DGTL Raiffeisen

Грузим в Kafka из базы: с CDC и без

Андрей Серебрянский
Streaming Platform Owner в Raiffeisen Bank

Потоковые данные

Пара примеров

Event Streaming Platform

Apache Kafka 

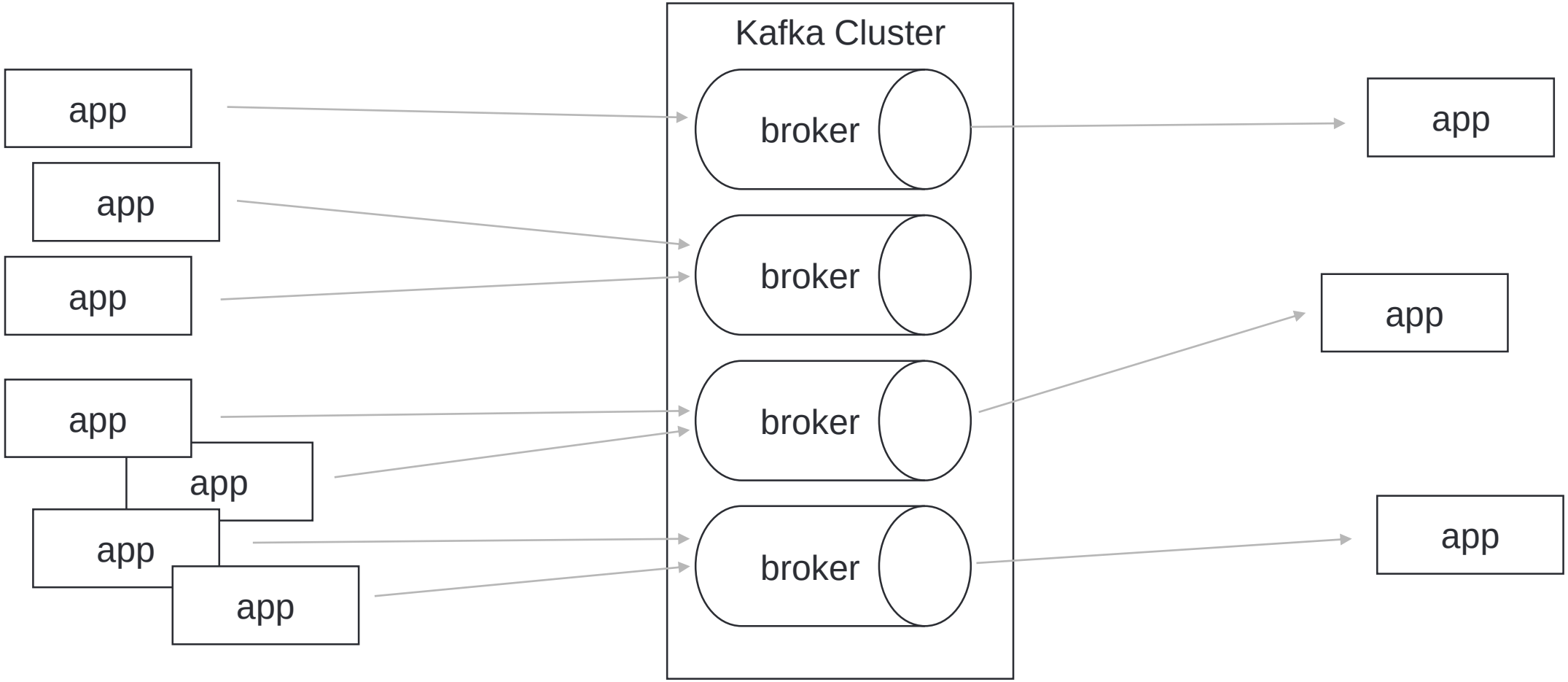
De facto стандарт в индустрии

Распределенный, реплицируемый лог сообщений

Apache Kafka

De facto стандарт в индустрии

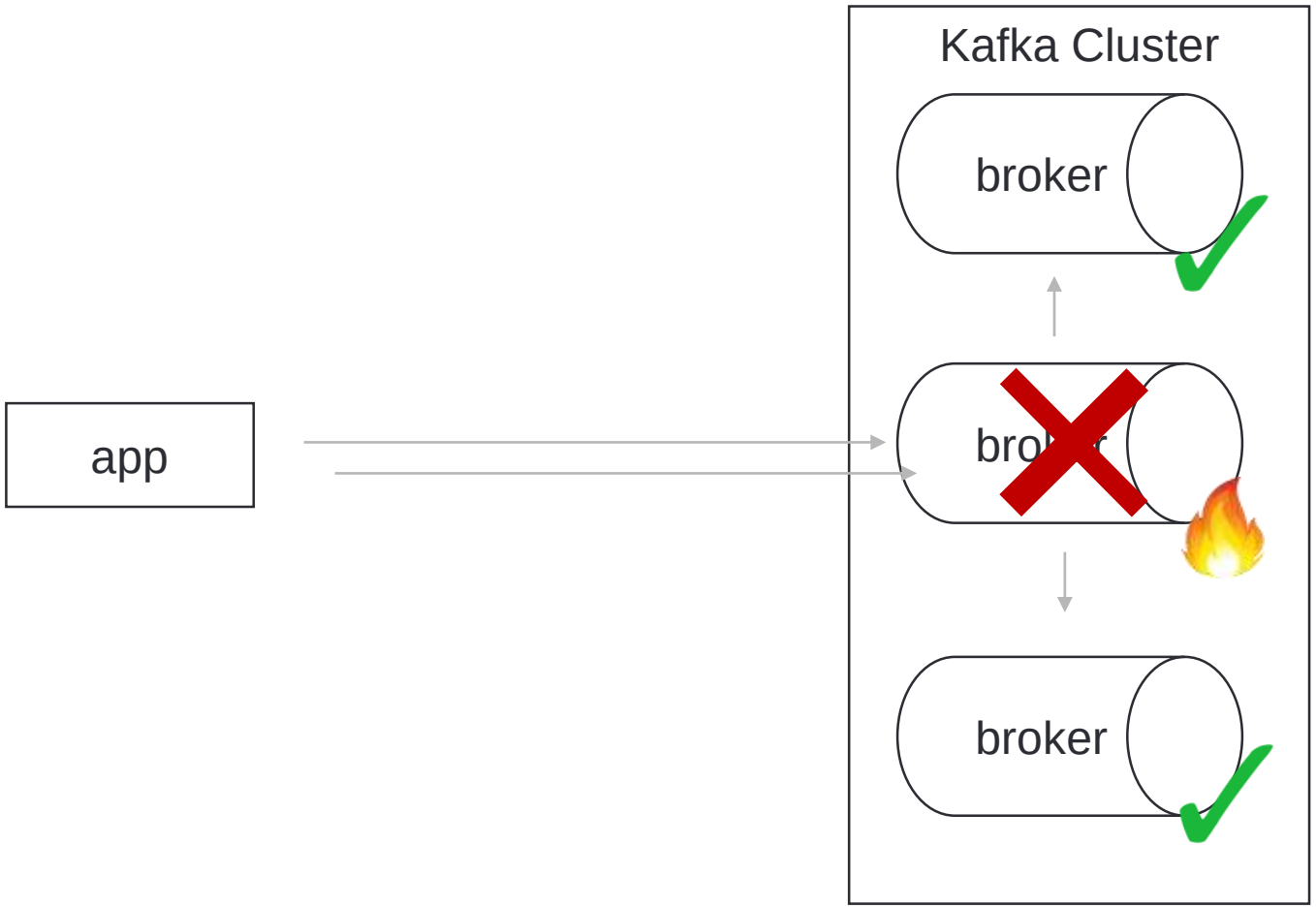
Распределенный, реплицируемый лог сообщений



