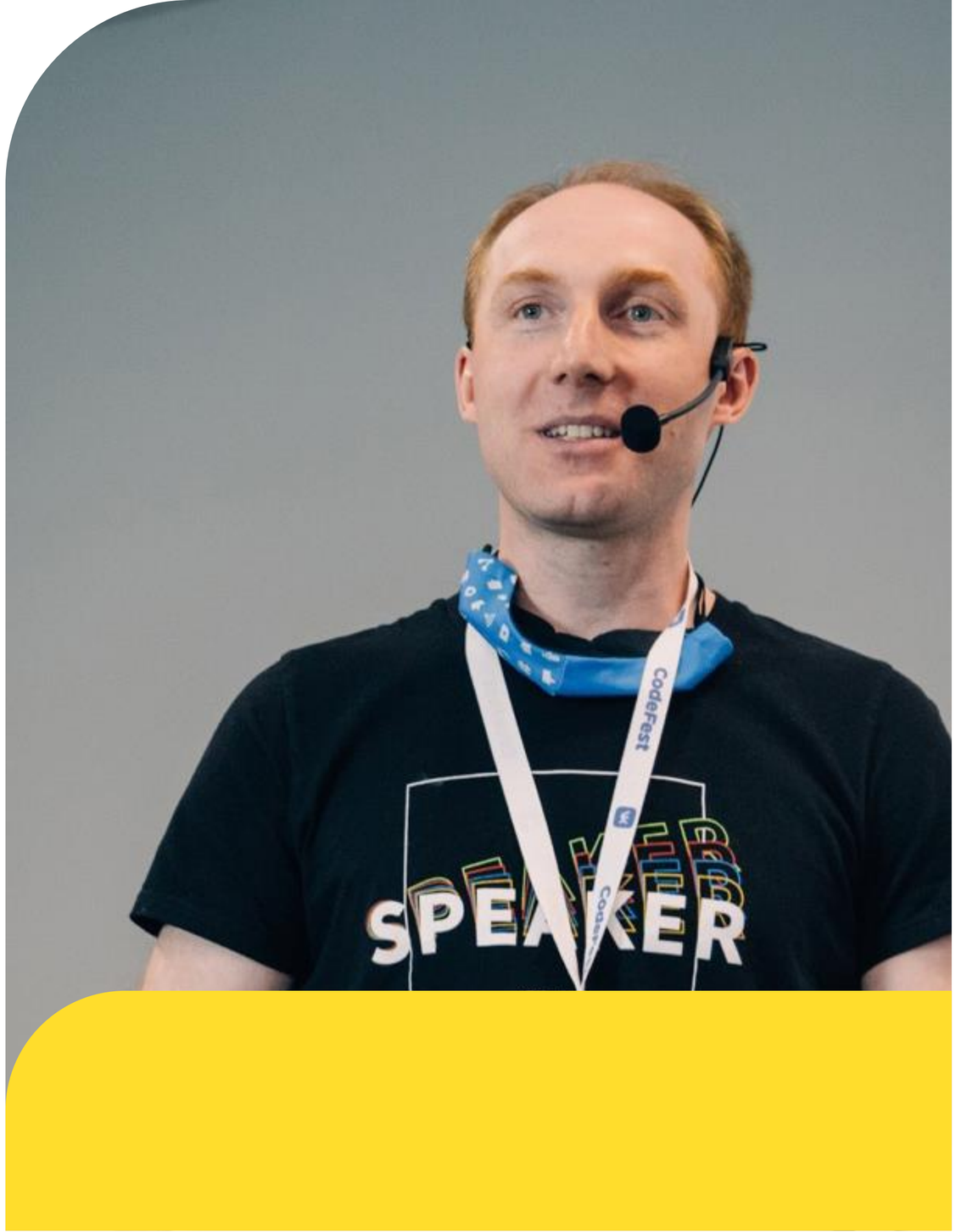
Zero-downtime deployment и базы данных

Андрей Цветцих

Т БАНК



Андрей Цветцих

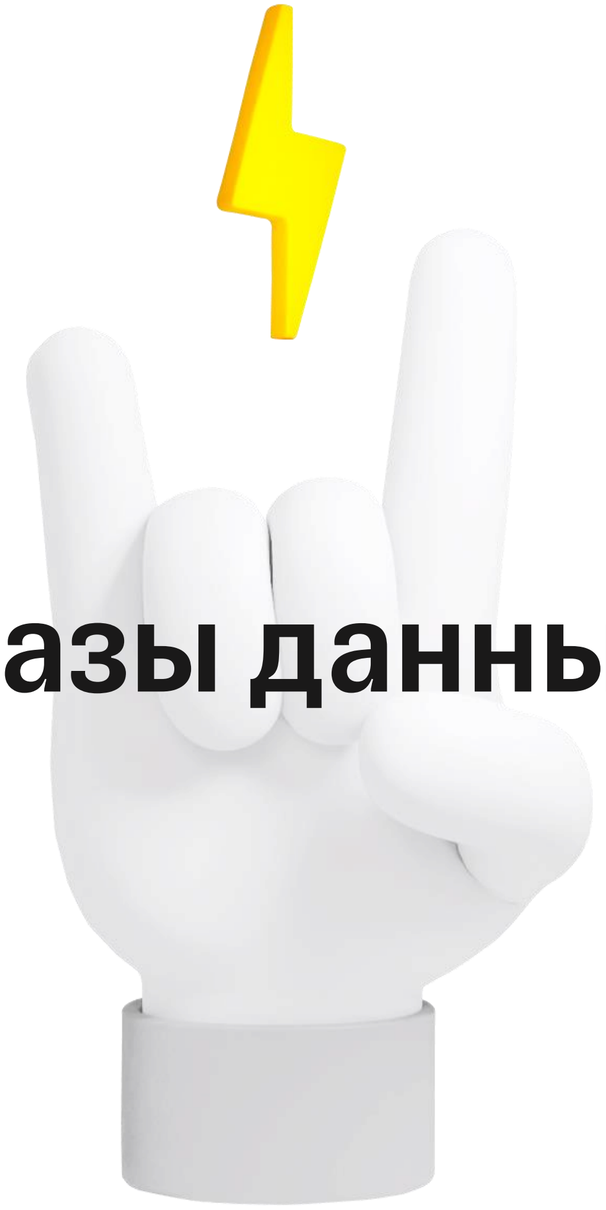
Ведущий разработчик

DevBrothers

Andrew.Tsw@gmail.com

@AndrewTsw

Zero-downtime deployment



Базы данных

Для кого этот доклад

- Вы разрабатываете микросервисы

Для кого этот доклад

- Вы разрабатываете микросервисы
- И у вас запущено более одного инстанса сервиса

Для кого этот доклад

- Вы разрабатываете микросервисы
- И у вас запущено более одного инстанса сервиса
- Вы хотите разрабатывать микросервисы

Для кого этот доклад

- Вы разрабатываете микросервисы
- И у вас запущено более одного инстанса сервиса
- Вы хотите разрабатывать микросервисы
- Вам понравилось название доклада

Пример релиза новой версии

- В новой версии переименовали колонку

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

Пример релиза новой версии

В новой версии переименовали колонку

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

Пример релиза новой версии

В новой версии переименовали колонку

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

Пример релиза новой версии

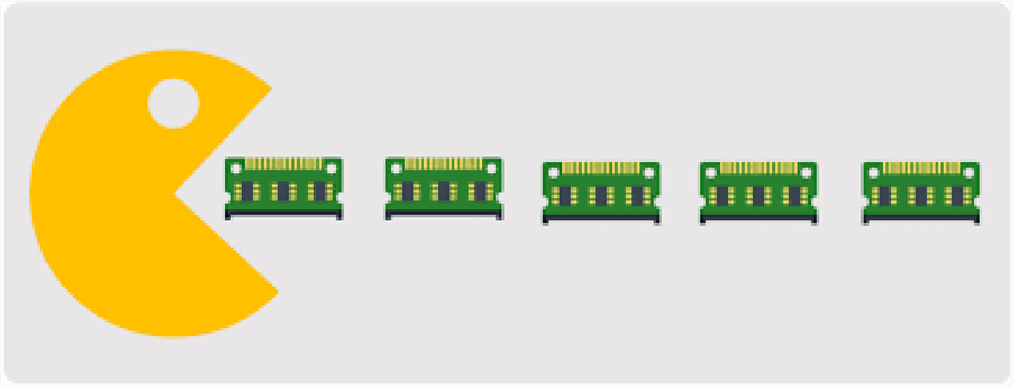
В новой версии переименовали колонку

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

Тёмной-претемной ночью после релиза

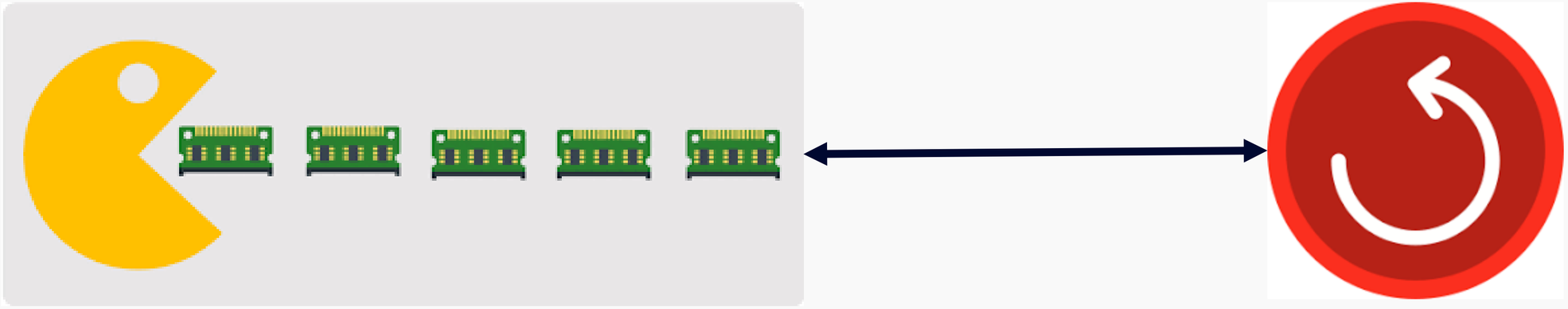
Тёмной-претемной ночью после релиза

- При запуске фоновой задачи для интеграции со сторонним сервисом – наш сервис начинает потреблять много памяти (оказался memory leak)



Тёмной-претемной ночью после релиза

При запуске фоновой задачи для интеграции со сторонним сервисом – наш сервис начинает потреблять много памяти (оказался memory leak)



Сервис выедает всю доступную память и перезапускается

Тёмной-претемной ночью после релиза

Тёмной-претемной ночью после релиза

Тёмной-претемной ночью после релиза

Сервис сломан

Старая версия сервиса не может работать с новой колонкой

Сервис сломан

Старая версия сервиса не может работать с новой колонкой

При импорте данных постоянно возникают ошибки

Сервис сломан

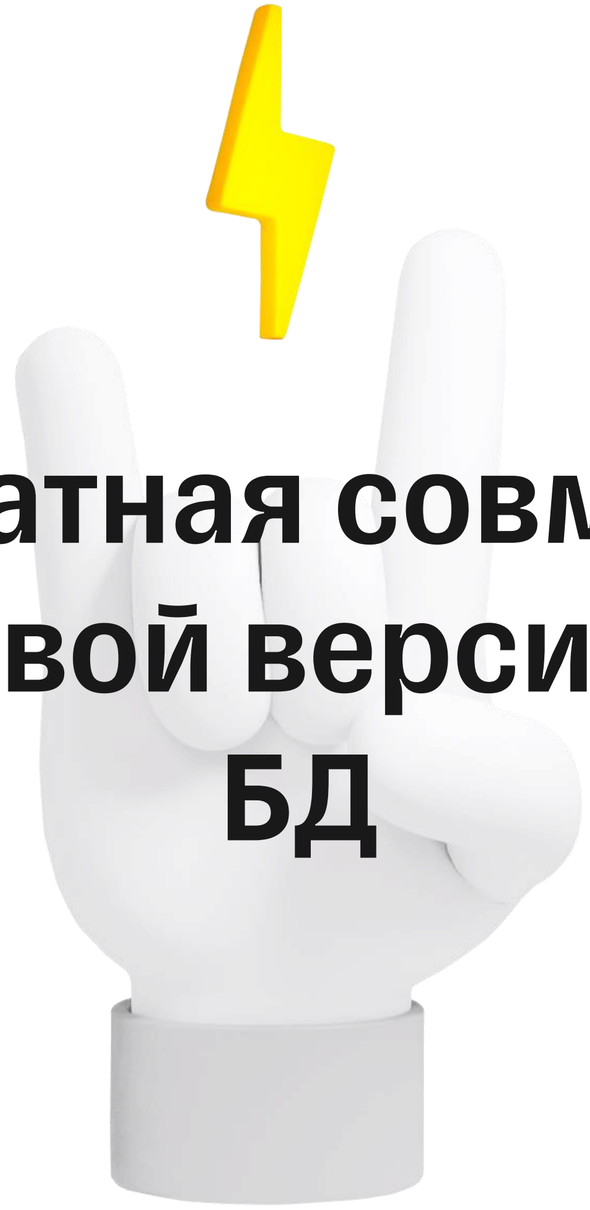
Старая версия сервиса не может работать с новой колонкой

