



Инструкция по установке
Единая система приема платежей

 Ростелеком	Инструкция по установке	
	Единая система приема платежей	Стр. 2 из 3

Содержание

1	Сборка и разворачивание системы	3
2	Вспомогательные сервисы	3
3	База данных.....	3
4	Конфигурация сервисов и приложений	3

	Инструкция по установке	
	Единая система приема платежей	Стр. 3 из 3

1 Сборка и разворачивание системы

Система развернута в отдельном пространстве в ОКД.

Процесс сборки и развертывания был реализован с помощью helm-чартов и образов приложений, которые хранятся в Gitlab Registry.

2 Вспомогательные сервисы

Помимо серверов приложений, которые являются приложениями на Spring Boot 2, внутри namespace развернуты вспомогательные сервисы:

1. Redis v7.0.9 - система управления базами данных класса NoSQL;
2. RabbitMQ v3.10.13 - брокер сообщений;
3. RabbitMQ-Exporter v1.0.0-RC19 - сбор дополнительных метрик RabbitMQ.

Для инициализации RabbitMQ необходимо добавить и включить следующие расширения: `rabbitmq_delayed_message_exchange`, `rabbitmq_management`. Кроме того, необходимо добавить рабочего пользователя RabbitMQ (`usermq`) с правами администратора, с помощью которого будет происходить подключение из серверов приложений. Инициализация внутренних объектов брокера (очередей, раздатчиков и т.д.) реализуется через конфигурацию внутри приложений, иными словами, поднимается в контексте фреймворка Spring.

3 База данных

На отдельной виртуальной машине с операционной системой РЕД ОС развернута БД Platform V Rangolin с пользователем, с помощью которого будет происходить подключение из серверов приложений. Кроме того, на самом сервере БД, необходимо добавить базу `rtkpay_db`. Миграции БД накатываются единой командой при инициализации модуля `rtkcheck-core`, с помощью средства миграции Liquibase.

Перед запуском серверов приложений (поднятием подов) требуется удостовериться, что подняты все вспомогательные сервисы и существует база `rtkpay_db`.

4 Конфигурация сервисов и приложений

Файлы с конфигурацией сервисов и приложений хранятся в объектах ОКД (`config maps`, `secrets`); они также автоматически генерируются с помощью helm-чартов.

Для каждого Spring Boot приложения создается jar-архив, который запускается в контейнере, с помощью команды: `java ${JAVA_OPTS} -jar ./app.jar`, где `${JAVA_OPTS}` это переменная среды, которая позволяет устанавливать пользовательские настройки JVM.