Apache Kafka

De facto стандарт в индустрии

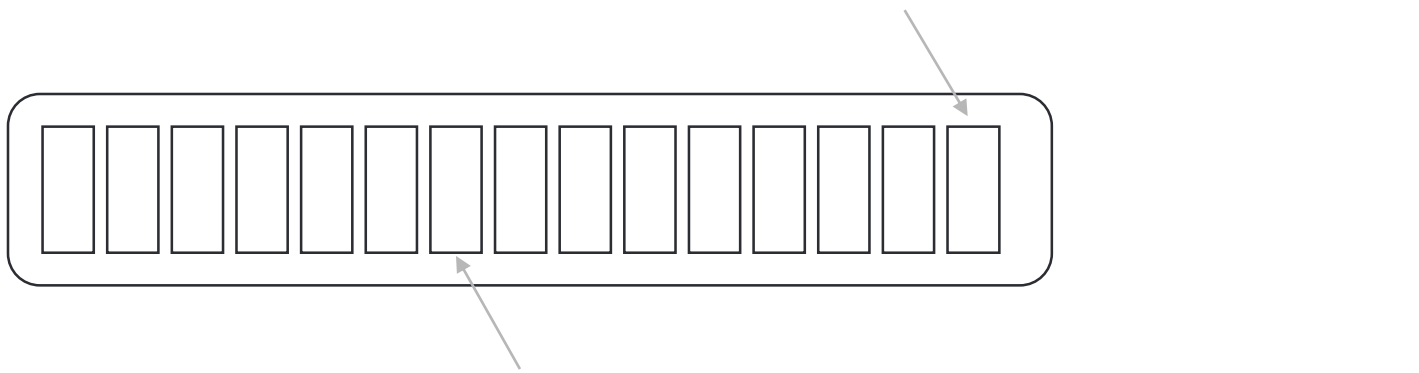
Распределенный, **реплицируемый** лог сообщений