При импорте данных постоянно возникают ошибки

SRE не могут самостоятельно исправить проблему – пора будить разработчиков

Сервис сломан

- Старая версия сервиса не может работать с новой колонкой
- При импорте данных постоянно возникают ошибки
- SRE не могут самостоятельно исправить проблему – пора будить разработчиков



Нужна обратная совместимость старой и новой версий на уровне БД

О чем поговорим:

01

Какие виды деплоя кода бывают

О чем поговорим:

01

Какие виды деплоя кода бывают

02

Как запускать миграции

О чем поговорим:

01

Какие виды деплоя кода бывают

02

Как запускать миграции

03

Как мигрировать данные на новую версию

О чем поговорим:

01

Какие виды деплоя кода бывают

02

Как запускать миграции

03

Как мигрировать данные на новую версию

04

Как обеспечить обратную совместимость для БД - примеры

О чем поговорим:

01

Какие виды деплоя кода бывают

02

Как запускать миграции

03

Как мигрировать данные на новую версию

04

Как обеспечить обратную совместимость для БД - примеры

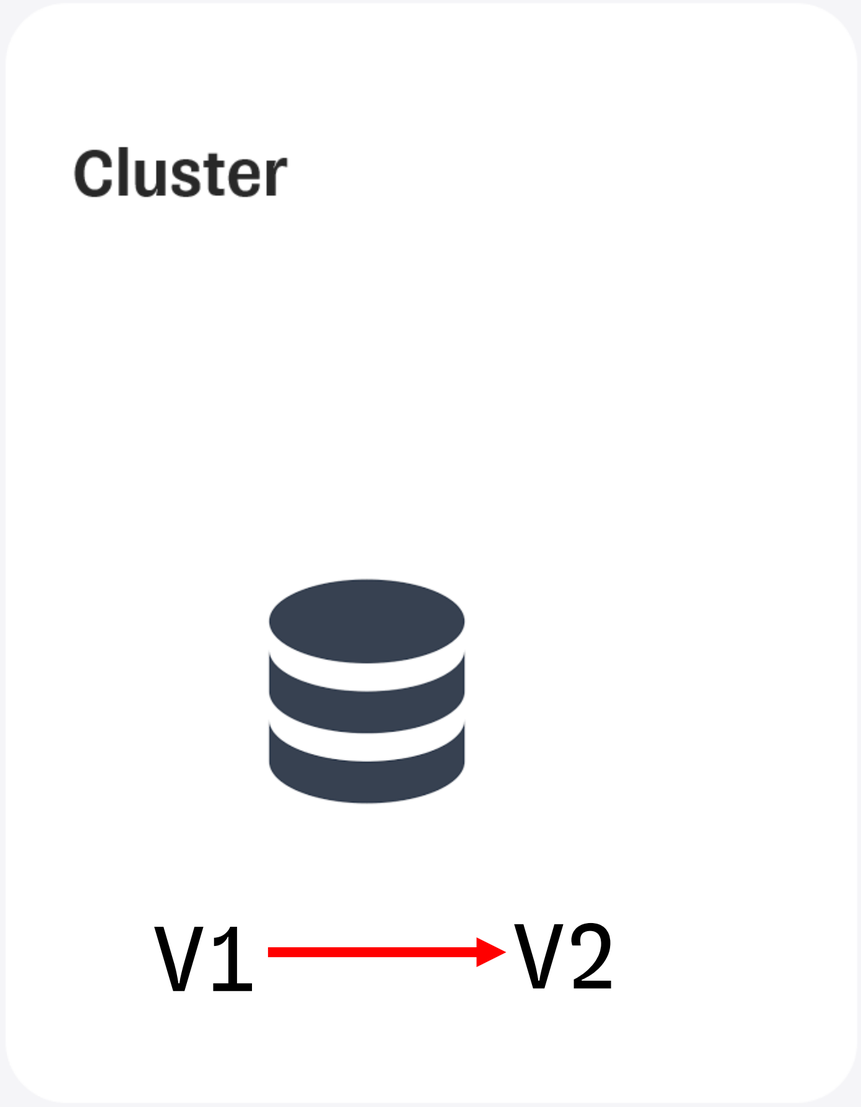
05

Чем NoSql поможет в этом деле

1. Как деплоить код на кластер

Delete & Upload

Delete & Upload



Delete & Upload

Delete & Upload:

Плюсы

Просто

Не требует дополнительных ресурсов в кластере

Одновременно работает только одна версия сервиса

Delete & Upload:

Плюсы

Просто

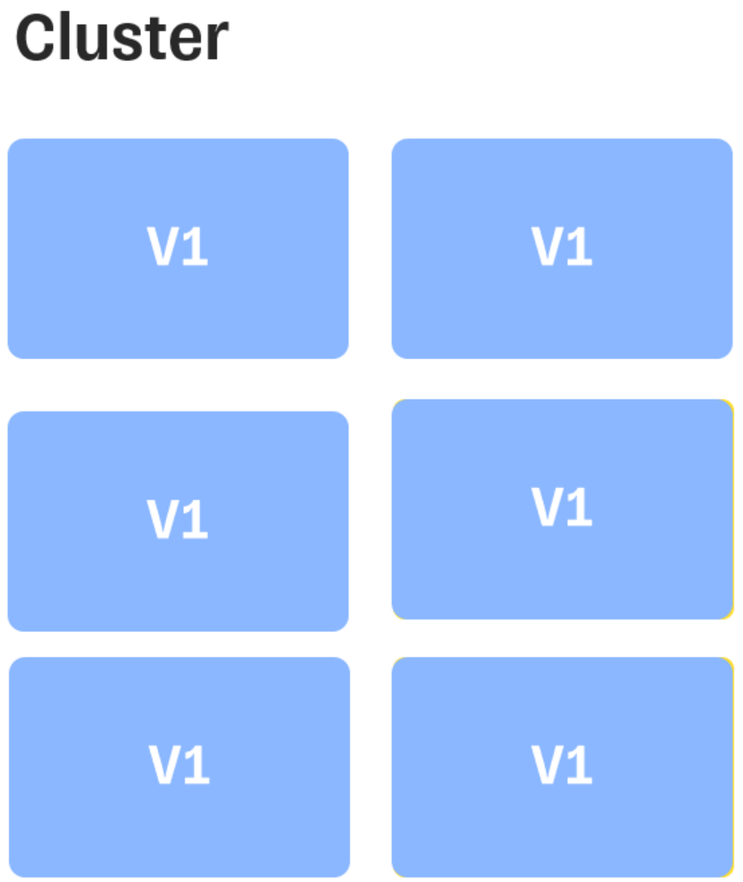
Не требует дополнительных ресурсов в кластере

