteTo__0__Args__outputTemplate`	Шаблон строки консольного журнала	`[{Timestamp:HH:mm:ss}] {Level:u3}] [{TraceId}] [{SpanId}] {Message:l}] <s: {SourceContext}>{NewLine} {Exception}`
`Serilog__MinimumLevel__Default`	Минимальный уровень журналирования	`Information` или `Debug`
`Serilog__WriteTo__1__Name`	Файловый sink Serilog	`File`
`Serilog__WriteTo__1__Args__path`	Путь к файлам журнала внутри контейнера	`./logs/log-.log`
`Serilog__WriteTo__1__Args__rollingInterval`	Интервал ротации файлов журнала	`Day`
`Serilog__WriteTo__1__Args__formatter`	Форматтер файлового журнала	`Serilog.Formatting.Compact.CompactJsonFormatter, Serilog.Formatting.Compact`
`Serilog__WriteTo__2__Name`	OpenTelemetry sink Serilog	`OpenTelemetry`
`Serilog__WriteTo__2__Args__Endpoint`	Адрес коллектора OpenTelemetry для Serilog	`http://\${OTLP_ENDPOINT}`

`Serilog__Using__0`	Подключаемый sink Serilog	`Serilog.Sinks.Console`
`Serilog__Using__1`	Подключаемый sink Serilog	`Serilog.Sinks.File`
`Serilog__Using__2`	Подключаемая библиотека Serilog expressions	`Serilog.Expressions`
`Serilog__MinimumLevel__Override__Microsoft.AspNetCore.Mvc`	Уровень логирования MVC	`Debug`, `Information` или `Warning`
`Serilog__MinimumLevel__Override__Microsoft.AspNetCore.Routing`	Уровень логирования routing	`Debug`, `Information` или `Warning`
`Serilog__MinimumLevel__Override__Microsoft.AspNetCore.Hosting`	Уровень логирования hosting	`Debug`, `Information` или `Warning`
`Serilog__MinimumLevel__Override__Microsoft.EntityFrameworkCore`	Уровень логирования Entity Framework Core	`Warning`
`Serilog__MinimumLevel__Override__Microsoft.EntityFrameworkCore.Database.Command`	Уровень логирования SQL-команд EF Core	`Information`
`Serilog__MinimumLevel__Override__Microsoft.Extensions.Http.DefaultHttpClientFactory`	Уровень логирования HttpClientFactory	`Information`
`Serilog__MinimumLevel__Override__System.Net.Http.HttpClient.OtlpMetricExporter`	Уровень логирования OTel metric exporter	`Warning`
`Serilog__MinimumLevel__Override__System.Net.Http.HttpClient.OtlpTraceExporter`	Уровень логирования OTel trace exporter	`Warning`
`Logging__LogLevel__Default`	Базовый уровень стандартного логирования ASP.NET Core	`Information`
`Logging__LogLevel__Microsoft.AspNetCore`	Уровень стандартного логирования ASP.NET Core	`Warning`

9.1.2 ЛОГ-ФАЙЛЫ КОНТЕЙНЕРОВ

Таблица 2. Параметры лог-файлов контейнеров

Параметр	Назначение	Типовое значение
`logging.driver`	Docker logging driver	`json-file`
`logging.options.max-size`	Максимальный размер одного	

	docker log-файла	`125m`, для отдельных сервисов `100m` или `250m`
`logging.options.max-file`	Количество файлов ротации docker log	`4`, для `otel_collector` в локальном compose `5`

При включенном файловом sink Serilog прикладные логи пишутся в каталог `./logs/` внутри контейнера. При отсутствии файлового sink доступны stdout/stderr контейнера через Docker logging driver.

9.1.3 АУТЕНТИФИКАЦИЯ JWT/OIDC

Таблица 3. Параметры аутентификации

Параметр	Назначение	Типовое значение
`Authentication__Schemes__Bearer__RequireHttpsMetadata`	Требовать HTTPS при получении metadata OpenID Connect	`false`
`Authentication__Schemes__Bearer__Authority`	Authority/issuer endpoint для проверки подписи JWT	`\${AUTHENTICATION_AUTHORITY}`
`Authentication__Schemes__Bearer__ValidIssuer`	Допустимый issuer токена	`\${AUTHENTICATION_ISSUER}`
`Authentication__Schemes__Bearer__ValidAudiences__0`	Допустимая audience токена	`realm-management`
`Authentication__Schemes__Bearer__ValidAudiences__1`	Допустимая audience токена	`api`
`Authentication__Schemes__Bearer__ValidAudiences__2`	Допустимая audience токена	`account`
`Authentication__Schemes__Bearer__ValidateAudience`	Проверять audience токена; используется для integration-gateway	`false`

Примечание: в compose-файлах несколько audiences местами указаны с одним индексом `__0``. Docker Compose при нормализации `environment`` оставляет последнее значение с одинаковым именем переменной, поэтому фактически в контейнер попадает только одно значение, например `ValidAudiences__0=account``. Это работает, если токены содержат эту audience. Если требуется явно разрешить несколько audiences, для массива ASP.NET Core нужно использовать разные индексы: `__0``, `__1``, `__2``.