Apache Kafka

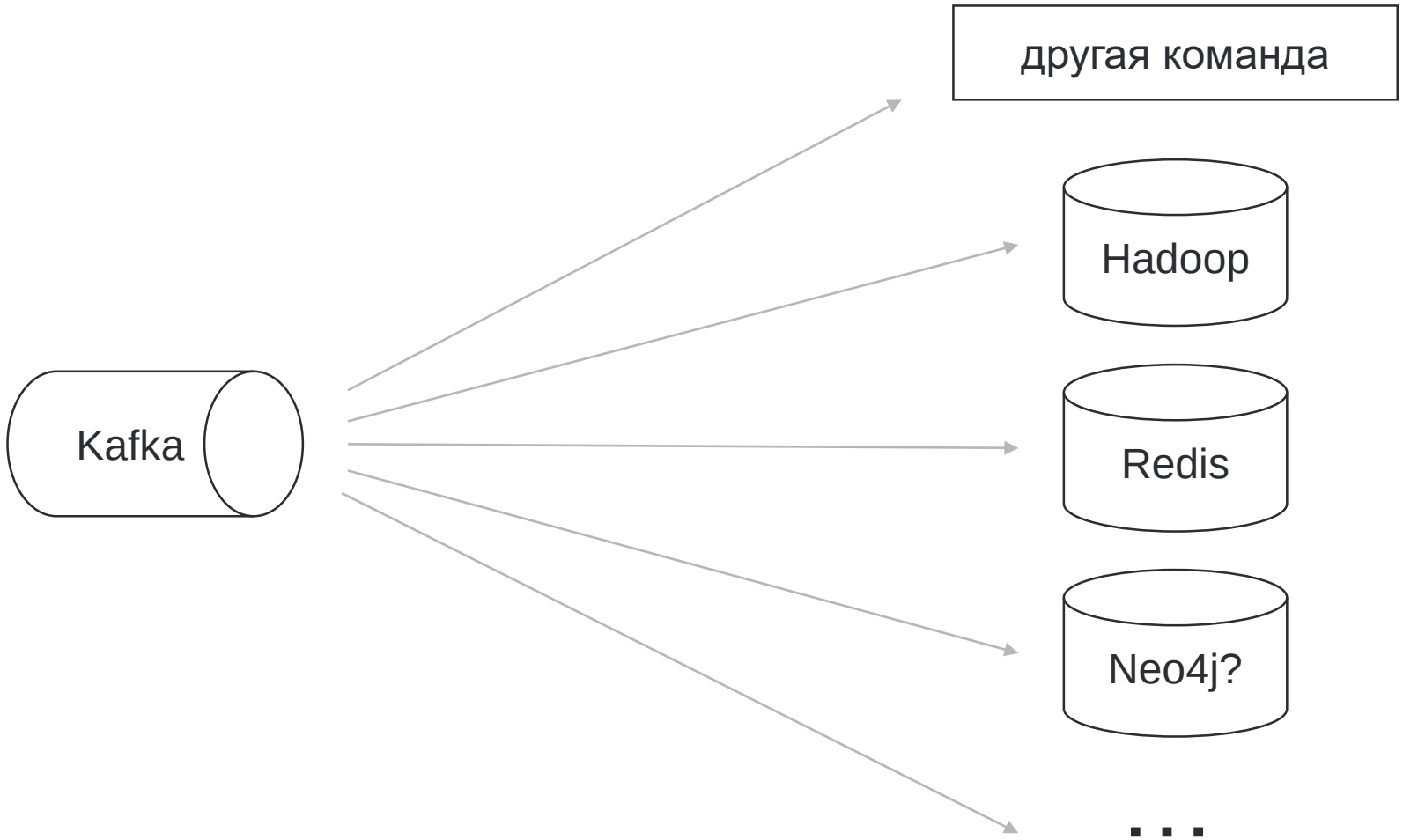
De facto стандарт в индустрии

Распределенный, реплицируемый **лог сообщений**



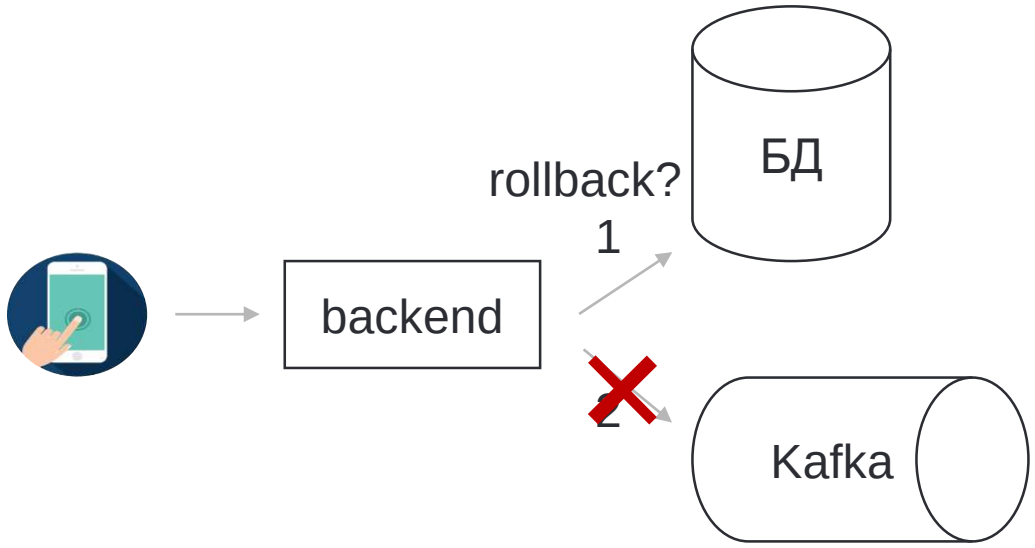
Как нам загрузить данные в Kafka?

Зачем нам данные в Kafka?



Гипотеза 1

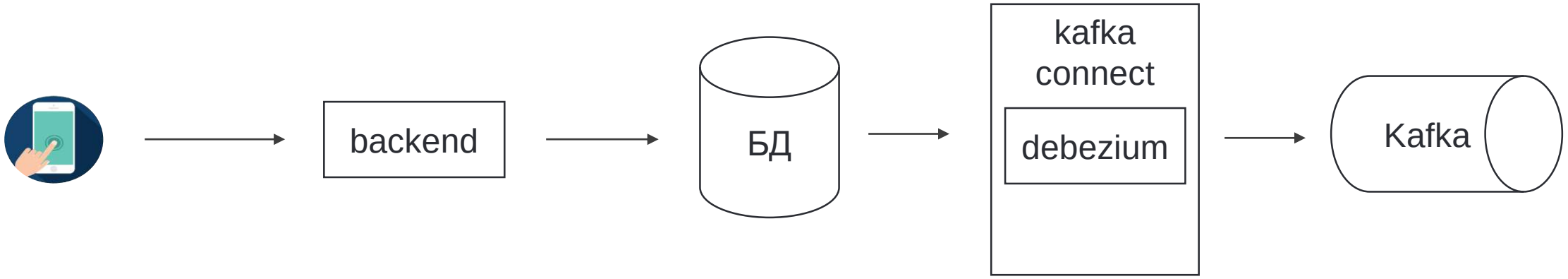
После записи в базу, отправляем сообщение в Kafka