Одновременно работает только одна версия сервиса

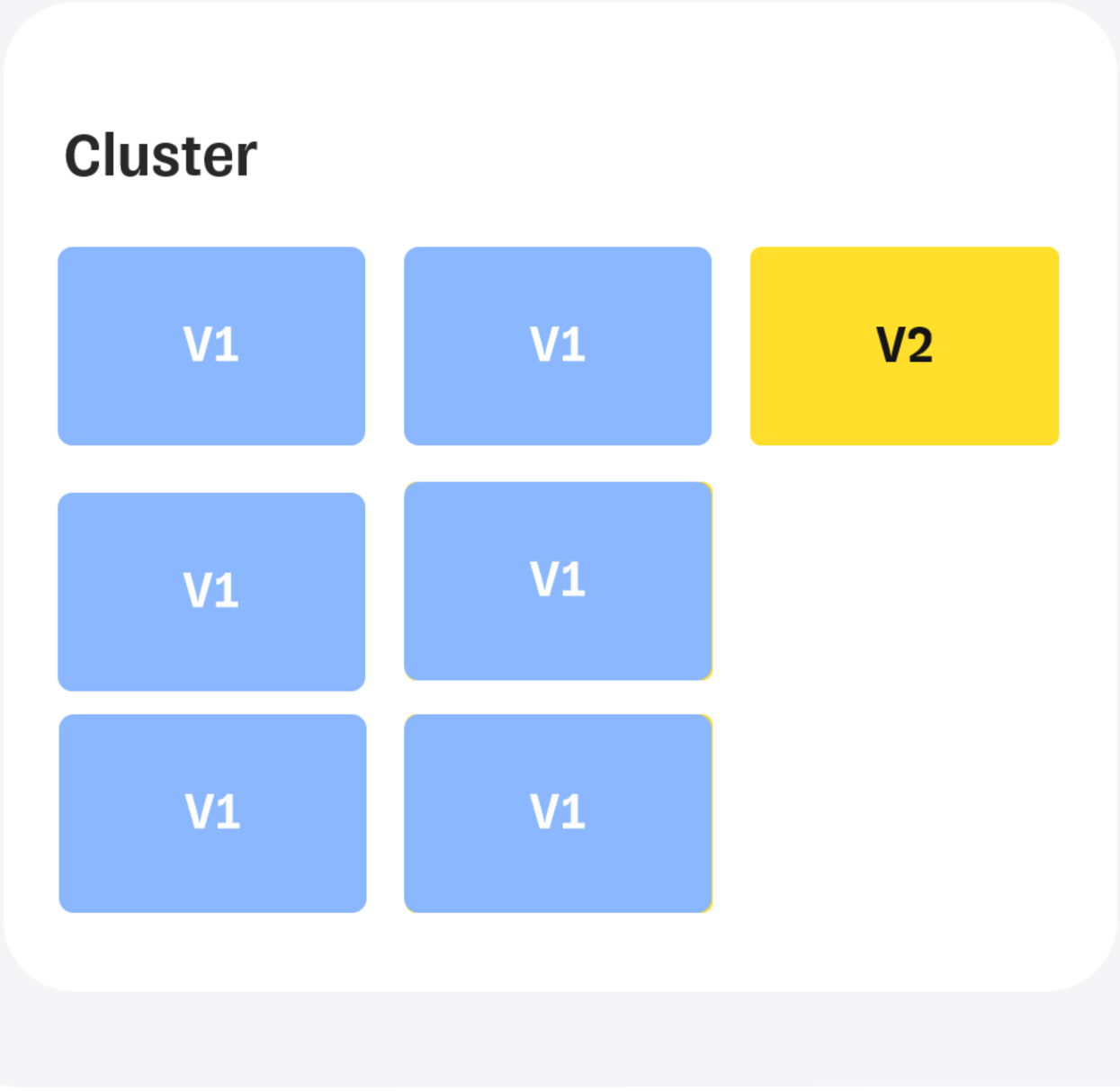
Минусы

Downtime

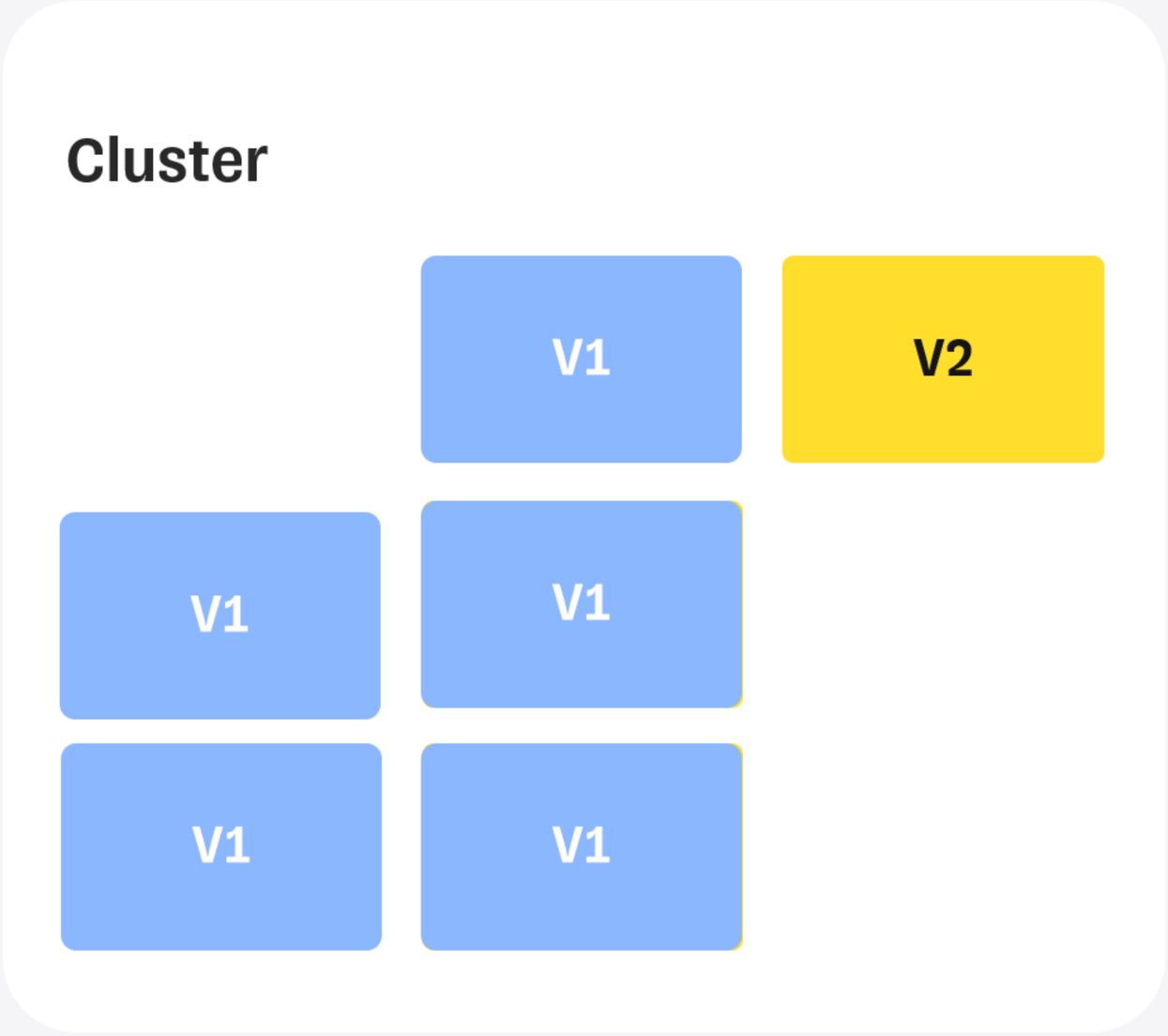
Rolling Update



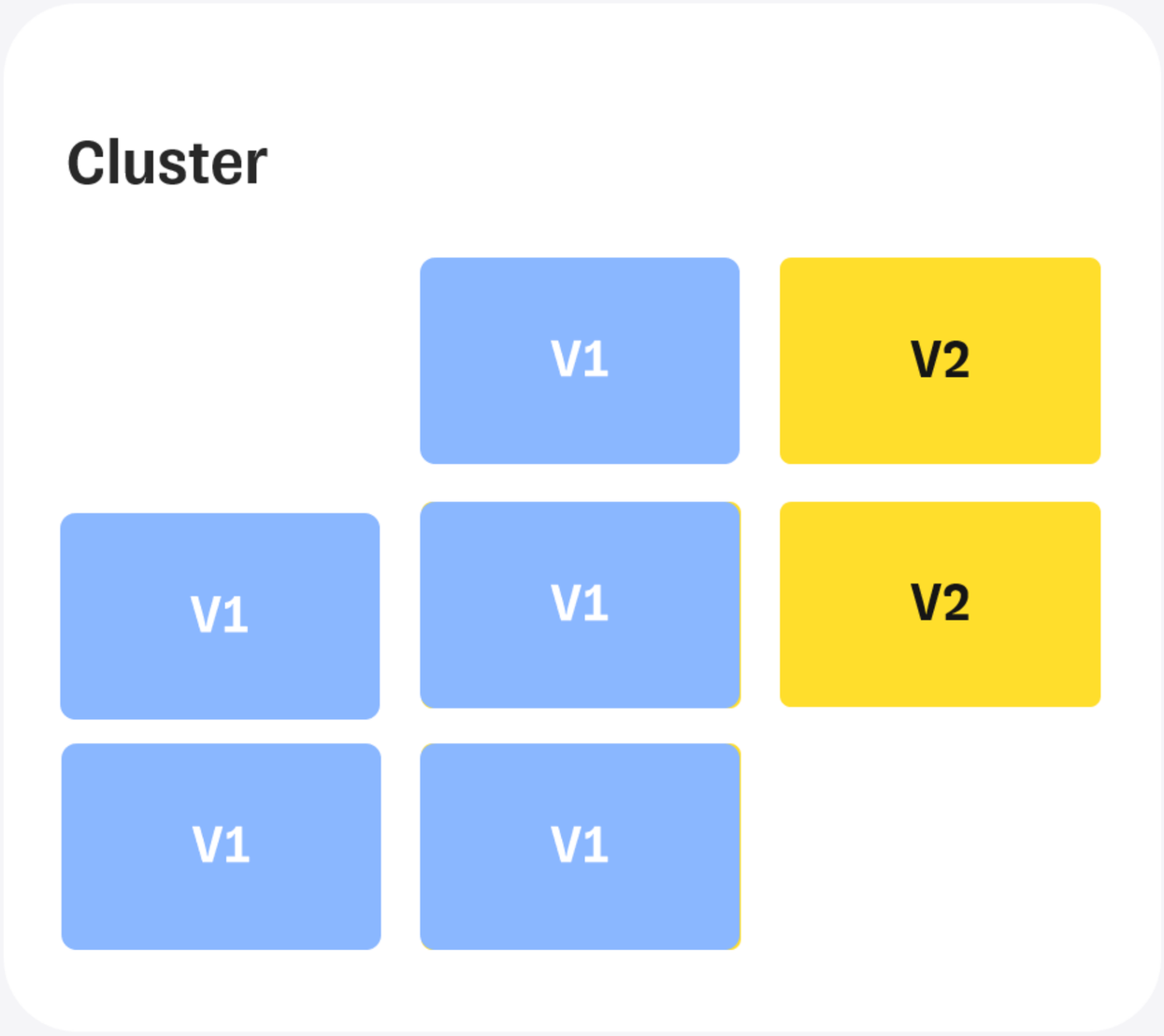
Rolling Update



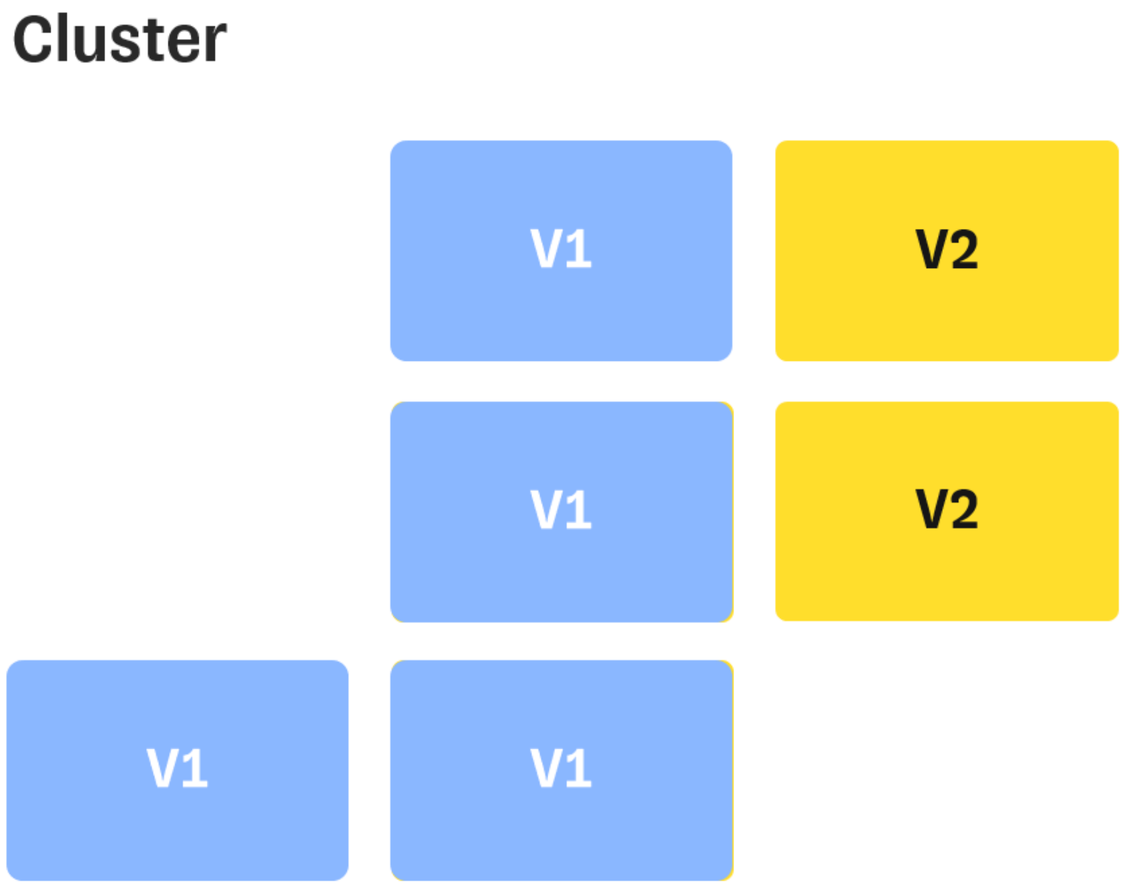
Rolling Update



Rolling Update



Rolling Update