9.1.4 АВТОРИЗАЦИЯ ЧЕРЕЗ USERS

Таблица 4. Параметры авторизации

Параметр	Назначение	Типовое значение
`AuthClient__Protocol`	Протокол обращения к сервису Users/Auth	`http`
`AuthClient__Host`	Хост сервиса Users/Auth	`users`

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

`AuthClient__Port`	Порт сервиса Users/Auth	`8080`
`AuthClient__Prefix`	Путь получения ролей текущего пользователя	`api/v1/Users/my/roles`

9.1.5 RABBITMQ

Таблица 5. Параметры RabbitMQ

Параметр	Назначение	Dev	Prod
`RabbitMQ__Host`	Хост брокера сообщений	`rabbit-mq`	`rabbit-mq`
`RabbitMQ__Port`	AMQP-порт	`5672`	`5672`
`RabbitMQ__VirtualHost`	Виртуальный хост	`develop`	`/`
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USER}`
`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`

9.1.6 TLS И ОГРАНИЧЕНИЯ KESTREL

Таблица 6. TSL и ограничения Kestrel

Параметр	Назначение	Типовое значение
`ASPNETCORE_URLS`	Адрес и порт прослушивания ASP.NET Core	`http://0.0.0.0:80`, `http://+:80`
`ASPNETCORE_Kestrel__Certificates__Default__KeyPath`	Путь к приватному ключу TLS внутри контейнера	`/https/\${SSL_KEY}`
`ASPNETCORE_Kestrel__Certificates__Default__Path`	Путь к сертификату TLS внутри контейнера	`/https/\${SSL_CERT}`
`Kestrel__Limits__MaxRequestBodySize`	Максимальный размер тела запроса, байт	`32000000` или `500000000`

9.2 FRONTEND-СЕРВИСЫ

Настройки UI-сервисов задаются не только через переменные окружения. Переменные окружения контейнера используются в основном для шаблонов Nginx `10-variables.conf.template`: upstream API Gateway, authority redirect для `/.well-known`, OpenTelemetry и имя сервиса. Runtime-настройки самого React-приложения могут лежать в статических JSON-файлах `public/cfg/\*.json`, которые поставляются вместе со статикой. Для локальной разработки дополнительно используются `.env` и `src/setupProxy.js`.

Таблица 7. Переменные окружения

Источник	Для чего используется
Переменные окружения Docker Compose	Подстановка значений в Nginx templates при запуске контейнера
`nginx/10-variables.conf.template` или `Configs/nginx-ui*/10-variables.conf.template`	`map \$authority`, upstream `backend`, upstream UI-модулей, OpenTelemetry exporter

`nginx.conf`	Правила Nginx: маршрутизация SPA, `/api`, `/.well-known`, access/error log, cache headers
`public/cfg/auth.config.json`	Runtime-настройки OIDC-клиента React-приложения, если файл присутствует в поставке
`public/cfg/auth.config.json.example`	Шаблон OIDC-конфига, если фактический `auth.config.json` не хранится в репозитории
`public/cfg/main-page.config.json`	Runtime-конфигурация главной страницы/списка сервисов
`.env` и `src/setupProxy.js`	Только локальная разработка: проксирование `realms`, `/.well-known/openid-configuration`, `/api`

9.2.1 WEB-UI-TRAININGMANAGEMENT

Для `web-ui-trainingmanagement` параметры делятся на три группы:

- переменные окружения контейнера для Nginx;
- статические runtime-файлы `public/cfg/auth.config.json` и `public/cfg/main-page.config.json`;
- `.env` переменные локального dev-сервера React, которые используются только `src/setupProxy.js`.

Таблица 8. Параметры сервиса

Параметр	Назначение	Dev	Prod
`PORT`	Порт Nginx внутри контейнера	`80`	`80`
`API_GATEWAY`	Upstream API Gateway	`gateway`	`gateway`
`AUTHENTICATION_AUTHORITY`	Authority для UI/OIDC	`\${AUTHENTICATION_ISSUER}`	`\${AUTHENTICATION_ISSUER}`
`OTLP_ENDPOINT_URL`	Адрес OTel exporter в Nginx	`\${OTLP_ENDPOINT}`	`\${OTLP_ENDPOINT}`
`OTEL_SERVICE_NAME`	Имя сервиса в телеметрии	`Web-UI`	`UI-Training-Management`
`auth.config.json:authority`	Authority для oidc-client в UI	`/`	задается конфигом окружения
`auth.config.json:client_id`	OIDC client id	`hrs-web-ui`	задается конфигом окружения
`auth.config.json:redirect_uri`	Redirect URI после входа	`/main`	задается конфигом окружения
`auth.config.json:silent_redirect_uri`	Silent renew redirect URI	`/login`	задается конфигом окружения
`auth.config.json:automaticSilentRenew`	Автоматическое обновление токена	`true`	задается конфигом окружения

`main-page.config.json:title`	Заголовок страницы сервисов	`Сервисы`	задается файлом поставки
`main-page.config.json:items[]`	Список пунктов главной страницы, ссылки, сортировка, доступ	`/training`, `/kpi`, `/admin`	задается файлом поставки
`REACT_APP_DEV_MODE`	Режим разработки React-приложения	`true/false`	не используется в prod
`API_AUTH_REALM`	Realm/URL Keycloak для локальной сборки	`<keycloak realm url>`	не используется в prod
`API_BASE_URL`	Базовый URL API для локальной сборки	`<base api url>`	не используется в prod