1. Это дольше для пользователя
2. Это требует распределенной транзакции

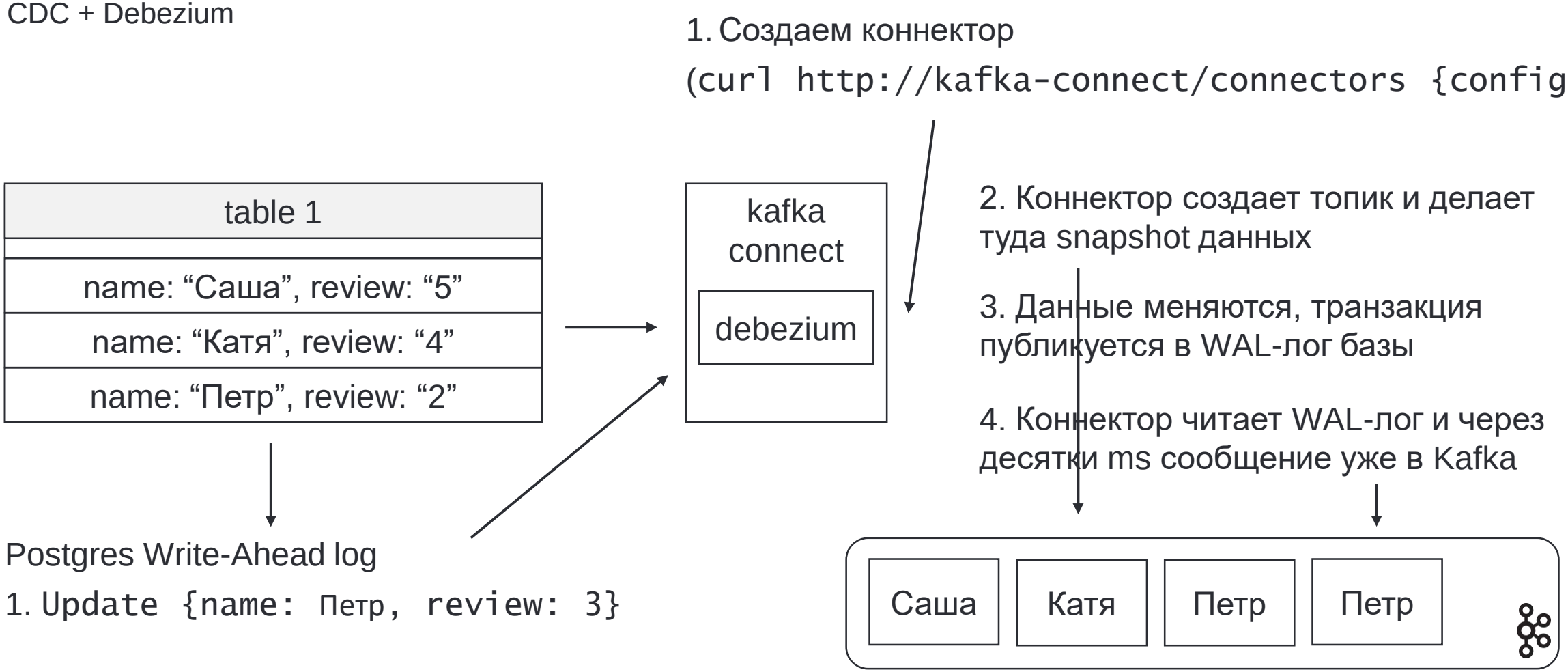
Гипотеза 2

CDC + Debezium



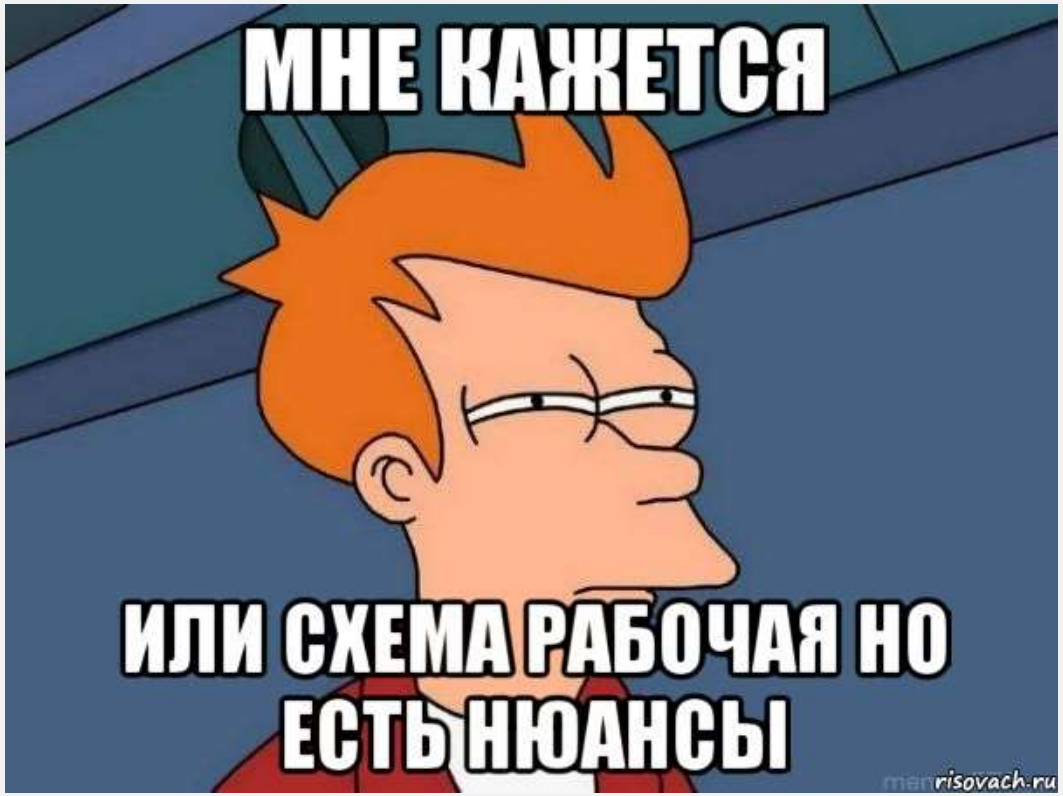
Гипотеза 2

CDC + Debezium



CDC – Change Data Capture. Захват изменений в данных

Выглядит круто, в чем подвох?