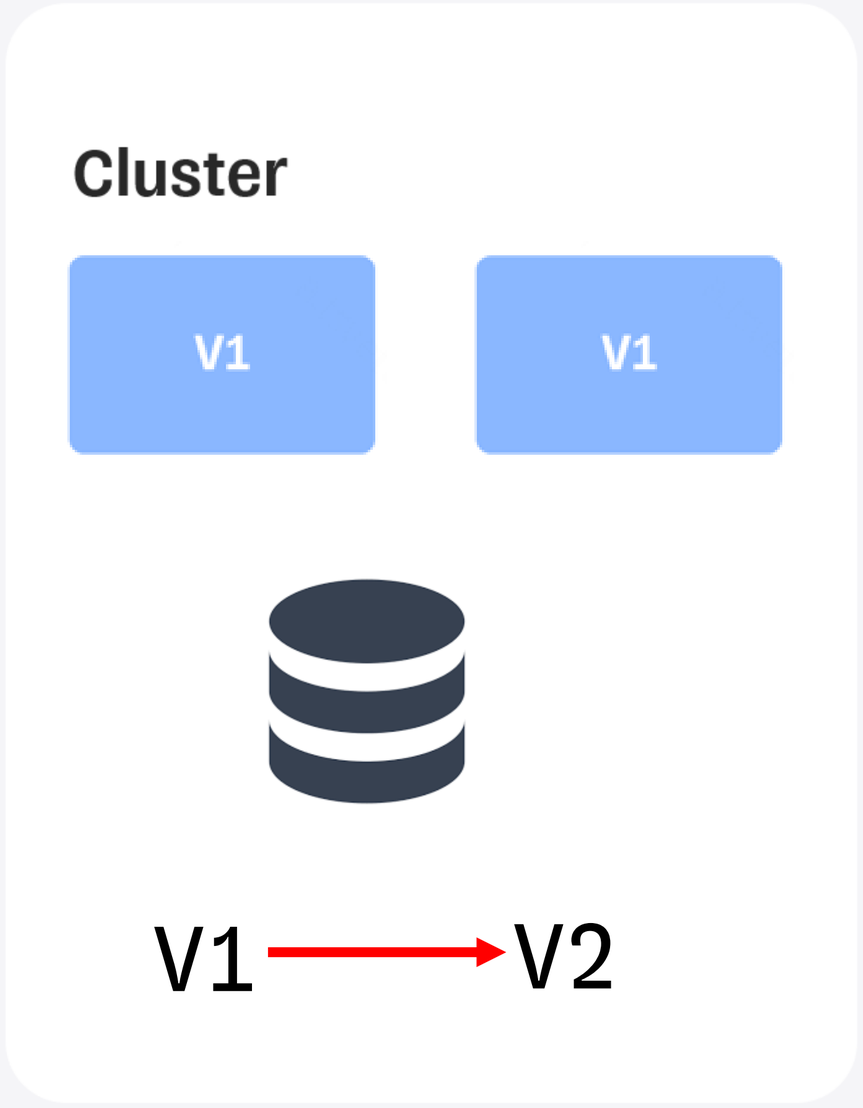


Rolling Update

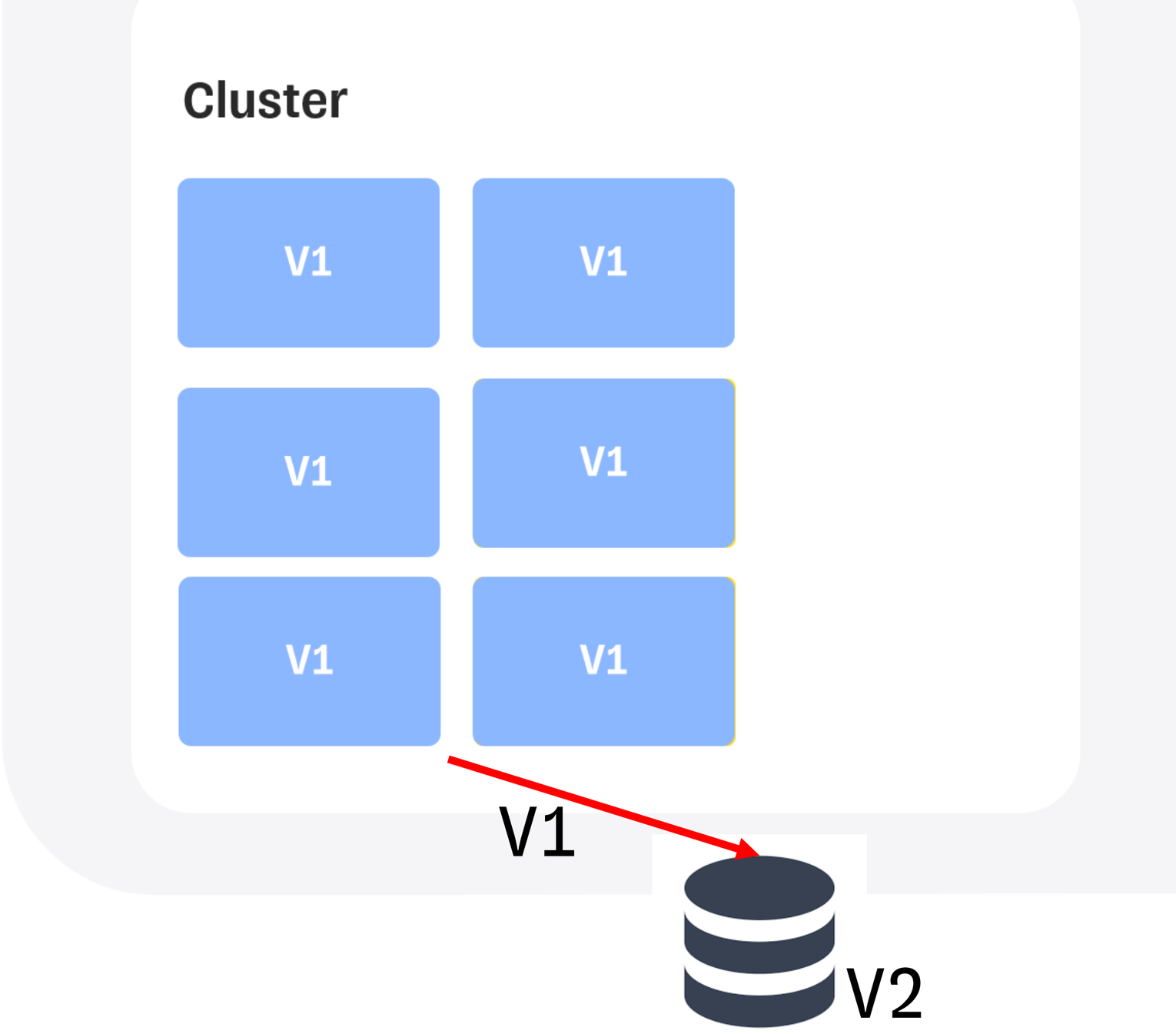
Rolling Update

Rolling Update

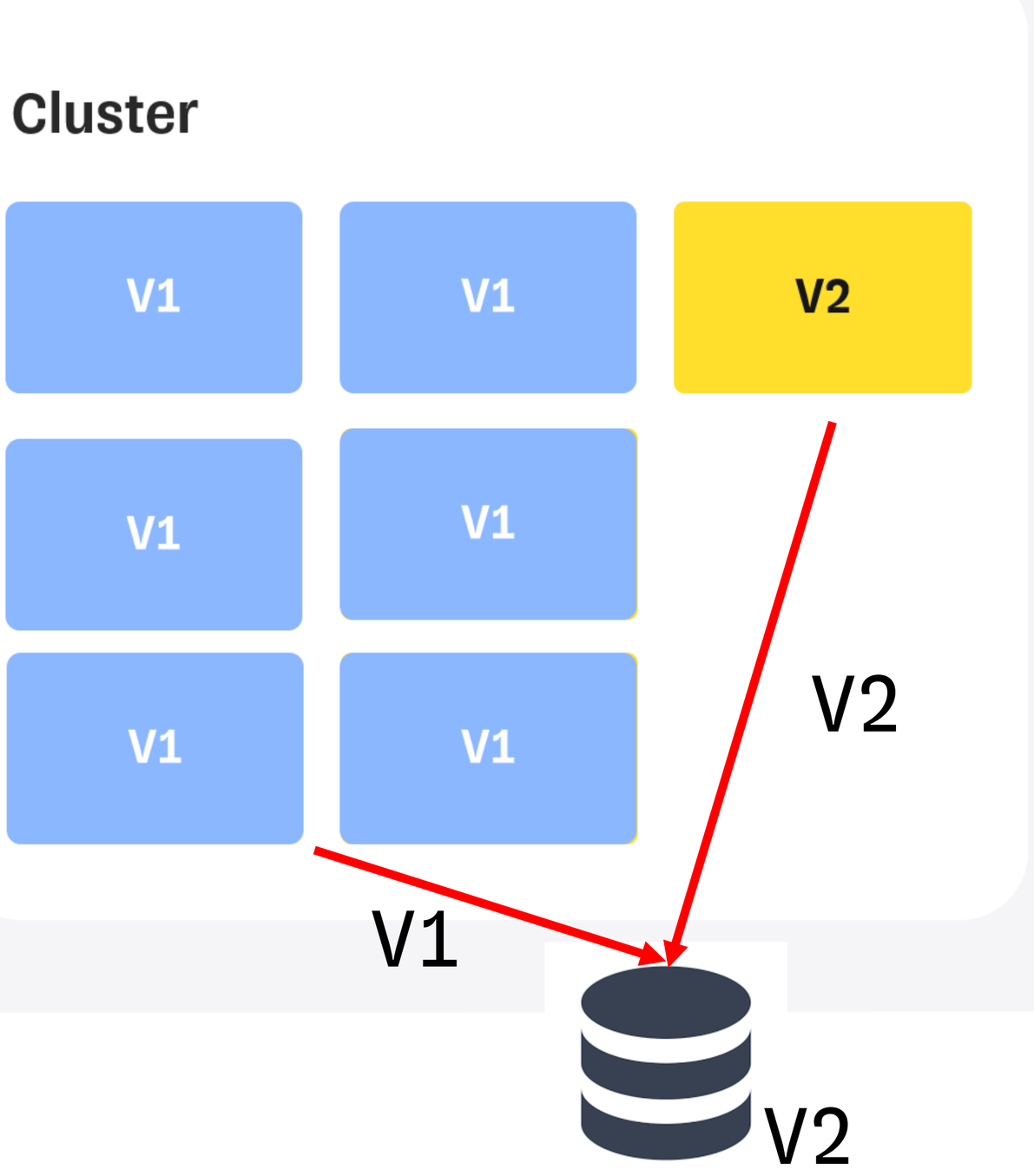
Rolling Update



Rolling Update



Rolling Update



Rolling Update:

Плюсы

Zero-downtime

Требует мало дополнительных ресурсов в кластере

Rolling Update:

Плюсы

Zero-downtime

Требует мало дополнительных ресурсов в кластере

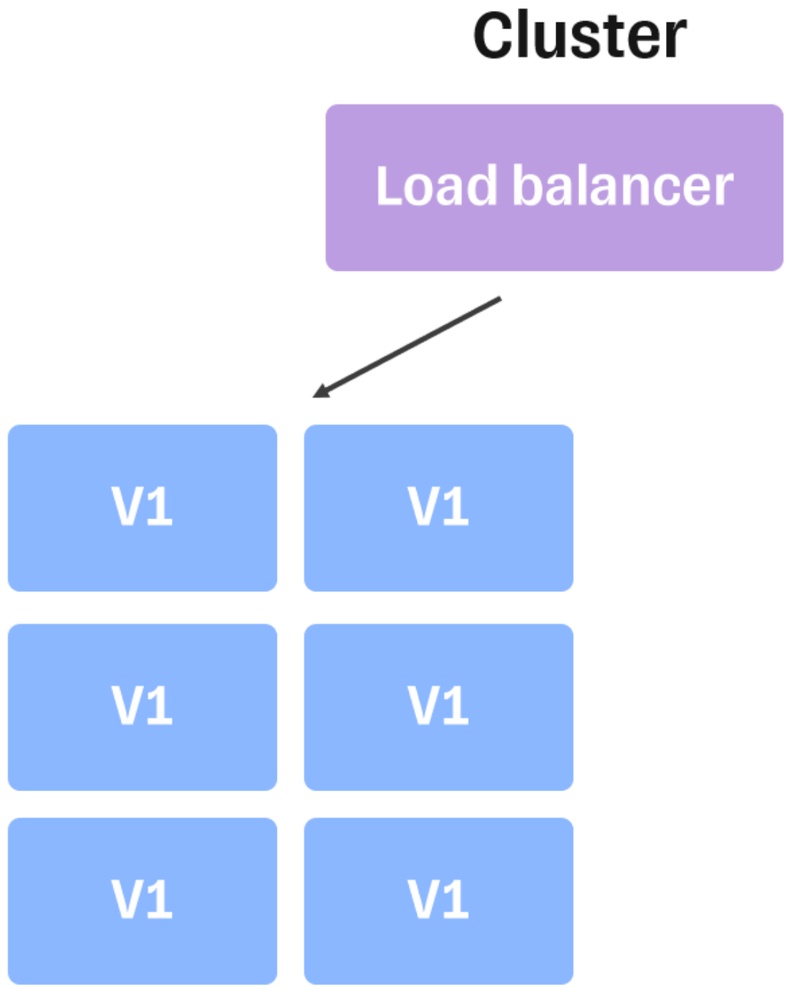
Минусы

Может быть долго

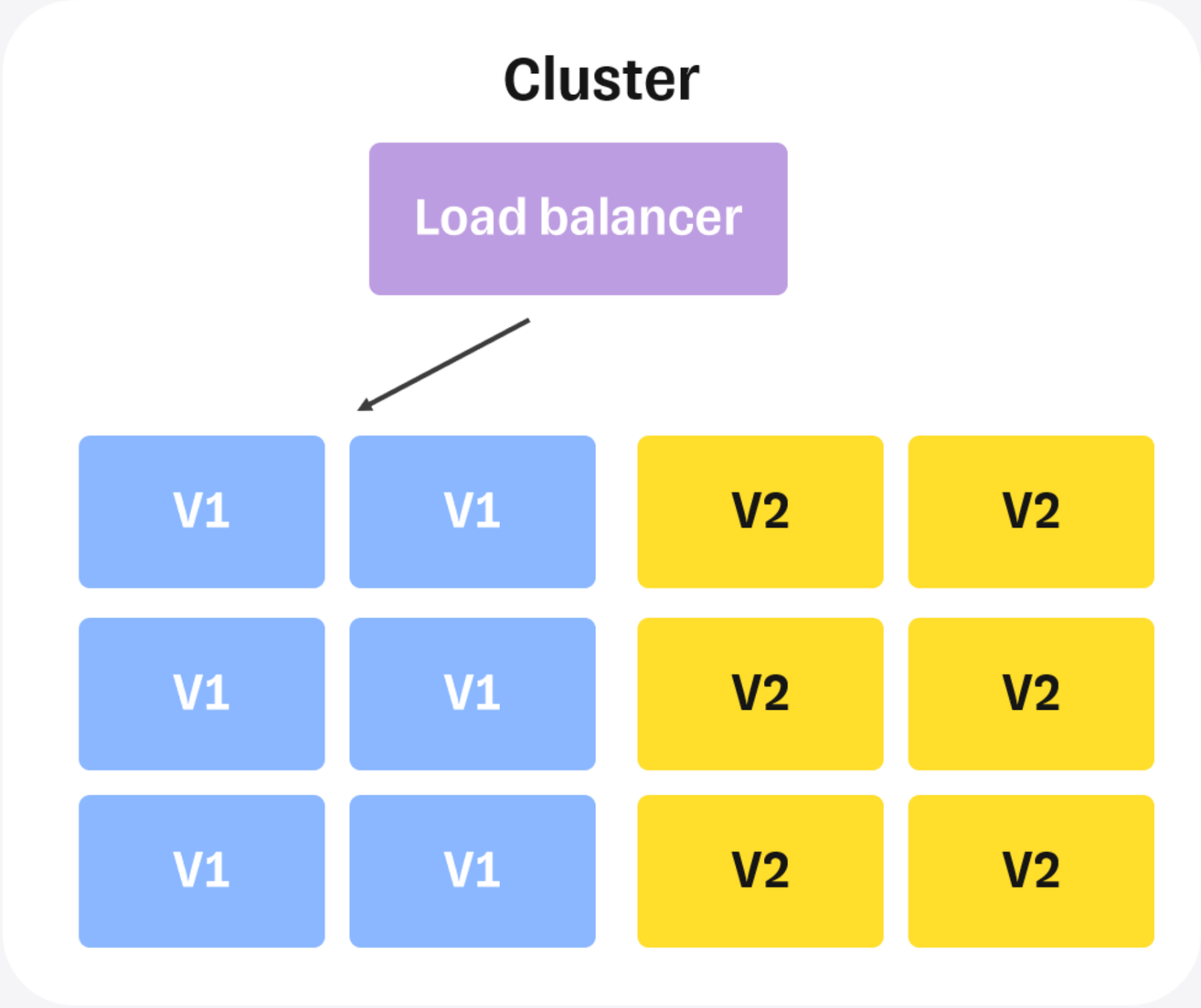
Одновременно работает две версии сервиса