9.2.2 WEB-UI-MAIN

Таблица 9. Параметры сервиса

Параметр	Назначение	Dev	Prod
`PORT`	Порт Nginx внутри контейнера	`80`	`80`
`API_GATEWAY`	Upstream API Gateway	`gateway`	`gateway`
`TRAINING_MANAGEMENT_HOST`	Upstream UI обучения	`web-ui`	`web-ui-trainingmanagement`
`AUTHENTICATION_AUTHORITY`	Authority для UI/OIDC	`\${AUTHENTICATION_ISSUER}`	`\${AUTHENTICATION_ISSUER}`
`OTLP_ENDPOINT_URL`	Адрес OTel exporter в Nginx	`\${OTLP_ENDPOINT}`	`\${OTLP_ENDPOINT}`
`OTEL_SERVICE_NAME`	Имя сервиса в телеметрии	`Web-UI-Main`	`Web-UI-Main`
`auth.config.json:authority`	Authority для oidc-client в UI	`/`	задается файлом поставки
`auth.config.json:client_id`	OIDC client id	`hrs-web-ui`	задается файлом поставки
`auth.config.json:redirect_uri`	Redirect URI после входа	`/main`	задается файлом поставки
`auth.config.json:silent_redirect_uri`	Silent renew redirect URI	`/login`	задается файлом поставки
`auth.config.json:post_logout_redirect_uri`	Redirect URI после выхода	`/realms/HRService/protocol/openid-connect/logout`	задается файлом поставки
`main-page.config.json:items[]`	Список сервисов	`/training`, `/kpi`,	задается файлом поставки

	главной страницы, маршруты и права доступа	`/admin`	
`API_AUTH_REALM`	Realm/URL Keycloak для локального проху	`<keycloak realm url>`	не используется в prod
`API_BASE_URL`	Базовый URL API для локального проху	`<base api url>`	не используется в prod

Для `web-ui-main` переменные окружения также настраивают Nginx, а параметры OIDC и главной страницы берутся из `public/cfg/\*.json`. В репозитории присутствует фактический `public/cfg/auth.config.json`.

9.2.3 WEB-UI-ADMIN

Для `web-ui-admin` env-переменные используются Nginx-контейнером. Фактический `auth.config.json` в репозитории отсутствует, есть `auth.config.json.example`; README указывает, что файл нужно добавить/отредактировать при локальной настройке и поставке.

Таблица 10. Параметры сервиса

Параметр	Назначение	Dev	Prod
`PORT`	Порт Nginx внутри контейнера	`80`	`80`
`API_GATEWAY`	Upstream API Gateway	`gateway`	`gateway`
`AUTHENTICATION_AUTHORITY`	Authority для UI/OIDC	`\${AUTHENTICATION_ISSUER}`	`\${AUTHENTICATION_ISSUER}`
`OTLP_ENDPOINT_URL`	Адрес OTel exporter в Nginx	`\${OTLP_ENDPOINT}`	`\${OTLP_ENDPOINT}`
`OTEL_SERVICE_NAME`	Имя сервиса в телеметрии	`Web-UI-Admin`	`Web-UI-Admin`
`auth.config.json.example: authority`	Шаблон authority для oidc-client	`/`	должен быть задан фактическим `auth.config.json`
`auth.config.json.example: client_id`	Шаблон OIDC client id	`hrs-web-ui`	должен быть задан фактическим `auth.config.json`
`auth.config.json.example: redirect_uri`	Шаблон redirect URI после входа	`/main`	должен быть задан фактическим `auth.config.json`
`main-page.config.json:items[]`	Список сервисов главной страницы, маршруты и права доступа	`/training`, `/kpi`, `/admin`	задается файлом поставки

`API_AUTH_REALM`	Realm/URL Keycloak для локального проху	`<keycloak realm url>`	не используется в prod
`API_BASE_URL`	Базовый URL API для локального проху	`<base api url>`	не используется в prod

9.2.4 WEB-UI-GOALS

Для `web-ui-goals` env-переменные используются Nginx-контейнером. Фактический `auth.config.json` в репозитории отсутствует, есть `auth.config.json.example`; README указывает, что файл нужно добавить/отредактировать при локальной настройке и поставке.

Таблица 11. Параметры сервиса

Параметр	Назначение	Dev	Prod
`PORT`	Порт Nginx внутри контейнера	`80`	`80`
`API_GATEWAY`	Upstream API Gateway	`gateway`	`gateway`
`AUTHENTICATION_AUTHORITY`	Authority для UI/OIDC	`\${AUTHENTICATION_ISSUER}`	`\${AUTHENTICATION_ISSUER}`
`OTLP_ENDPOINT_URL`	Адрес OTel exporter в Nginx	`\${OTLP_ENDPOINT}`	`\${OTLP_ENDPOINT}`
`OTEL_SERVICE_NAME`	Имя сервиса в телеметрии	`Web-UI-Goals`	`Web-UI-Goals`
`auth.config.json.example:authority`	Шаблон authority для oidc-client	`/`	должен быть задан фактическим `auth.config.json`
`auth.config.json.example:client_id`	Шаблон OIDC client id	`hrs-web-ui`	должен быть задан фактическим `auth.config.json`
`auth.config.json.example:redirect_uri`	Шаблон redirect URI после входа	`/main`	должен быть задан фактическим `auth.config.json`
`main-page.config.json:items[]`	Список сервисов главной страницы, маршруты и права доступа	`  '/training', '/kpi', '/admin`	задается файлом поставки
`API_AUTH_REALM`	Realm/URL Keycloak для локального проху	`<keycloak realm url>`	не используется в prod
`API_BASE_URL`	Базовый URL API для локального проху	`<base api url>`	не используется в prod
`setupProxy.js:/test`	Дополнительный	`http://hrs-	не используется в prod