Debezium connector

Для Postgres

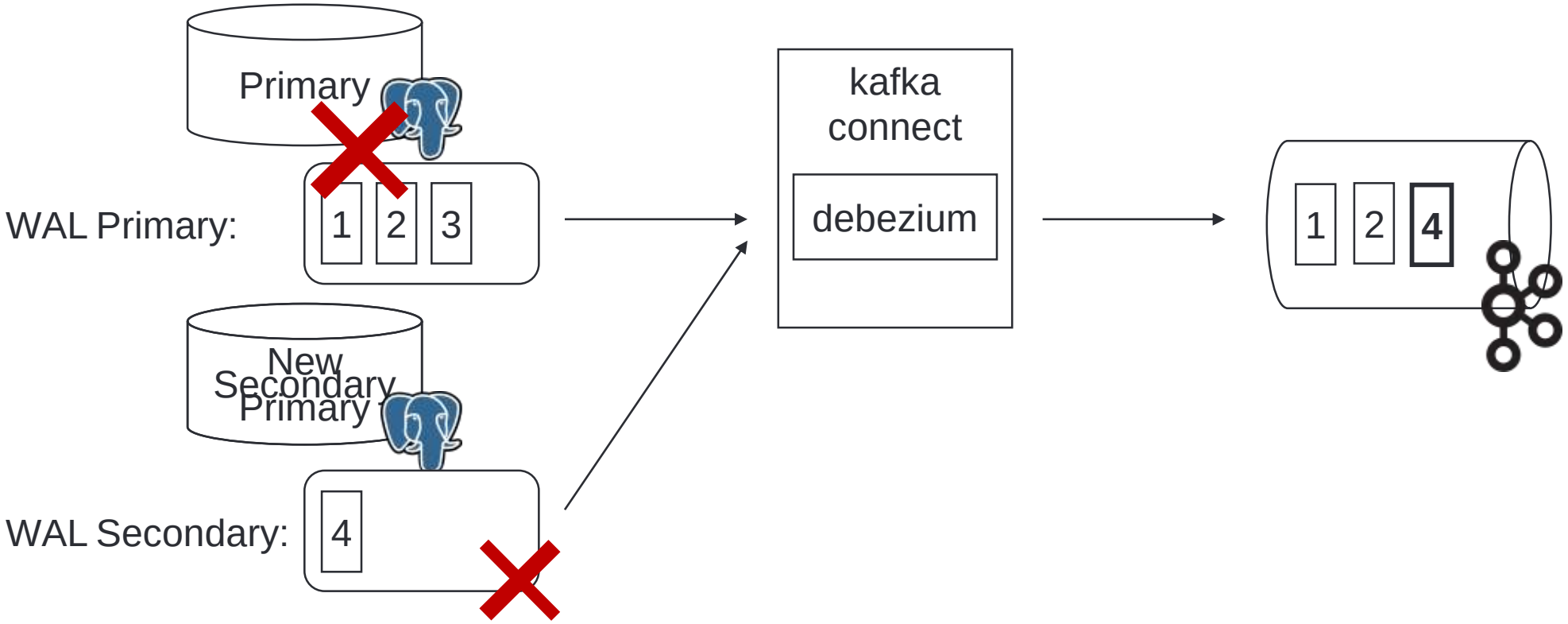
Довольно легко поднять из коробки (https://debezium.io/documentation/reference/stable/connectors/postgresql.html)

Но есть нюансы!

Нюанс: отказоустойчивость

Debezium connector

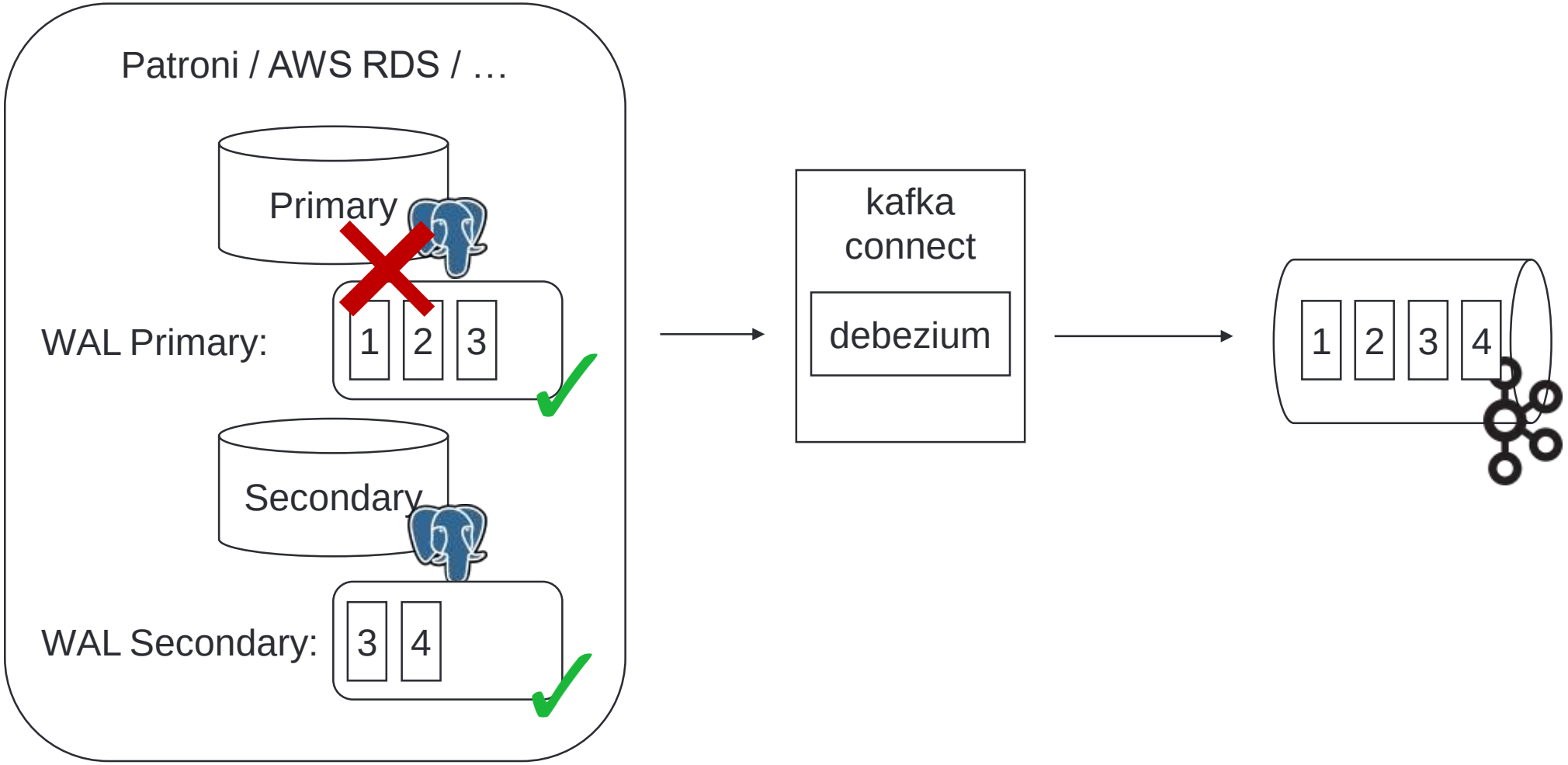
1. Репликация может быть включена только на Primary ноде!
2. Непрочитанные WAL-логи с primary надо перенести **вручную**