Долгий откат на старую версию

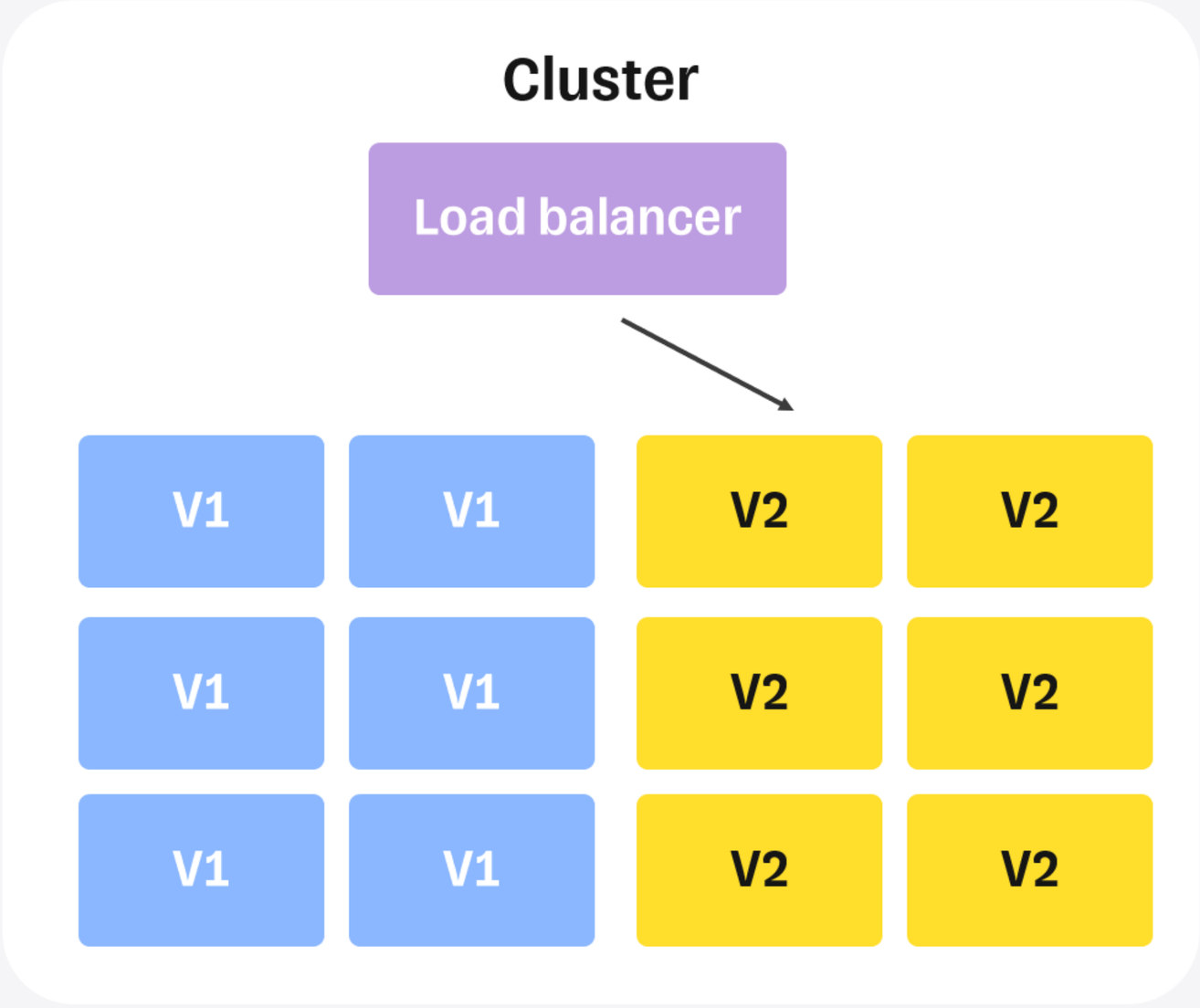
Blue-Green Deployment



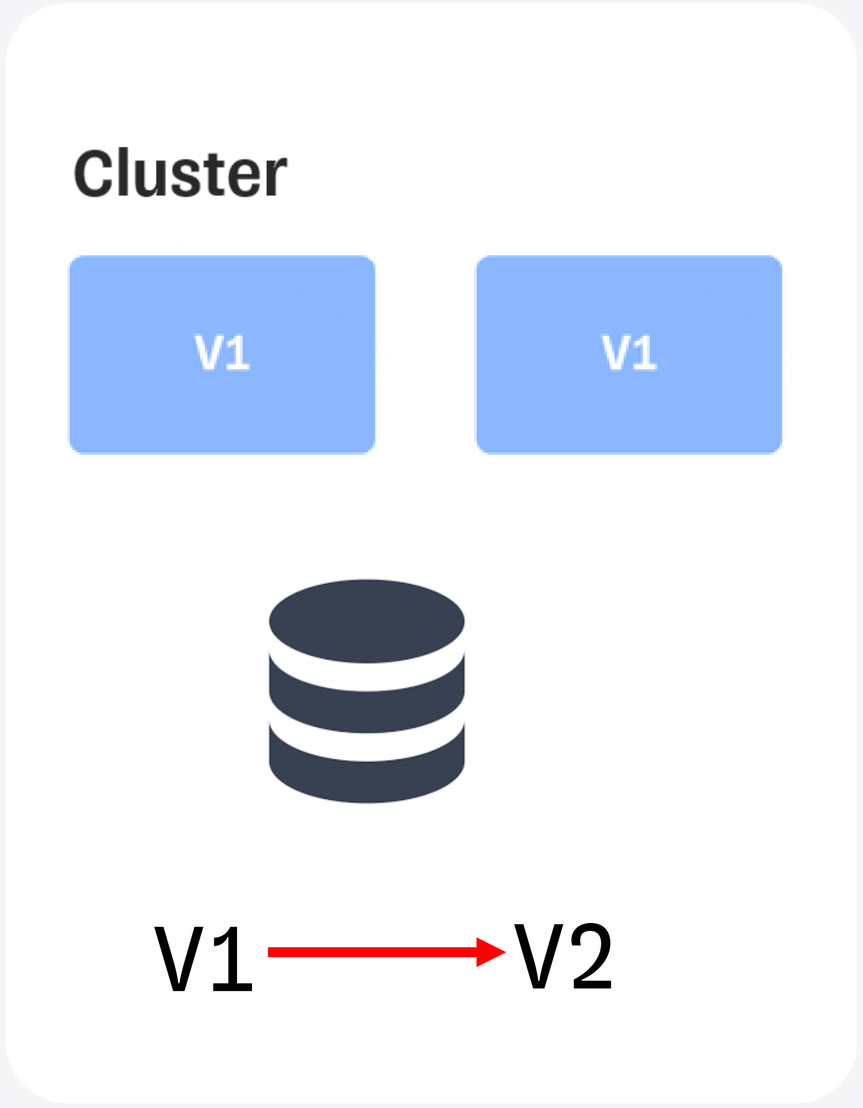
Blue-Green Deployment



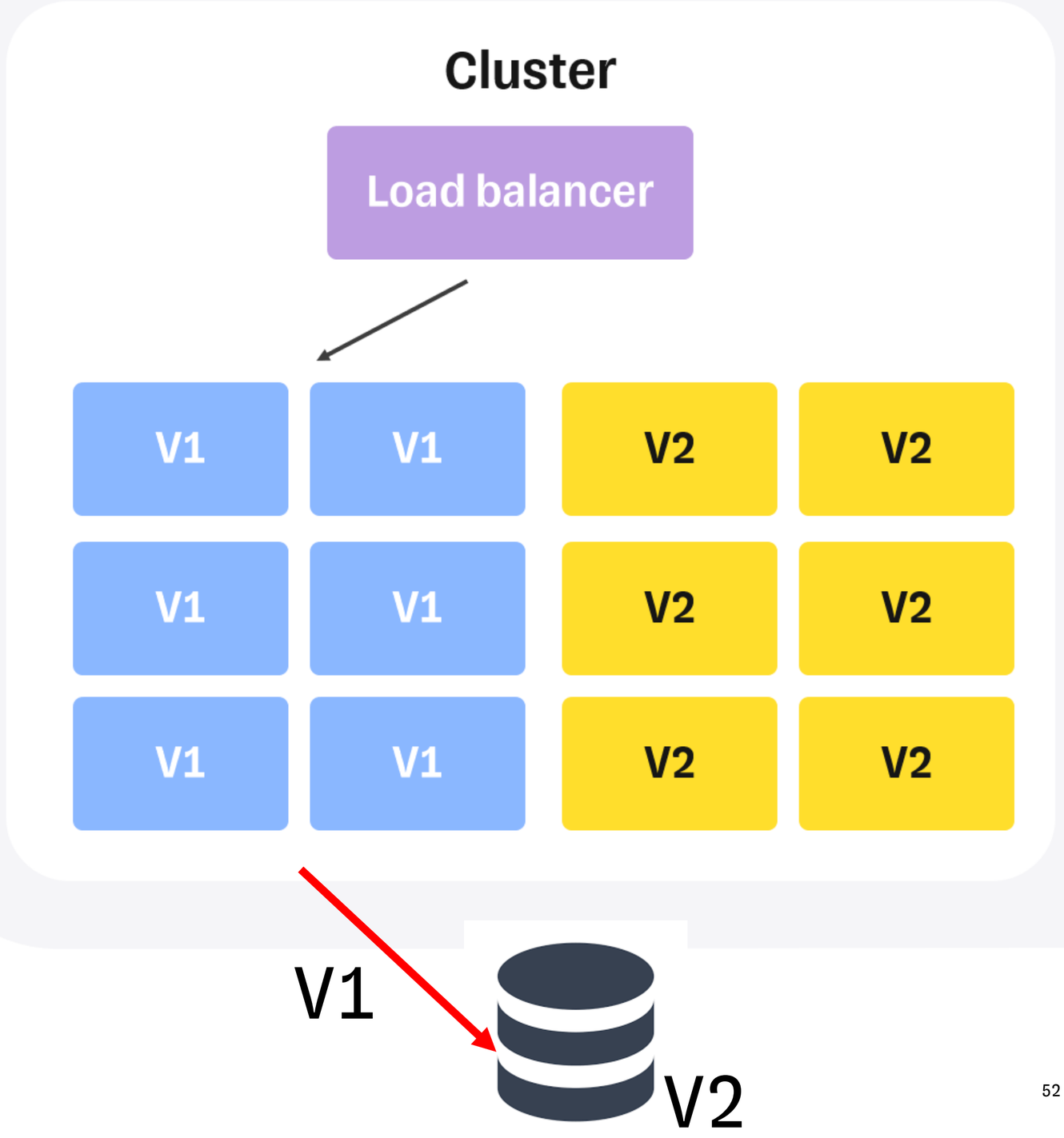
Blue-Green Deployment



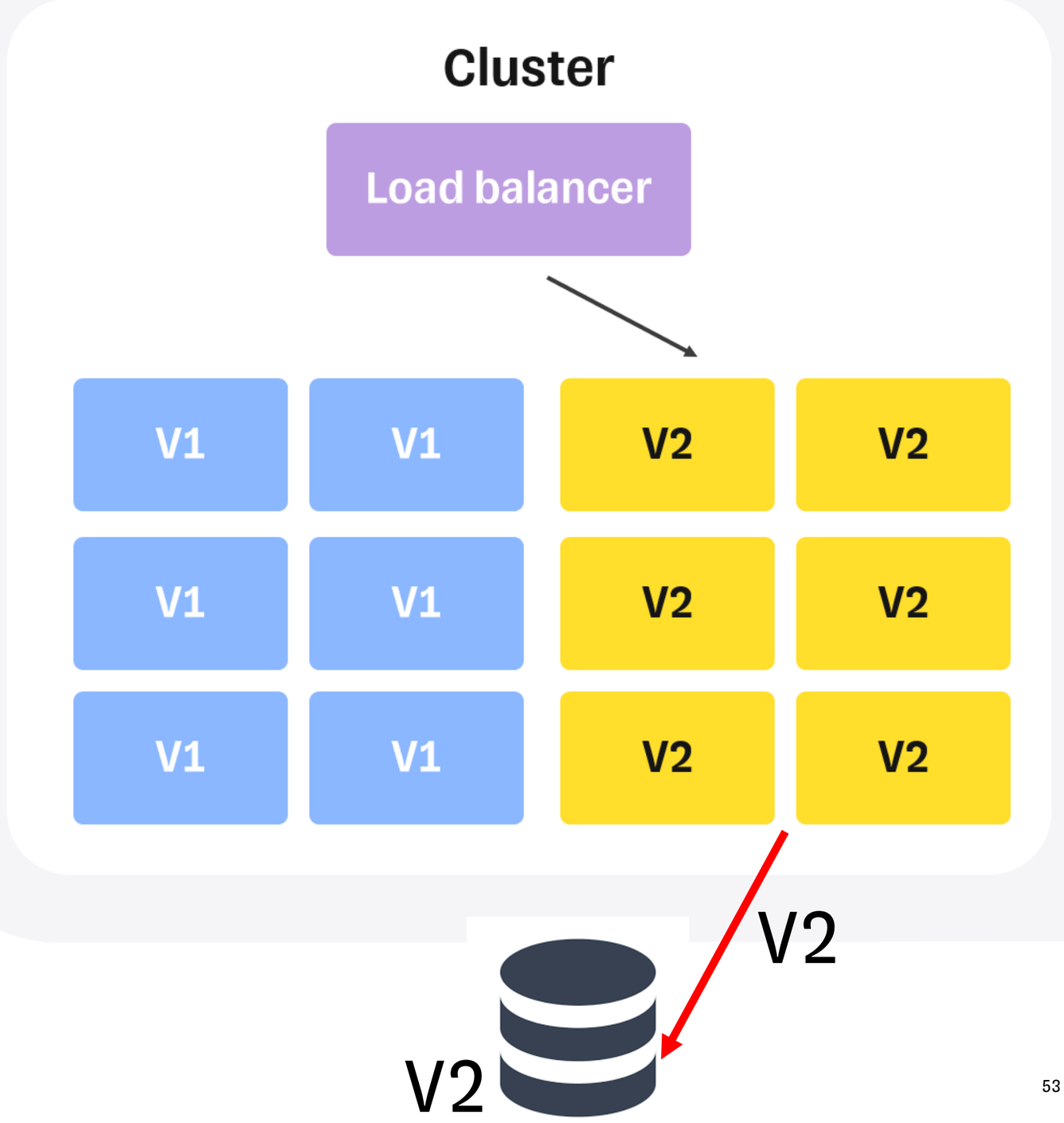
Blue-Green Deployment



Blue-Green Deployment



Blue-Green Deployment



Blue-Green Deployment:

Плюсы

Zero-downtime

Одновременно работает одна версия сервиса

Мгновенное переключение в случае проблем

Blue-Green Deployment:

Плюсы

Zero-downtime

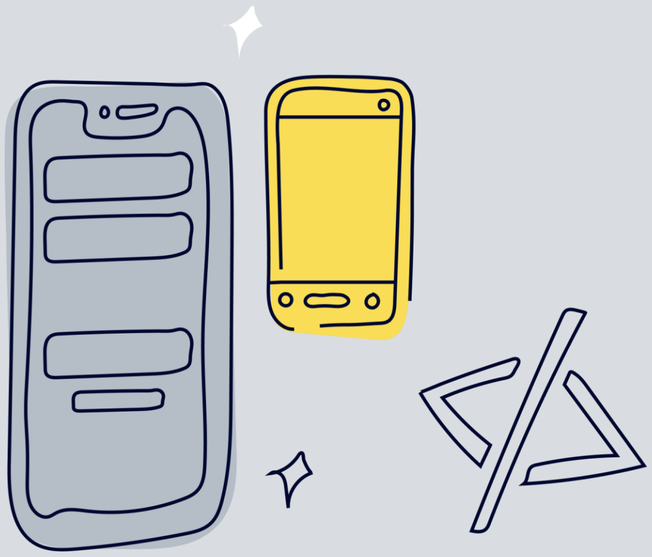
Одновременно работает одна версия сервиса

Мгновенное переключение в случае проблем

Минусы

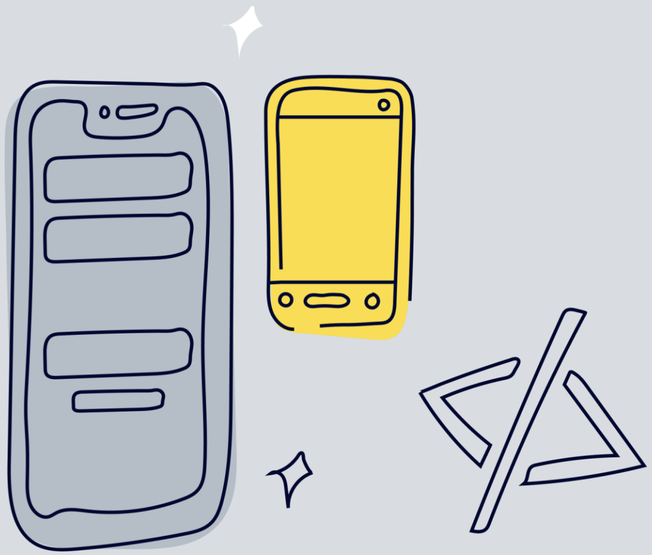
Требует много дополнительных ресурсов в кластере

Итоги по деплоям:



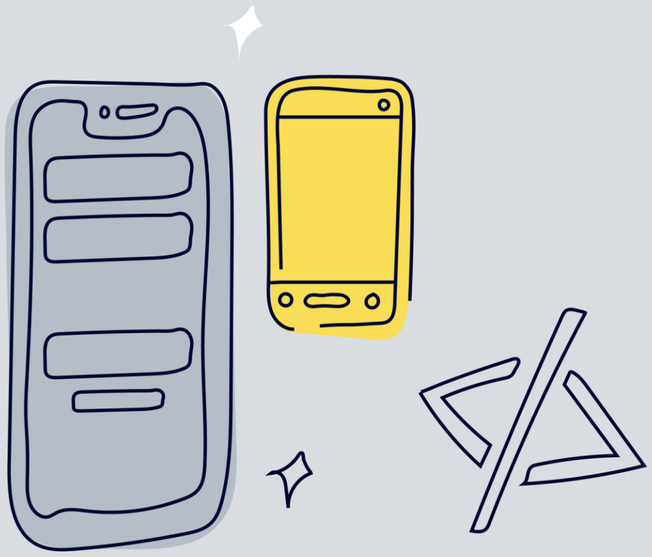
- Используем Rolling update/Blue-Green deployment на проде

Итоги по деплоям:



- Используем Rolling update/Blue-Green deployment на проде
- В обоих случаях старая версия кода работает с новой схемой базы

Итоги по деплоям:



- Используем Rolling update/Blue-Green deployment на проде
- В обоих случаях старая версия кода работает с новой схемой базы
- Поддержка двух версий приложения на уровне базы – цена за zero–downtime deployment

2. Как запускать миграции

Запуск миграции при старте

```
var builder = WebApplication.CreateBuilder(args);

var app = builder.Build();

await app.MigrateDbAsync();
await app.RunAsync();
```

Запуск миграции при старте

```
public static async Task MigrateDbAsync(this WebApplication app)
{
    using var scope = app.Services.CreateScope();

    var dbContext = scope.ServiceProvider.GetRequiredService<DefaultDbContext>();
    await dbContext.Database.MigrateAsync();
}
```

Миграции при старте:

Плюсы

Просто и быстро

Не требует дополнительных усилий при реализации

Миграции при старте:

Плюсы

Просто и быстро

Не требует дополнительных усилий

Минусы

БД при старте сервиса может быть недоступна

Одновременный запуск при старте нескольких инстансов

Инстансы стартуют дольше

Инстансы требуют больше ресурсов при старте

Что делать если не удалось мигрировать?

Запуск миграции в pipeline

```
dotnet-ef migrations bundle --project MyProject.DataAccess

./efbundle --connection "$CONNECTIONSTRINGS__DBCONNECTION"
```

Можно ли откатить миграцию?

В новой версии переименовали колонку:

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

Можно ли откатить миграцию?

В новой версии переименовали колонку:

```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

А если удалили колонку?

```
ALTER TABLE categories DROP COLUMN category_notes;
```

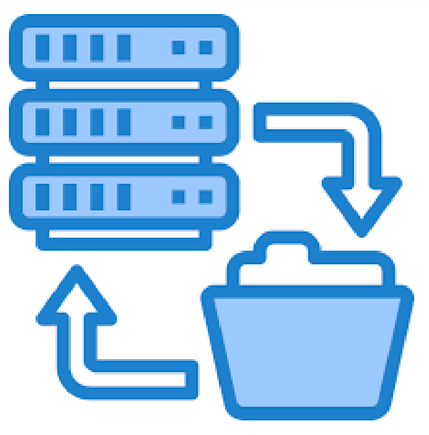
Можно ли откатить миграцию?

В новой версии переименовали колонку:

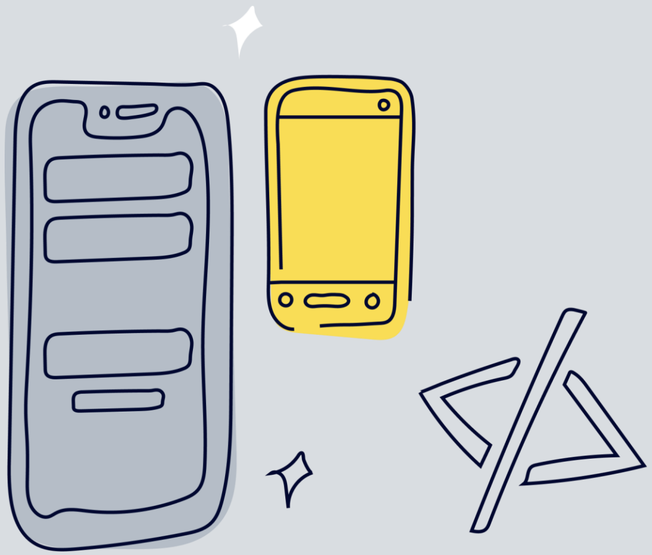
```
ALTER TABLE categories RENAME COLUMN category_notes TO notes;
```

А если удалили колонку?