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

	локальный проху к Reports API	dev.internal:10013/api`	
--	----------------------------------	-------------------------	--

9.2.5 UI-DOCS

Таблица 12. Параметры сервиса

Параметр	Назначение	Dev	Prod
`PORT`	Порт Nginx внутри контейнера	`80`	`80`

9.3 BACKEND-СЕРВИСЫ

.NET-сервисы используют стандартную модель конфигурации ASP.NET Core. Поэтому настройки задаются не только через переменные окружения: базовые значения лежат в `appsettings.json`, окруженческие переопределения в `appsettings.Development.json` или `appsettings.Develop.json`, локальные профили в `Properties/launchSettings.json`, а Docker Compose/Helm передают production/dev overrides через env. Для gateway дополнительно используется маршрутизационный файл Ocelot.

Таблица 13. Источники настроек сервисов

Источник	Для чего используется
`appsettings.json`	Базовые значения сервиса: строки подключения по умолчанию, клиенты, feature flags, уровни логирования
`appsettings.Development.json` / `appsettings.Develop.json`	Переопределения для локальной разработки
`Properties/launchSettings.json`	Переменные и URL профиля запуска для IDE/`dotnet run`; в контейнерной поставке не используется
Docker Compose `environment`	Переопределения для контейнерного запуска, секреты и адреса соседних сервисов
Helm `values*.yaml`	Переопределения для Kubernetes-развертывания, если сервис разворачивается через Helm
`ocelot.json` / mounted `ocelot.*.json`	Только gateway/integration- gateway: маршруты, downstream/upstream и authentication provider keys

Таблица 14. Настройки сервиса

Сервис	Файловые конфиги	Env overrides в развертывании	Дополнительные источники
`gateway`	`appsettings.json`, `appsettings.Development.json`, `ocelot.json`	Docker Compose/Helm	mounted `ocelot.web-api.json`, `ocelot.integration.json`, `launchSettings.json`
`training-`	`appsettings.json`,	Docker	`launchSettings.json`

management`	`appsettings.Development.json`	Compose/Helm	
`goals`	`appsettings.json`, `appsettings.Develop.json`, `appsettings.IntegrationTests.json`	Docker Compose/Helm	`launchSettings.json`, test config
`users`	`appsettings.json`, `appsettings.Development.json`	Docker Compose/Helm	`launchSettings.json`
`reports`	`appsettings.json`	Docker Compose	`launchSettings.json`
`notifications`	`appsettings.json`, `appsettings.Development.json`	Docker Compose	`launchSettings.json`
`scheduler`	`appsettings.json`, `appsettings.Development.json`	Dev Docker Compose	`launchSettings.json`
`sap-pi-ait- integration- adapter`	`appsettings.json`, `appsettings.Development.json`	Docker Compose	`launchSettings.json`, OpenAPI client spec

9.3.1 GATEWAY

Таблица 15. Параметры сервиса

Параметр	Назначение	Значение
`ASPNETCORE_URLS`	Адрес прослушивания сервиса	`http://0.0.0.0:80`, `http://+:80`
`OTLP_ENDPOINT_URL`	Адрес OpenTelemetry Collector	`http://\${OTLP_ENDPOINT}`
`TenantClient__Protocol`	Протокол обращения к сервису tenants/users	`http`
`TenantClient__Host`	Хост сервиса tenants/users	`users`
`TenantClient__Port`	Порт сервиса tenants/users	`8080`
`TenantCache__Expiration`	Время жизни кеша tenant-данных	`00:05:00`
`Kestrel__Limits__MaxRequestBodySize`	Максимальный размер тела запроса	`32000000`, в development `500000000`
`VIRTUAL_PATH`	Виртуальный путь для integration-gateway	`/integration`
`Authentication__Schemes__ Bearer__ValidateAudience`	Отключение audience validation для integration-gateway	`false`

Дополнительно подключаются общие параметры журналирования, аутентификации и TLS.

9.3.2 TRAINING-MANAGEMENT

Таблица 16. Параметры сервиса

Параметр	Назначение	Dev	Prod
`ConnectionStrings__Default`	Строка подключения к БД PostgreSQL сервиса	`Host=postgres;Port=543 2;Database=training_ma`	`\${TRAINING_MAN AGEMENT_CONNEC`

		nagement;...`	TION_STRING}`
`Serilog__MinimumLevel__Default`	Минимальный уровень журналирования из `appsettings.json`	`Information`	`Information`
`AuthClient__Protocol`	Протокол обращения к Users/Auth	`http`	общий `AuthClient`
`AuthClient__Host`	Хост Users/Auth	`users`	общий `AuthClient`
`AuthClient__Port`	Порт Users/Auth	`8080`	общий `AuthClient`
`AuthClient__Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	общий `AuthClient`

Дополнительно подключаются общие параметры журналирования, аутентификации, TLS и `OTLP\_ENDPOINT\_URL`.