Подробнее о проблеме: https://debezium.io/documentation/reference/stable/connectors/postgresql.html#postgresql-cluster-failures

Нюанс: отказоустойчивость (решение)

Debezium connector



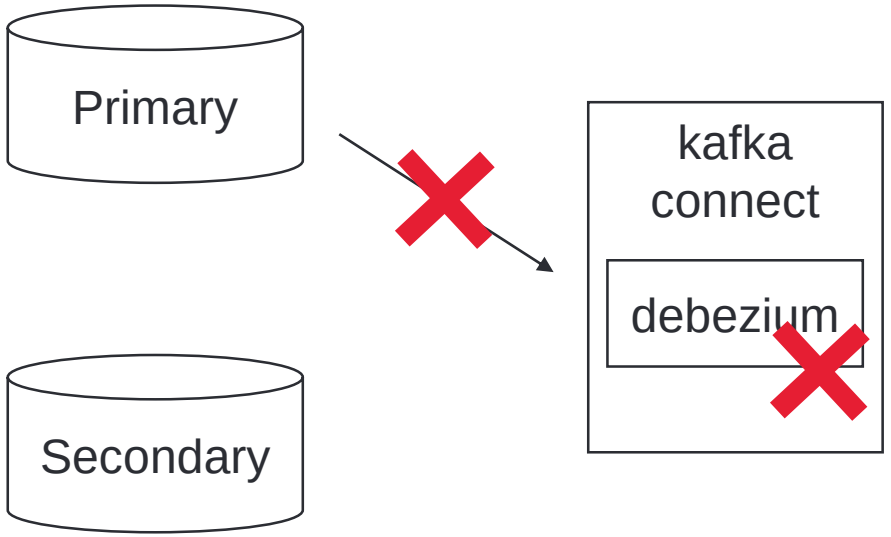
Нюанс: отказоустойчивость

Debezium connector

```
"database.hostname": ”jdbc:postgre   //s-msk-t-psql-sdp-
1.raiffeisen.ru:5432,s-msk-t-psql
2.raiffeisen.ru:5432/streaming_   targ   ServerType=master”
```

Так не сработает.

Нужен только один хост - мастера

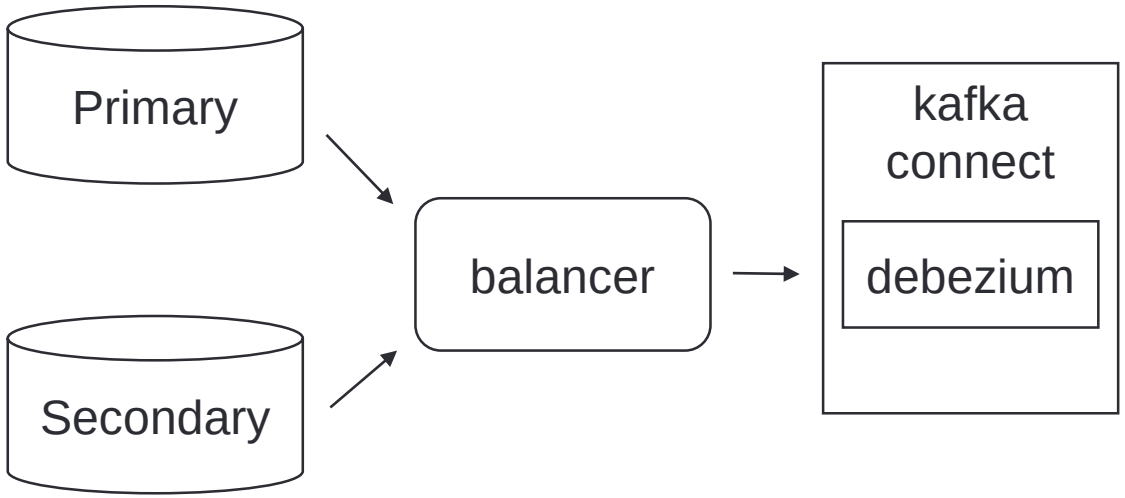



```
database.hostname: my-primary-host.raif.ru
```

Нюанс: отказоустойчивость (решение)

Debezium connector

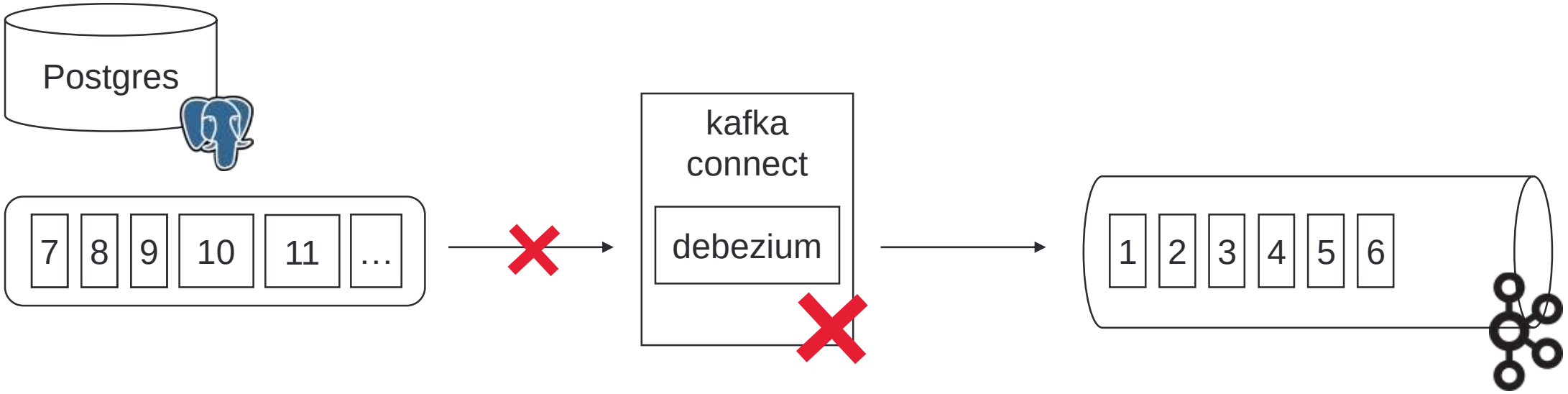
Используйте балансер перед кластером Postgres)




```
database.hostname: my-balancer.raif.ru
```