```
ALTER TABLE categories DROP COLUMN category_notes;
```

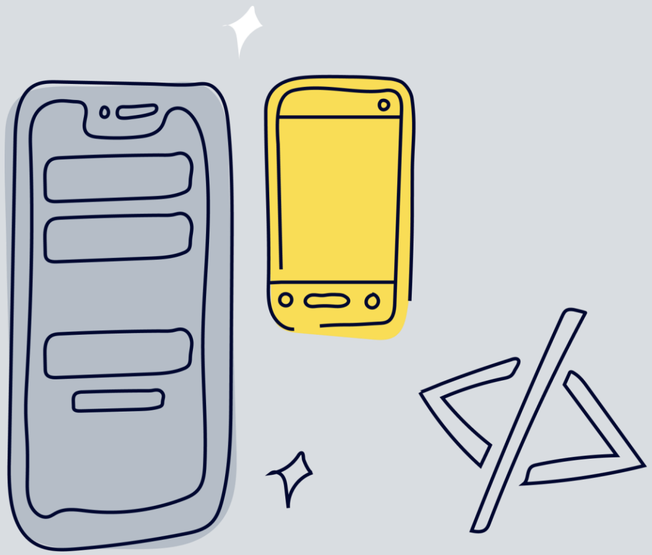


Итоги по запуску миграций:



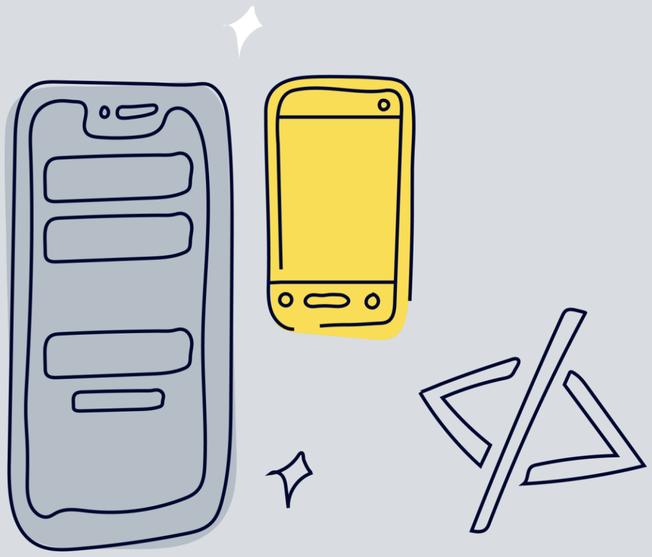
- Миграции в pipeline!

Итоги по запуску миграций:



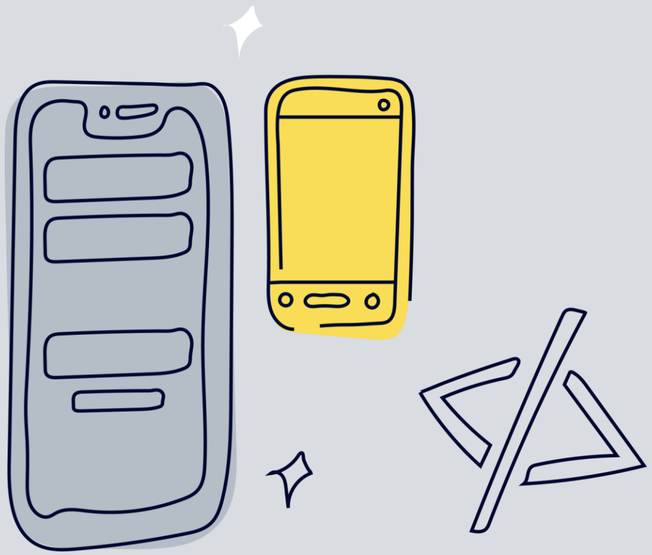
- Миграции в pipeline!
- Запуск из k8s jobs

Итоги по запуску миграций:



- Миграции в pipeline!
- Запуск из k8s jobs
- Мигрируйте на EF Core 6.0!

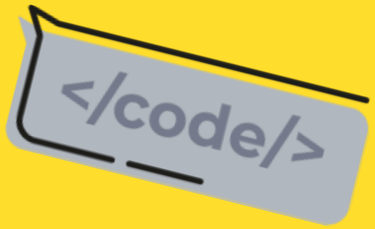
Итоги по запуску миграций:



- Миграции в pipeline!
- Запуск из k8s jobs
- Мигрируйте на EF Core 6.0!
- Если не можете – можно использовать Evolve или отдельный command line проект

3. Как мигрировать данные на новую версию

Обратно-совместимые операции



CREATE TABLE …
ADD COLUMN …

Обратно-совместимые операции

ADD COLUMN WITH DEFAULT

Обратно-совместимые операции

ADD COLUMN WITH DEFAULT

А если в таблице 100500 строк?

Начиная с 11 версии postgres константное значение по умолчанию не проблема

Обратно-совместимые операции

ADD COLUMN WITH DEFAULT

А если в таблице 100500 строк?

Начиная с 11 версии postgres константное значение по умолчанию не проблема

А если нужно вставить сложное значение?

Заполнение новой колонки

- Пишем значение при первом обращении. GET будет писать в базу!

Заполнение новой колонки

Пишем значение при первом обращении. GET будет писать в базу!

Запускаем долгую процедуру заполнения (пишем блоками)

- k8s job
- command line tool
- фоновая задача в сервисе...

Добавление обязательной колонки

Добавить необязательную колонку

Добавление обязательной колонки

Добавить необязательную колонку

Заполнить значением по умолчанию

```
UPDATE categories SET notes = category_notes;
```

Добавление обязательной колонки

Добавить необязательную колонку

Заполнить значением по умолчанию

```
UPDATE categories SET notes = category_notes;
```

Сделать колонку обязательной

Добавление обязательной колонки

Добавить необязательную колонку

Заполнить значением по умолчанию

```
UPDATE categories SET notes = category_notes;
```

Сделать колонку обязательной

Удалить значение по умолчанию

Добавляем обязательную колонку

Добавление колонки и миграция данных в одном релизе

Добавляем обязательную колонку

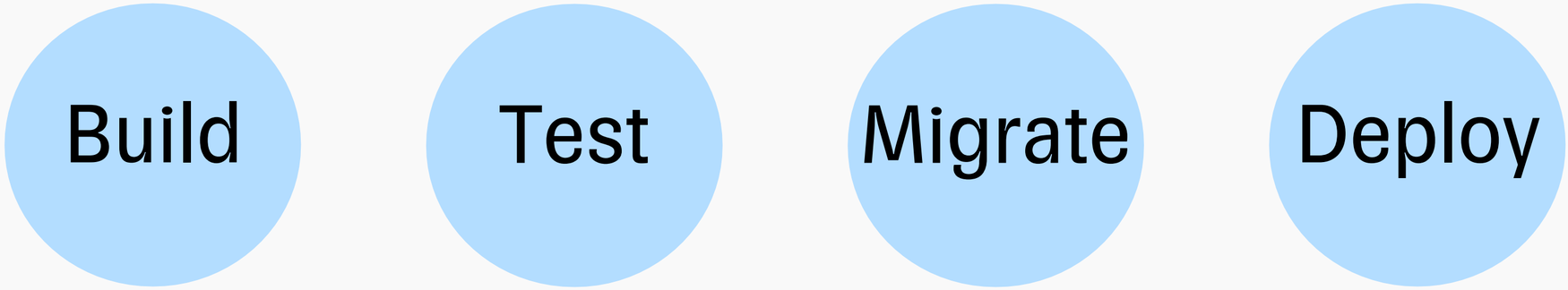
Добавление колонки и миграция данных в одном релизе

Миграции применяются в pipeline до деплоя кода

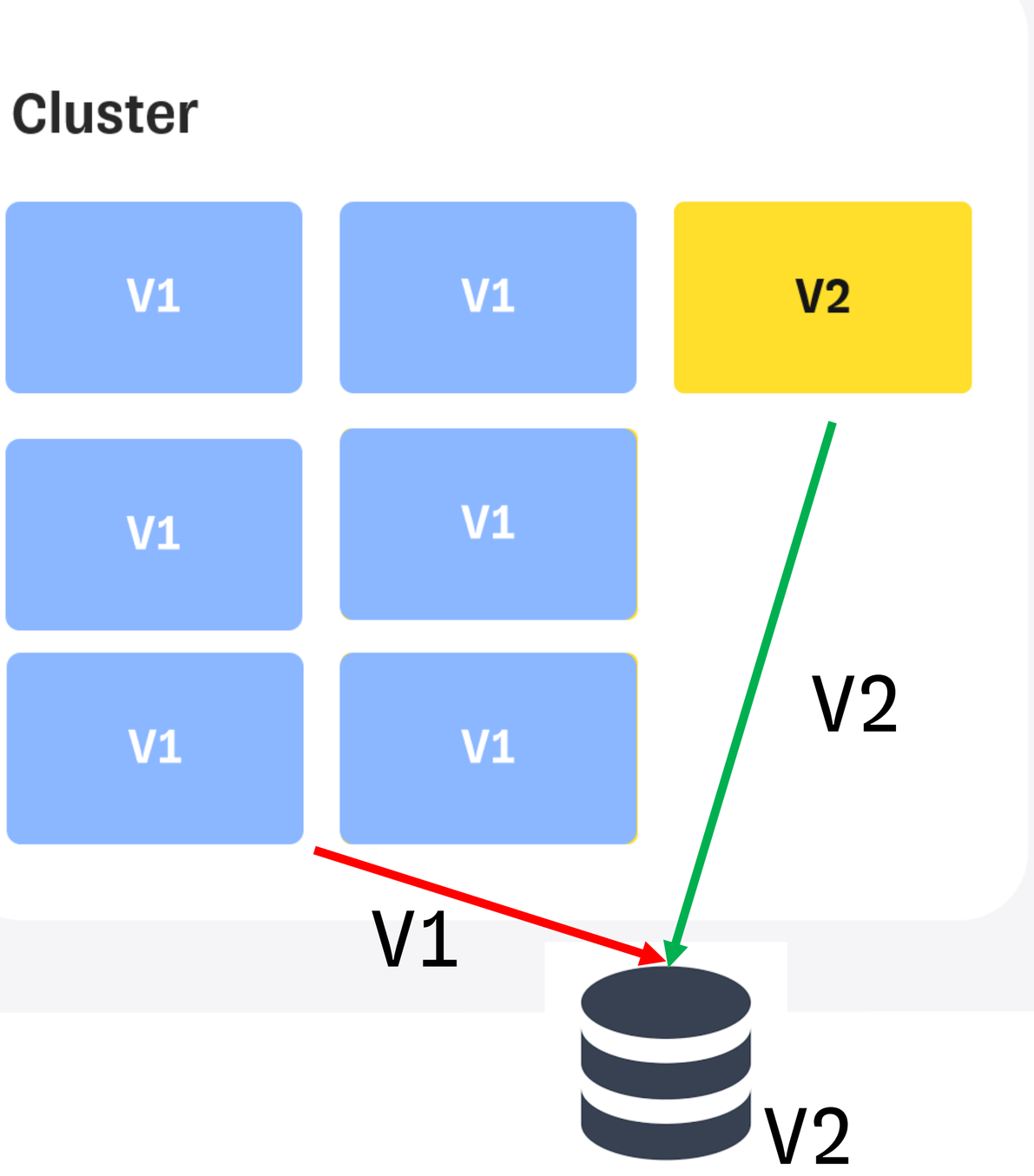
Добавляем обязательную колонку

- Добавление колонки и миграция данных в одном релизе
- Миграции применяются в pipeline до деплоя кода

ADD notes column
UPDATE … SET notes = …



Rolling Update



Как чинить?

На уровне сервиса:

Сделать миграцию в следующем релизе

Повторить миграцию в следующем релизе

Как чинить?

На уровне сервиса:

Сделать миграцию в следующем релизе

Повторить миграцию в следующем релизе

Совместимость на уровне БД:

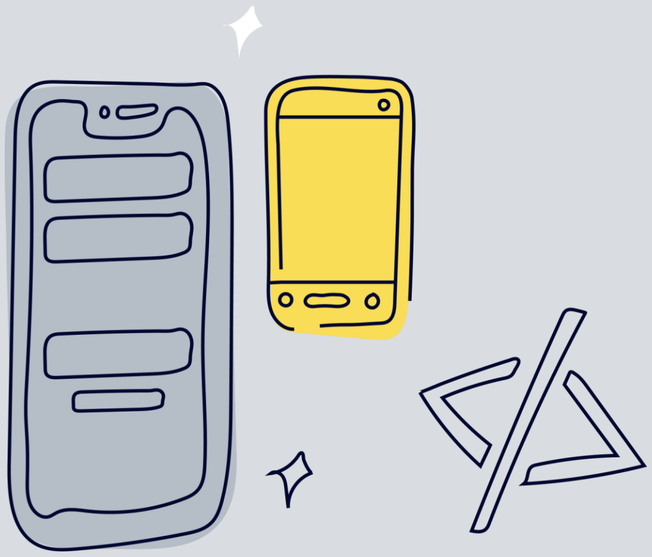
DEFAULT VALUE

- работает для простых случаев

Триггеры

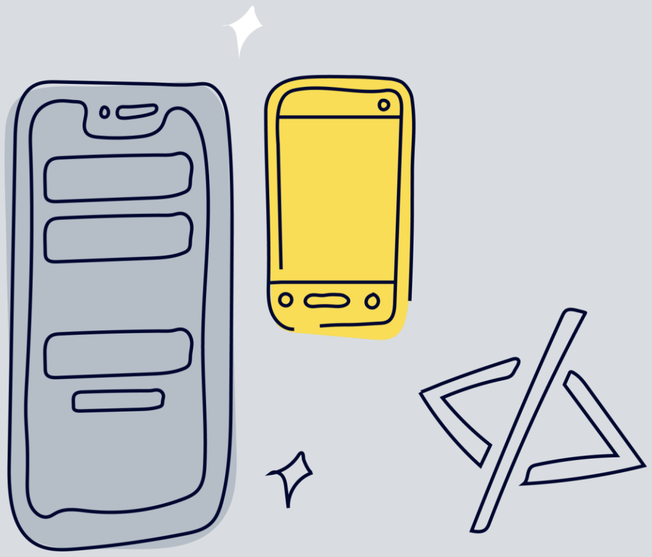
- если новое значение можно посчитать на уровне базы

Итоги по миграции данных:



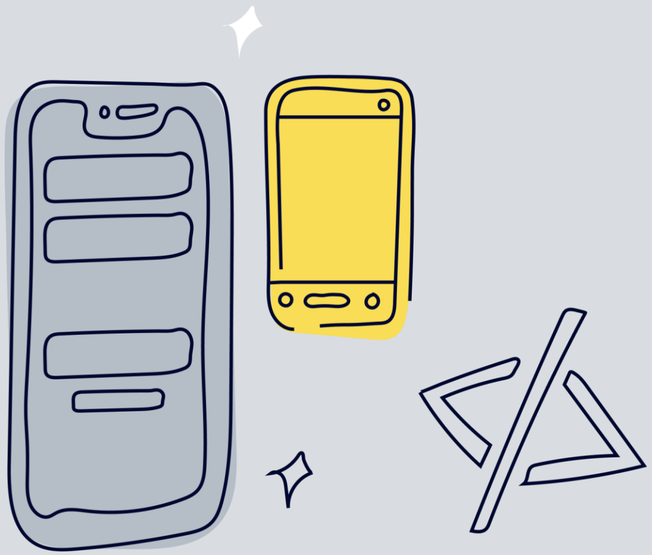
- Если данных много – мигрируем при первом обращении или в фоне

Итоги по миграции данных:



- Если данных много – мигрируем при первом обращении или в фоне
- Нужно помнить что во время деплоя работают 2 версии сервиса

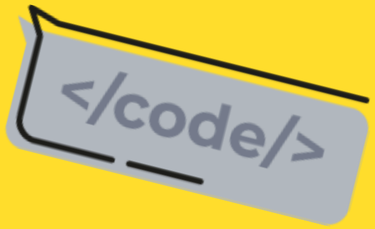
Итоги по миграции данных:



- Если данных много – мигрируем при первом обращении или в фоне
- Нужно помнить что во время деплоя работают 2 версии сервиса
- Убирайте за собой - удаляйте всё ненужное после миграции (код, триггеры и т.д.)

4. Как обеспечить обратную совместимость

Обратно-несовместимые операции



DROP TABLE …
DROP COLUMN …
RENAME TABLE …
RENAME COLUMN …

Удаление колонки

Удаляем использование колонки

Удаление колонки

- Удаляем использование колонки

- Добавляем миграцию