9.3.3 GOALS

Таблица 17. Параметры сервиса

Параметр	Назначение	Dev	Prod
`ConnectionStrings__PostgreSQL`	Строка подключения к БД PostgreSQL сервиса Goals	`Host=postgres;Port=5432;Database=goals;...`	`\${GOALS_CONNECTION_STRING}`
`ConnectionStrings__NoPostgreSQL`	Альтернативная/устаревшая строка подключения из `appsettings.json`	`Host=hrs-dev01.ru-central1.internal;...`	не задается
`UsersClient__Protocol`	Протокол обращения к Users	`http`	`http`
`UsersClient__Host`	Хост Users	`users`	`users`
`UsersClient__Port`	Порт Users	`8080`	`8080`
`UsersClient__Prefix`	Префикс API Users	пусто или `/`	пусто
`TenantsClient__Protocol`	Протокол обращения к Tenants/Users из `appsettings.json`	`http`	не задается
`TenantsClient__Host`	Хост Tenants/Users из `appsettings.json`	`localhost`	не задается
`TenantsClient__Port`	Порт Tenants/Users из `appsettings.json`	`5100`	не задается
`TenantsClient__Prefix`	Префикс Tenants/Users из `appsettings.json`	`/`	не задается
`RabbitMQ__Host`	Хост RabbitMQ	`rabbit-mq`	`rabbit-mq`
`RabbitMQ__Port`	Порт RabbitMQ	`5672`	`5672`
`RabbitMQ__VirtualHost`	Виртуальный хост RabbitMQ	`develop`	`/`
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USER}`

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`
`MinIO__AccessKey`	Access key S3/MinIO	`\${MINIO_ACCESS_KEY}`	`\${MINIO_ACCESS_KEY}`
`MinIO__SecretKey`	Secret key S3/MinIO	`\${MINIO_SECRET_KEY}`	`\${MINIO_SECRET_KEY}`
`MinIO__ServiceURL`/ `MinIO__ServiceUrl`	Адрес MinIO	`minio`	`minio`
`MinIO__UseSsl`	Использовать SSL при обращении к MinIO	`false`	`false`
`MinIO__Port`	Порт MinIO	`9000`	`9000`
`Bucket__Name`	Имя bucket для файлов Goals	`goals-bucket`	`goals-bucket`
`CardKpiVersions__BackfillOnStartup`	Выполнять backfill версий KPI при старте	`true`	`true`
`CardKpiVersions__BackfillBatchSize`	Размер пакета backfill	`500`	`500`
`Docker__Host`	Docker daemon для development/integration tests	`tcp://develop:2375`	не задается
`Docker__TlsVerify`	Проверка TLS Docker daemon	`false`	не задается
`Kestrel__Limits__MaxRequestBodySize`	Максимальный размер тела запроса	`32000000`	`32000000`

Дополнительно подключаются общие параметры журналирования, аутентификации, авторизации и `OTLP\_ENDPOINT\_URL`.

9.3.4 USERS

Таблица 18. Параметры сервиса

Параметр	Назначение	Dev	Prod
`ConnectionStrings__Default`	Строка подключения к БД PostgreSQL сервиса Users	`Host=postgres;Port=5432;Database=Users;...`	`\${USERS_CONNECTION_STRING}`
`ConnectionStrings__Default1`	Альтернативная/устаревшая строка подключения из `appsettings.json`	`Host=hrs-dev01.central1.internal;...`	не задается
`FeatureFlags__UseEmployeesHeadsWithGapCache`	Использовать кеш руководителей с GAP	`false`	`false`
`FeatureFlags__UseEmployeesHeadsOfBusinessDirectionCache`	Использовать кеш руководителей бизнес-направлений	`false`	`false`

`RabbitMQ__Host`	Хост RabbitMQ	`rabbit-mq`	`rabbit-mq`
`RabbitMQ__Port`	Порт RabbitMQ	`5672`	`5672`
`RabbitMQ__VirtualHost`	Виртуальный хост RabbitMQ	`develop`	`/`
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USER}`
`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`
`AuthClient__Protocol`	Протокол обращения к Users/Auth	`http`	`http`
`AuthClient__Host`	Хост Users/Auth	`localhost`	`localhost`
`AuthClient__Port`	Порт Users/Auth	`8080`	`8080`
`AuthClient__Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	`api/v1/Users/my/roles`

Дополнительно подключаются общие параметры журналирования, аутентификации, TLS и `OTLP\_ENDPOINT\_URL`.