Нюанс: риск сломать базу

Debezium connector



Решение: 24/7 мониторинг, готовый Disaster Recovery plan

Нюанс: как добавить новую таблицу

Debezium connector

Создаем коннектор:

```
"table.include.list": "your_table",
"snapshot.mode": "initial"
```

-> Загружает snapshot, начинает отправку обновлений

Update connector:

```
"table.include.list": "your_table, new_table",
"snapshot.mode": "initial"
```

-> **Не** загружает snapshot, начинает отправку обновлений для обеих таблиц

Update connector:

```
"table.include.list": "your_table, new_table",
"snapshot.mode": "always"
```

-> Загружает snapshot обеих таблиц каждый рестарт

Решение: создавать новый коннектор при добавлении новых таблиц / смириться с дубликатами

А если нет CDC?

Как это нет CDC?

- база очень старая

- CDC не работает в отказоустойчивом сетапе

- вендор CDC ушел из России (IBM replicator, Oracle Golden Gate)

А данные в Kafka все равно нужны!

Решение есть: микробатчи!

1. 12:05 – первый запуск коннектора. Выгружаем все ряды с `updated_at < 12:05`

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	4	12:03
Петр	3	12:02

Решение есть: микробатчи!

1. 12:05 – первый запуск коннектора. Выгружаем все ряды с `updated_at < 12:05`
2. 12:10 – второй запуск коннектора. Выгружаем все ряды с `updated_at > 12:03 AND updated_at < 12:10`
3. 12:15 – третий запуск коннектора и тд.

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	4	12:03
Петр	**4**	**12:06**

Confluent JDBC Source Connector

Задаем

- poll.intervall.ms – интервал, с которым мы будем поллить базу
- table.include.list – список таблиц, которые надо забирать
- mode – batch / timestamp / timestamp + incremental

https://docs.confluent.io/kafka-connectors/jdbc/current/source-connector/overview.html

Confluent JDBC Source Connector

Режимы работы

Batch

Раз в 30 секунд:

SELECT * FROM table_name;

Timestamp

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at > last_loaded_ts;

Timestamp + incrementing

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at >= last_loaded_ts && pk > last_loaded_pk;

Нюансы!

JDBC source connector

Batch

Раз в 30 секунд:

SELECT * FROM table_name;

Timestamp

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at **>** last_loaded_ts;

AT MOST ONCE!

ВОЗМОЖНА ПОТЕРЯ ДАННЫХ

Timestamp +incrementing

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at >= last_loaded_ts && pk > last_loaded_pk;

Нюансы!

В чем проблема timestamp mode?

2. 12:10 – изменилось 2 ряда. Выгружаем все ряды с `updated_at > 12:03`

Коннектор записывает Катю и сохраняет прогресс: `{last_loaded_updated_at: 12:06}`.
Затем сервер перезагружается, коннектор падает

3. 12:15 – коннектор починили. Выгружаем все ряды с `updated_at > 12:06`

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	**3**	**12:06**
Петр	**4**	**12:06**

Нюансы!

В чем проблема timestamp mode?

2. 12:10 – изменилось 2 ряда. Выгружаем все ряды с `updated_at > 12:03`

Коннектор записывает Катю и сохраняет прогресс: `{last_loaded_updated_at: 12:06}`.
Затем сервер перезагружается, коннектор падает

3. 12:15 – коннектор починили. Выгружаем все ряды с `updated_at > 12:06`

Теряем update оценки Пети

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	3	12:06
Петр	**4**	**12:06**

Нюансы!

JDBC source connector

Batch

Раз в 30 секунд:

SELECT * FROM table_name;

Timestamp

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at **>** last_loaded_ts;

AT MOST ONCE!

ВОЗМОЖНА ПОТЕРЯ ДАННЫХ

Timestamp +incrementing

Раз в 30 секунд:

SELECT * FROM table_name

WHERE updated_at >= last_loaded_ts && pk > last_loaded_pk;

Нюансы!

Как timestamp + increment нам поможет?

2. 12:10 – изменилось 2 ряда. Выгружаем все ряды с `updated_at > 12:03`

✓ Коннектор записывает Катю и сохраняет прогресс:
`{last_loaded_updated_at: 12:06, last_loaded_pk: Катя}`.

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	**3**	**12:06**
Петр	**4**	**12:06**

Нюансы!

Как timestamp + increment нам поможет?

2. 12:10 – изменилось 2 ряда. Выгружаем все ряды с `updated_at > 12:03`

✓ Коннектор записывает Катю и сохраняет прогресс:
`{last_loaded_updated_at: 12:06, last_loaded_pk: Катя}.`

3. 12:15 – коннектор починили. Выгружаем все ряды с
`updated_at >= 12:06 AND pk > “Катя”`

Table: participants_reviews		
name	review	updated_at
Саша	5	12:00
Катя	**3**	**12:06**
Петр	**4**	**12:06**

А если нет ни CDC ни timestamp?

Какая перед нами задача?

- База данных без CDC (например, Oracle)
- В таблице нет поля, которое заполняется при изменении
- Изменения из таблицы все равно нужны в Kafka!

Table: participants_reviews	
name	review
Саша	5
Катя	3
Петр	4

Решение: пишем свой коннектор

Насколько это сложно?

