

Отчет по домашнему заданию №10 курса Highload Architect - Распределенные транзакции

Запуск:

git clone <https://github.com/filatkinen/distr-transactions>

cd distr-transactions

make up

(Запуск контейнеров намеренно проходит не в режиме демона, чтобы в консоль выводились сообщения от сервисов.)

Останов и очистка:

Ctrl+C

make down

Проверка:

1. Открыть дополнительную консоль
2. Перейти в папку distr-transactions/deployment
3. Запустить ./check.sh

4. Перейти в консоль с контейнерами и посмотреть на сообщения сервисов об обработке заказов. Скрипт сделан так, что отправляет заказ в цепочку сервисов, взаимодействующих через kafka. Перед каждым заказом последовательно настраиваются сервисы таким образом, что результат их обработки возвращается с ошибкой. В зависимости от того, какую роль выполняет сервис, видно как реализована обработка заказа.

Описание решения

За основу взят шаблон с сагой оркестратором, которая реализует процесс оформления регистрации заказа. Коммуникация между сервисами через брокер kafka

Сервисы:

1. Сам сервис регистрации: через REST API получает новый заказ и направляет его через брокер в адрес сервиса оркестратора
2. Сервис Workflow — реализация оркестратора, получает сообщения от сервиса регистрации и организывает коммуникацию между сервисами Резервирования товара, оплаты, нотификации
3. Сервис резервирования. Выполняется до Pivot транзакции, которую выполняет сервис Payment. В случае если Payment завершается с ошибкой, то в данном сервисе создается компенсирующая транзакция.
4. Сервис Payment — выполняет роль Pivot транзакции
5. Сервис Нотификации — выполняется он или нет, оркестратору это не важно, но он фиксирует результат его выполнения.

Схема данных:

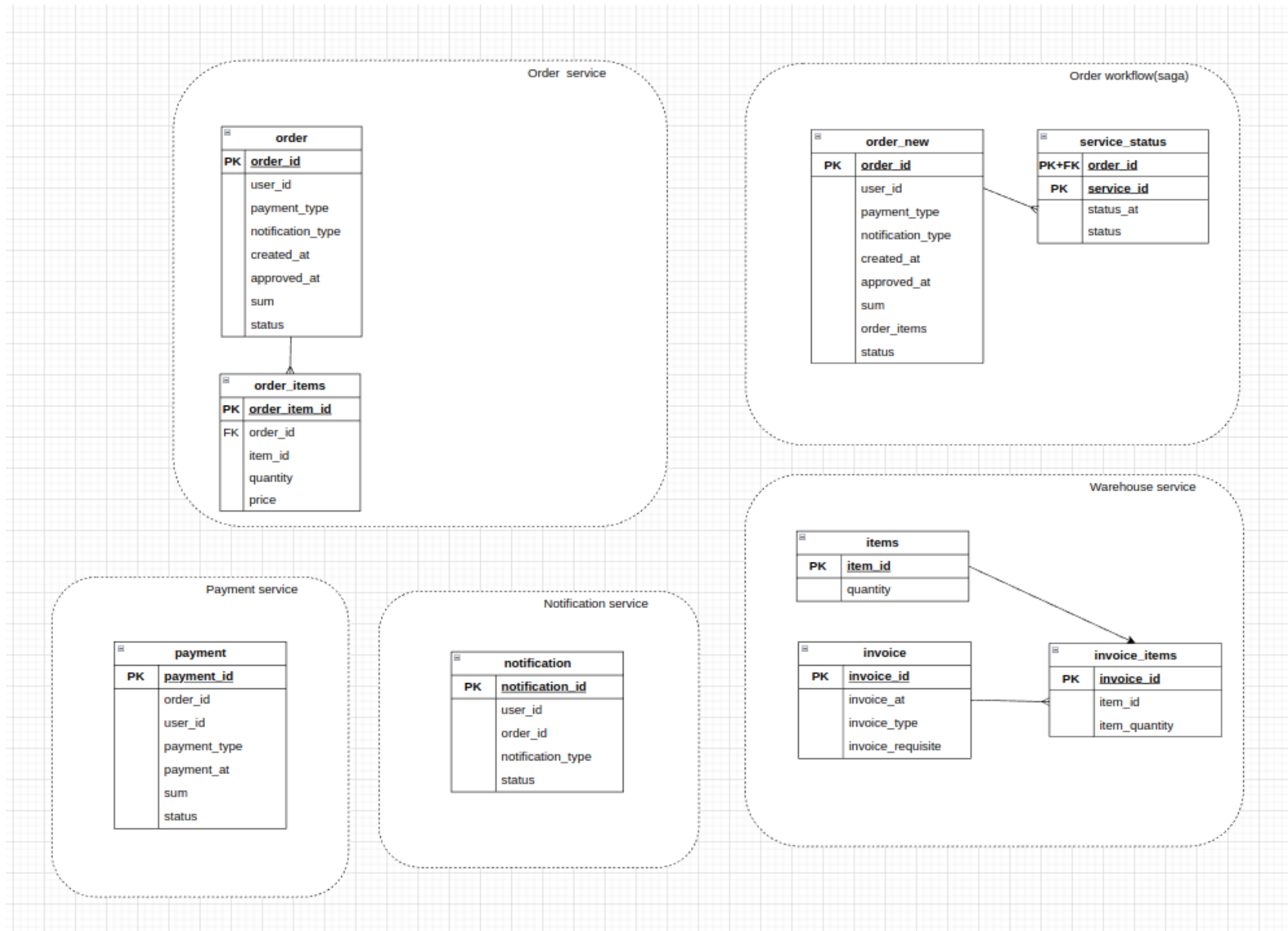


Схема взаимодействия с топиками