9.3.5 REPORTS

Таблица 19. Параметры сервиса

Параметр	Назначение	Dev	Prod
`HRS_HOST`	Внешний хост HRS для ссылок	`\${HRS_HOST}`	`\${HRS_HOST}`
`ConnectionStrings__PostgreSQL`	Строка подключения к БД PostgreSQL сервиса Reports	`Host=postgres;Port=5432;Database=reports;...`	`\${REPORTS_CONNECTION_STRING}`
`RabbitMQ__Host`	Хост RabbitMQ	`rabbit-mq`	`rabbit-mq`
`RabbitMQ__Port`	Порт RabbitMQ	`5672`	`5672`
`RabbitMQ__VirtualHost`	Виртуальный хост RabbitMQ	`develop`	`/`
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USER}`
`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`
`GoalsClient__Protocol`	Протокол обращения к Goals	`http`	`http`
`GoalsClient__Host`	Хост Goals	`goals`	`goals`
`GoalsClient__Port`	Порт Goals	`8080`	`8080`
`GoalsClient__Prefix`	Префикс API Goals	пусто	пусто
`UsersClient__Protocol`	Протокол обращения к	`http`	`http`

	Users		
`UsersClient__Host`	Хост Users	`users`	`users`
`UsersClient__Port`	Порт Users	`8080`	`8080`
`UsersClient__Prefix`	Префикс API Users	пусто	пусто
`AuthClient__Protocol`	Протокол обращения к Users/Auth	`http`	`http`
`AuthClient__Host`	Хост Users/Auth	`users`	`users`
`AuthClient__Port`	Порт Users/Auth	`8080`	`8080`
`AuthClient__Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	`api/v1/Users/my/roles`

Дополнительно подключаются общие параметры журналирования, аутентификации и `OTLP\_ENDPOINT\_URL`.

9.3.6 NOTIFICATIONS

Таблица 20. Параметры сервиса

Параметр	Назначение	Dev	Prod
`HRS_HOST`	Внешний хост HRS для ссылок	`\${HRS_HOST}`	`\${HRS_HOST}`
`ConnectionStrings__Default`	Строка подключения к БД PostgreSQL сервиса Notifications Строка подключения к БД PostgreSQL сервиса Notifications	`Host=postgres;Port=5432;Database=notifications_v2;...`	`\${NOTIFICATIONS_CONNECTION_STRING}`
`DeliveryDelayOptions__MinutesDelay`	Задержка доставки уведомлений, минут	`5`	`5`
`EmailNotificationOptions__Name`	Отображаемое имя отправителя	`Сервис уведомлений HRS DEV`	`Сервис уведомлений`
`EmailNotificationOptions__ServiceMail`	Адрес отправителя	`hrs-srvc@devops.alrosa.ru`	`hrs-srvc@alrosa.ru`
`EmailNotificationOptions__Password`	Пароль SMTP	`\${EmailNotificationPassword}`	задается секретом, если используются credentials
`EmailNotificationOptions__Server`	SMTP-сервер	`devops.alrosa.ru`	`msk-rpc.alrosa.ru`
`EmailNotificationOptions__Port`	SMTP-порт	`465`	`25`
`EmailNotificationOptions__Use`	Использовать SSL/TLS	`true`	не задано/по

Ssl`	SMTP		умолчанию
`EmailNotificationOptions __Use Credentials`	Использовать учетные данные SMTP	`true`	`false`
`EmailNotificationOptions __Use FakeAdapter`	Использовать fake-адаптер отправки	`true` в `appsettings.Development.json`	не задано
`Substitutions __BaseLinkUrl`	Базовый URL ссылок в уведомлениях	`https://\${HRS_HOST}`	`https://\${HRS_HOST}`
`Substitutions __HelpDeskUrl`	URL службы поддержки	`https://helpdesk.alrosa.ru/portal/sc?item=168230308506813516`	значение из окружения/конфига
`Substitutions __KpiAdmin __Full Name`	ФИО администратора KPI	`Ильиной Алине Сергеевне`	значение из окружения/конфига
`Substitutions __KpiAdmin __EmailAddress`	Email администратора KPI	`ilinaAS@alrosa.ru`	значение из окружения/конфига
`RabbitMQ __Host`	Хост RabbitMQ	`rabbit-mq`	`rabbit-mq`
`RabbitMQ __Port`	Порт RabbitMQ	`5672`	`5672`
`RabbitMQ __VirtualHost`	Виртуальный хост RabbitMQ	`develop`	`/`
`RabbitMQ __Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USERNAME}`
`RabbitMQ __Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`
`AuthClient __Protocol`	Протокол обращения к Users/Auth	`http`	`http`
`AuthClient __Host`	Хост Users/Auth	`users`	`users`
`AuthClient __Port`	Порт Users/Auth	`8080`	`8080`
`AuthClient __Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	`api/v1/Users/my/roles`
`UsersClient __Host`	Хост Users	`users`	`users`
`UsersClient __Port`	Порт Users	`8080`	`8080`
`GoalsClient __Protocol`	Протокол обращения к Goals	не задано	`http`
`GoalsClient __Host`	Хост Goals	не задано	`goals`
`GoalsClient __Port`	Порт Goals	не задано	`8080`

Дополнительно подключаются общие параметры журналирования, аутентификации, TLS и `OTLP\_ENDPOINT\_URL`.