- заимплементить два интерфейса:
  - SourceConnector
  - SourceTask
- сделать класс конфига коннектора
- сбилдить
- добавить jar-ник в образ kafka connect

Детальная инструкция: https://docs.confluent.io/platform/current/connect/devguide.html#connector-developer-guide

Talk is cheap

Show me the code!

А если не Hello World?)

Как деплоить в Kafka Connect

REST интерфейс:

- POST /connectors – создать новый коннектор
- POST /connectors/{{ connector_name }}/config – обновить конфиг

1. Найти изменившиеся конфиги

```
git diff --name-only HEAD~1 HEAD --diff-filter=AM
```

2. Для каждого изменившегося конфига отправить запрос на изменение

```
curl --fail-with-body -X POST $CONNECT_URL/connectors \
   -H 'Content-Type: application/json' \
   -d "$JSON"
;;
```

Как безопасно указать секреты

```
{
  “database.user”: “admin”,
  “database.password: “admin”
}
```

```
{
  “database.user”: “${file:path/to/file:username}”,
  “database.password: “${vault:path/to/secret: password}”
}
```

Больше про file config provider - https://docs.confluent.io/platform/current/connect/security.html#fileconfigprovider

Vault provider - https://github.com/jcustenborder/kafka-config-provider-vault

Как мониторить?

Prometheus exporter + alert rules

Алерт, если коннектор упал:

```
sum by (connector)
(sum_over_time(kafka_connect_connect_worker_metrics_connector_failed_task_count
{connector=~”your_connector_name"}[10m])) > 3
```

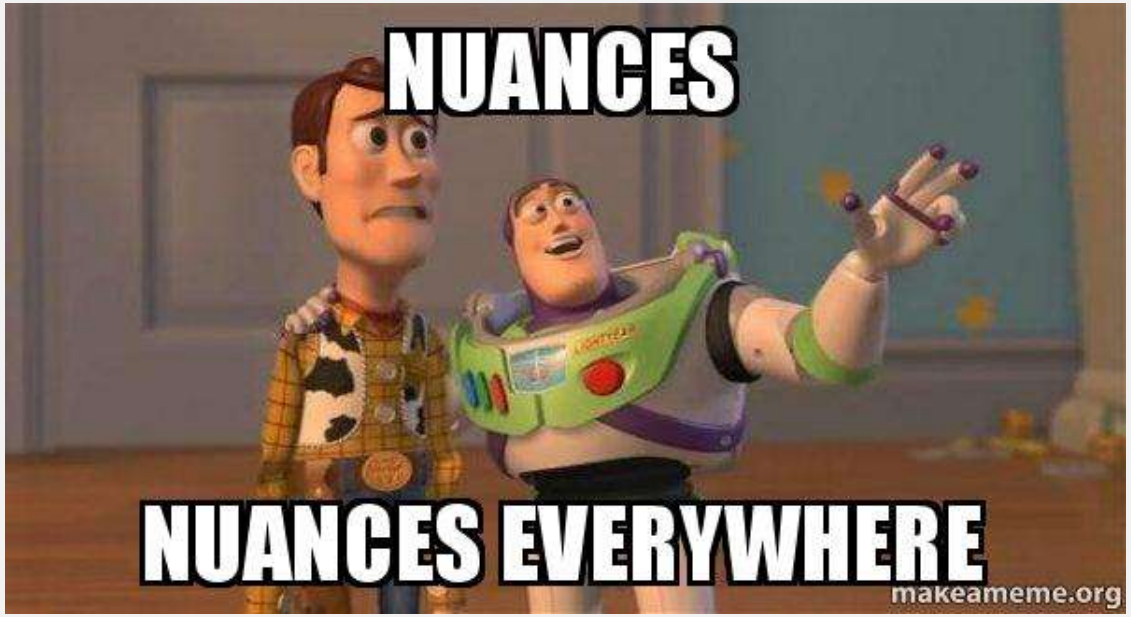
Алерт, если у коннектора нет тасков:

```
sum by (connector)
(kafka_connect_connect_worker_metrics_connector_total_task_count{
connector=~" connector "}) == 0
```

+ еще десятки других метрик

Все метрики тут: https://docs.confluent.io/platform/current/connect/monitoring.html#use-jmx-to-monitor-kconnect

Круто! Теперь будет работать?



Что еще надо учесть?

- at least once – при сбоях будут дубли

Что еще надо учесть?

- at least once – при сбоях будут дубли
- schema evolution

По умолчанию в Schema Registry стоит проверка Backward Compatibility

Например, добавление обязательной колонки – несовместимая операция.
Поставка данных остановится.

Больше про совместимость схем: https://docs.confluent.io/platform/current/schema-registry/avro.html#compatibility-types

Что еще надо учесть?

- at least once – при сбоях будут дубли
- schema evolution
- разные метаданные

Debezium:

```
{
  "after": {...},
  "source": {
    "version": "1.9.3.Final",
    ....
    "db": "rci_preview_msk",
    "table": "t_payments",
    "txId": 217464612,
    "lsn": 749299044504
  },
  "op": "u",
  "ts_ms": 1674831512588
}
```

Golden Gate:

```
{
  "table": "TCTDBS.AVRO_AUTHORIZATIONS",
  "op_type": "U",
  "op_ts": "2022-06-16 13:41:04.000151",
  "current_ts": "2022-06-16 16:41:12.089003",
  "pos": "00000064500005678870",
  "SERNO": 853463,
  ”AMOUNT": "14101.802"
}
```

jdbc source connector: no meta

Что еще надо учесть?

- at least once – при сбоях будут дубли
- schema evolution
- разные метаданные

Решение: еще одно приложение!

Что еще надо учесть?

- at least once – при сбоях будут дубли
- schema evolution
- разные метаданные

Решение: еще одно приложение!

Мы сделали на Kafka Streams.

- убирает дубли
- проверяет схему и делает эволюцию безопасной
- унифицирует метаданные
- тема для отдельного доклада

DGTL Raiffeisen

Спасибо!

https://t.me/aserebryanskiy

a.serebrianskiy@gmail.com