9.3.7 SCHEDULER

Таблица 21. Параметры сервиса

Параметр	Назначение	Dev	Prod
`HRS_HOST`	Внешний хост HRS для ссылок/задач	`\${HRS_HOST}`	не найден в prod compose
`ConnectionStrings__Default`	Строка подключения к БД PostgreSQL сервиса Scheduler	`Host=postgres;Port=5432;Database=scheduler;...`	не найден в prod compose
`ASPNETCORE_URLS`	Адрес прослушивания сервиса	`http://0.0.0.0:80` по документации	не найден в prod compose
`RabbitMQ__Host`	Хост RabbitMQ	`rabbit-mq`	не найден в prod compose
`RabbitMQ__Port`	Порт RabbitMQ	`5672`	не найден в prod compose
`RabbitMQ__VirtualHost`	Виртуальный хост RabbitMQ	`develop`	не найден в prod compose
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	не найден в prod compose
`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	не найден в prod compose
`GoalsClient__Protocol`	Протокол обращения к Goals	`http`	не найден в prod compose
`GoalsClient__Host`	Хост Goals	`goals`	не найден в prod compose
`GoalsClient__Port`	Порт Goals	`8080`	не найден в prod compose
`GoalsClient__Prefix`	Префикс API Goals	пусто	не найден в prod compose
`UsersClient__Protocol`	Протокол обращения к Users	`http`	не найден в prod compose
`UsersClient__Host`	Хост Users	`users`	не найден в prod compose
`UsersClient__Port`	Порт Users	`8080`	не найден в prod compose
`UsersClient__Prefix`	Префикс API Users	пусто	не найден в prod compose
`AuthClient__Protocol`	Протокол обращения к Users/Auth	`http`	не найден в prod compose
`AuthClient__Host`	Хост Users/Auth	`users`	не найден в prod compose
`AuthClient__Port`	Порт Users/Auth	`8080`	не найден в prod compose
`AuthClient__Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	не найден в prod compose

Дополнительно подключаются общие параметры журналирования, аутентификации и `OTLP\_ENDPOINT\_URL`. В `appsettings.json` сервиса зафиксированы уровни Serilog: `Default=Information`, `Microsoft.AspNetCore.Mvc=Warning`, `Microsoft.AspNetCore.Routing=Warning`, `Microsoft.AspNetCore.Hosting=Warning`, `Microsoft.EntityFrameworkCore=Warning`.

9.3.8 SAP-PI-AIT-INTEGRATION-ADAPTER

Таблица 22. Параметры сервиса

Параметр	Назначение	Dev	Prod
`Adapter__CompanyId`	Идентификатор компании для SAP PI	`15201`	`15201`
`ConnectionStrings__Default`	Строка подключения к БД PostgreSQL адаптера	`Host=postgres;Port=5432;Database=sap_pi_ait_integration_adapter;...`	`\${SAP_PI_AIT_INTEGRATION_CONNECTION_STRING}`
`RabbitMQ__Host`	Хост RabbitMQ	`rabbit-mq`	`rabbit-mq`
`RabbitMQ__Port`	Порт RabbitMQ	`5672`	`5672`
`RabbitMQ__VirtualHost`	Виртуальный хост RabbitMQ	`develop`	`/`
`RabbitMQ__Username`	Пользователь RabbitMQ	`develop`	`\${RABBITMQ_USER}`
`RabbitMQ__Password`	Пароль RabbitMQ	`develop`	`\${RABBITMQ_PASSWORD}`
`AuthClient__Protocol`	Протокол обращения к Users/Auth	`http`	`http`
`AuthClient__Host`	Хост Users/Auth	`users`	`users`
`AuthClient__Port`	Порт Users/Auth	`8080`	`8080`
`AuthClient__Prefix`	Путь получения ролей	`api/v1/Users/my/roles`	`api/v1/Users/my/roles`
`SapPiClient__Protocol`	Протокол обращения к SAP PI	`http`	`https`
`SapPiClient__Host`	Хост SAP PI	`\${SAP_PI_HOST:-localhost}`	`\${SAP_PI_HOST}`
`SapPiClient__Port`	Порт SAP PI	`\${SAP_PI_PORT:-80}`	`\${SAP_PI_PORT}`
`SapPiClient__Prefix`	Префикс RESTAdapter SAP PI	не задано в Dev compose	`/RESTAdapter/ERP/HRS`
`SapPiClient__Username`	Пользователь SAP PI	`\${SAP_PI_USER:-user}`	`\${SAP_PI_USER}`
`SapPiClient__Password`	Пароль SAP PI	`\${SAP_PI_PASSWORD:-password}`	`\${SAP_PI_PASSWORD}`
`SapPiClient__CertificatePath`	Путь к сертификату SAP PI	не задано	`\${SAP_PI_CERT_PATH}`

`SapPiClient__Certificate`	Устаревший/закомментированный путь сертификата в `appsettings.Development.json`	`/opt/hrs/ssl/sap_pi_adapter_ait.crt`	не задано
`Kestrel__Limits__MaxRequestBodySize`	Максимальный размер тела запроса	`500000000`	`500000000`
`Serilog__MinimumLevel__Override__Microsoft`	Уровень логирования Microsoft namespace	не задано	`Warning`
`Serilog__MinimumLevel__Override__System`	Уровень логирования System namespace	не задано	`Warning`
`Serilog__MinimumLevel__Override__Alrosa.HRS.Infrastructure.HttpClient.HttpHandlers.LoggedHttpHandler`	Уровень логирования HTTP handler	не задано	`Debug`

Дополнительно подключаются общие параметры журналирования, аутентификации, TLS и `OTLP\_ENDPOINT\_URL`.

9.4 ИНФРАСТРУКТУРНЫЕ ПАРАМЕТРЫ DOCKER COMPOSE

9.4.1 ЛОКАЛЬНАЯ РАЗРАБОТКА

Таблица 23. Параметры Docker Compose

Параметр	Назначение	Типовое значение
`ASPNETCORE_ENVIRONMENT`	Окружение ASP.NET Core	`Development`
`DOTNET_ENVIRONMENT`	Окружение .NET Generic Host, если используется	`Development`
`ASPNETCORE_URLS`	Локальные URL профиля запуска	`http://localhost:<port>` или `http://+:80`
`applicationUrl`	URL из `Properties/launchSettings.json` для `dotnet run`/IDE	зависит от сервиса

9.4.2 ОБЩИЕ ПЕРЕМЕННЫЕ РАЗВЕРТЫВАНИЯ

Таблица 24. Общие параметры развертывания

Параметр	Назначение
`REGISTRY`	Docker registry с образами HRS
`PATH_TO_REPO`	Путь к каталогу конфигураций развертывания
`CERT_PATH`	Путь к каталогу сертификатов на хосте

Руководство администратора программы ««HRS Целеполагание и оценка»»	HRS
---	-----

`SSL_KEY`	Имя файла приватного ключа
`SSL_CERT`	Имя файла сертификата
`HRS_HOST`	Внешнее DNS-имя ИС HRS
`OTLP_ENDPOINT`	Адрес OpenTelemetry Collector без схемы или с портом
`AUTHENTICATION_AUTHORITY`	Authority OIDC
`AUTHENTICATION_ISSUER`	Issuer OIDC

9.4.3 ВЕРСИИ ОБРАЗОВ

Таблица 25. Версии образов

Параметр	Назначение
`UI_TRAININGMANAGEMENT_IMAGE_VERSION`	Версия образа web-ui-trainingmanagement
`UI_MAIN_IMAGE_VERSION`	Версия образа web-ui-main
`UI_ADMIN_IMAGE_VERSION`	Версия образа web-ui-admin
`UI_GOALS_IMAGE_VERSION`	Версия образа web-ui-goals
`UI_DOCS_IMAGE_VERSION`	Версия образа ui-docs
`GATEWAY_IMAGE_VERSION`	Версия образа gateway
`TRAINING_MANAGEMENT_IMAGE_VERSION`	Версия образа training-management
`SAP_PI_AIT_INTEGRATION_ADAPTER_IMAGE_VERSION`	Версия образа sap-pi-ait-integration-adapter
`GOALS_IMAGE_VERSION`	Версия образа goals
`USERS_IMAGE_VERSION`	Версия образа users
`NOTIFICATIONS_IMAGE_VERSION`	Версия образа notifications
`REPORTS_IMAGE_VERSION`	Версия образа reports
`SCHEDULER_IMAGE_VERSION`	Версия образа scheduler

9.4.4 СТРОКИ ПОДКЛЮЧЕНИЯ И СЕКРЕТЫ

Таблица 26. Строки подключения и секреты

Параметр	Назначение
`TRAINING_MANAGEMENT_CONNECTION_STRING`	Prod-строка подключения training-management
`SAP_PI_AIT_INTEGRATION_CONNECTION_STRING`	Prod-строка подключения SAP PI адаптера
`GOALS_CONNECTION_STRING`	Prod-строка подключения goals
`USERS_CONNECTION_STRING`	Prod-строка подключения users
`NOTIFICATIONS_CONNECTION_STRING`	Prod-строка подключения notifications
`REPORTS_CONNECTION_STRING`	Prod-строка подключения reports
`RABBITMQ_USER`	Пользователь RabbitMQ
`RABBITMQ_PASSWORD`	Пароль RabbitMQ

`MINIO_ACCESS_KEY`	Access key MinIO
`MINIO_SECRET_KEY`	Secret key MinIO
`EmailNotificationPassword`	Пароль SMTP для dev-окружения notifications
`SAP_PI_HOST`	Хост SAP PI
`SAP_PI_PORT`	Порт SAP PI
`SAP_PI_USER`	Пользователь SAP PI
`SAP_PI_PASSWORD`	Пароль SAP PI
`SAP_PI_CERT_PATH`	Путь к сертификату SAP PI

9.5 ИСТОЧНИКИ

Таблица 27. Источники данных

Источник	Что использовано
`alrosa.hrs.deploy/Docker/Dev/Services/docker-compose.yml`	Dev-переменные окружения прикладных сервисов
`alrosa.hrs.deploy/Docker/Prod/Services/docker-compose.yml`	Prod-переменные окружения прикладных сервисов
`docker-compose.yml`	Локальная конфигурация integration-gateway, RabbitMQ, PostgreSQL, OTel Collector
`*/appsettings*.json`	Базовые параметры сервисов, значения по умолчанию и development overrides
`alrosa.hrs.docs/docs/development/architecture/services/*/_configuration.mdx`	Документированные параметры сервисов
`alrosa.hrs.deploy/Docker/*/Configs/nginx-ui*/10-variables.conf.template`	Переменные Nginx для UI-сервисов
`alrosa.hrs.ui-*/public/cfg/*.json*`	Runtime-конфигурация UI авторизации и главной страницы, включая шаблоны `auth.config.json.example`
`alrosa.hrs.ui-*/src/setupProxy.js`	Локальные dev проху настройки UI