```
protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropColumn(name: "category_notes", table: "categories");
}
```

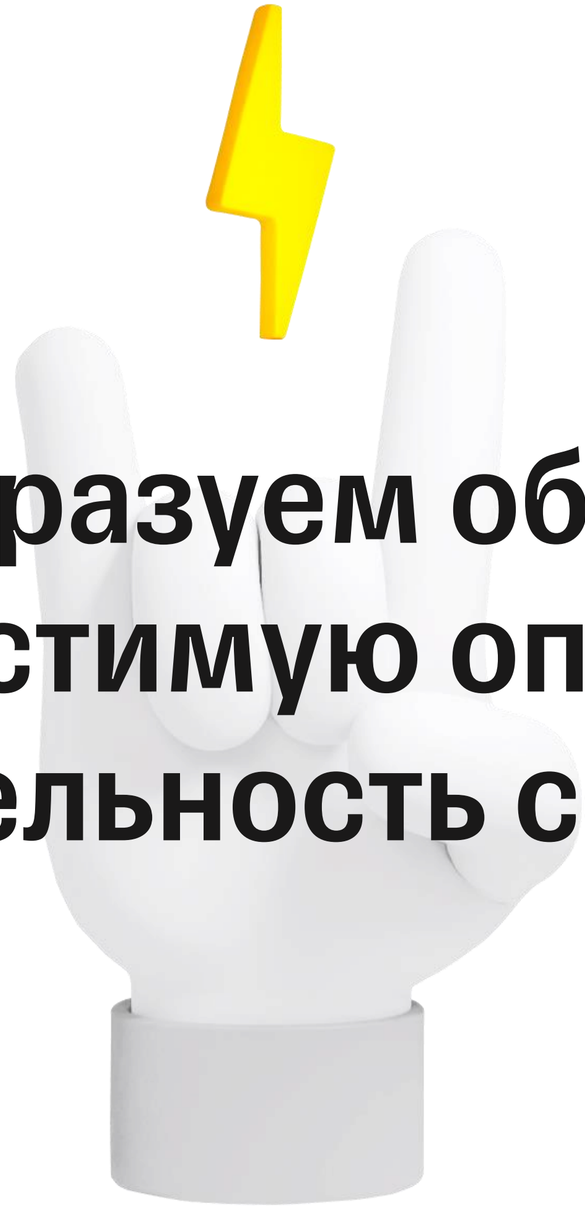
Удаление колонки

Удаляем использование колонки

Добавляем миграцию

```
protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropColumn(name: "category_notes", table: "categories");
}
```

Эта миграция на понадобится в следующей версии!



Преобразуем обратно-несовместимую операцию в последовательность совместимых

Миграция схемы и данных: дано

В таблице categories есть 2 колонки: status и deleted_at

Миграция схемы и данных: дано

В таблице categories есть 2 колонки: status и deleted_at

status – обязательный, enum со значениями: CREATED, APPROVED, ...

Миграция схемы и данных: дано

В таблице categories есть 2 колонки: status и deleted_at

status – обязательный, enum со значениями: CREATED, APPROVED, ...

deleted_at – не обязательный, timestamp для soft delete, дата и время удаления

Миграция схемы и данных: дано

В таблице categories есть 2 колонки: status и deleted_at

status – обязательный, enum со значениями: CREATED, APPROVED, ...

deleted_at – не обязательный, timestamp для soft delete, дата и время удаления

Активные категории – у которых не заполнено deleted_at

Миграция схемы и данных: дано

```
public enum CategoryStatus
{
    CREATED,
    APPROVED
}
public class Category
{
    public DateTimeOffset? DeletedAt { get; set; }
    public CategoryStatus Status { get; set; }

    public bool IsActive() => DeletedAt == null;
    public void Delete() => DeletedAt = DateTimeOffset.UtcNow;
}
```

Миграция схемы и данных: хотим

- Вместо колонки deleted_at использовать status: DELETED

Миграция схемы и данных: хотим

- Вместо колонки deleted_at использовать status: DELETED
- Добавить новый статус

Миграция схемы и данных: хотим

Вместо колонки deleted_at использовать status: DELETED

Добавить новый статус

Мигрировать данные

Миграция схемы и данных: хотим

Вместо колонки deleted_at использовать status: DELETED

Добавить новый статус

Мигрировать данные

Изменить логику: категория активна если статус не DELETED

Миграция схемы и данных: хотим

- Вместо колонки deleted_at использовать status: DELETED
- Добавить новый статус
- Мигрировать данные
- Изменить логику: категория активна если статус не DELETED
- Удалить неиспользуемую колонку

Миграция схемы и данных: хотим

```
public enum CategoryStatus
{
    CREATED,
    APPROVED,
    DELETED
}
public class Category
{
    public CategoryStatus Status { get; set; }

    public bool IsActive() => Status != CategoryStatus.DELETED;
    public void Delete() => Status = CategoryStatus.DELETED;
}
```

Миграция схемы и данных: релиз 1

```
public enum CategoryStatus
{
    CREATED,
    APPROVED,
    DELETED
}
public class Category
{
    public DateTimeOffset? DeletedAt { get; set; }
    public CategoryStatus Status { get; set; }

    public bool IsActive() => DeletedAt == null && Status != CategoryStatus.DELETED;
    public void Delete() => DeletedAt = DateTimeOffset.UtcNow;
}
```

Миграция схемы и данных: релиз 2

```
public enum CategoryStatus
{
    CREATED,
    APPROVED,
    DELETED
}
public class Category
{
    public DateTimeOffset? DeletedAt { get; set; }
    public CategoryStatus Status { get; set; }

    public bool IsActive() => DeletedAt == null && Status != CategoryStatus.DELETED;
    public void Delete() => Status = CategoryStatus.DELETED;
}
```

Миграция схемы и данных: релиз 3

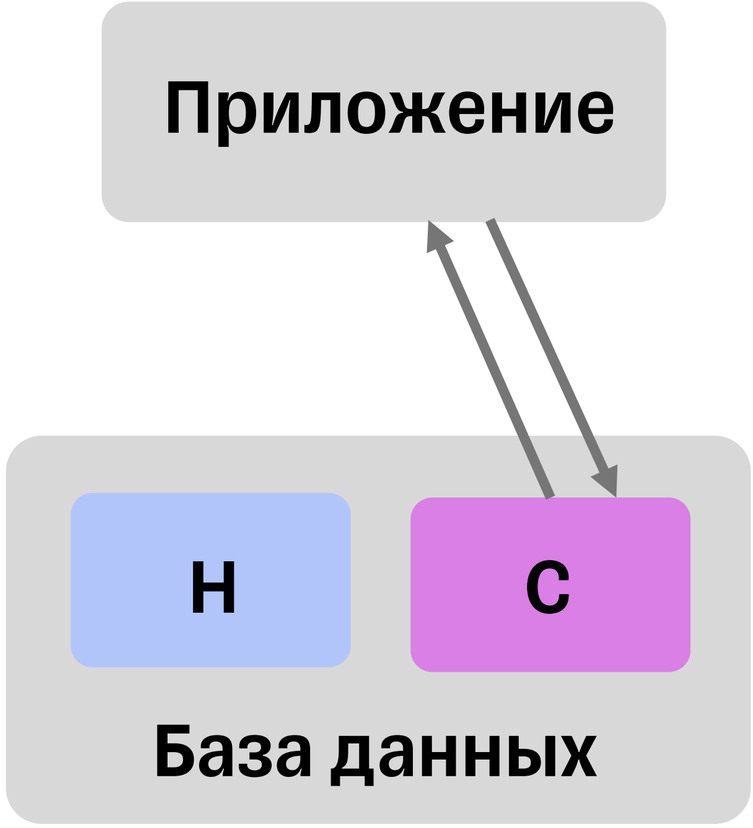
```
UPDATE categories SET status = ‘DELETED' WHERE deleted_at IS NOT NULL;
```

```
public enum CategoryStatus
{
    CREATED,
    APPROVED,
    DELETED
}
public class Category
{
    public CategoryStatus Status { get; set; }

    public bool IsActive() => Status != CategoryStatus.DELETED;
    public void Delete() => Status = CategoryStatus.DELETED;
}
```

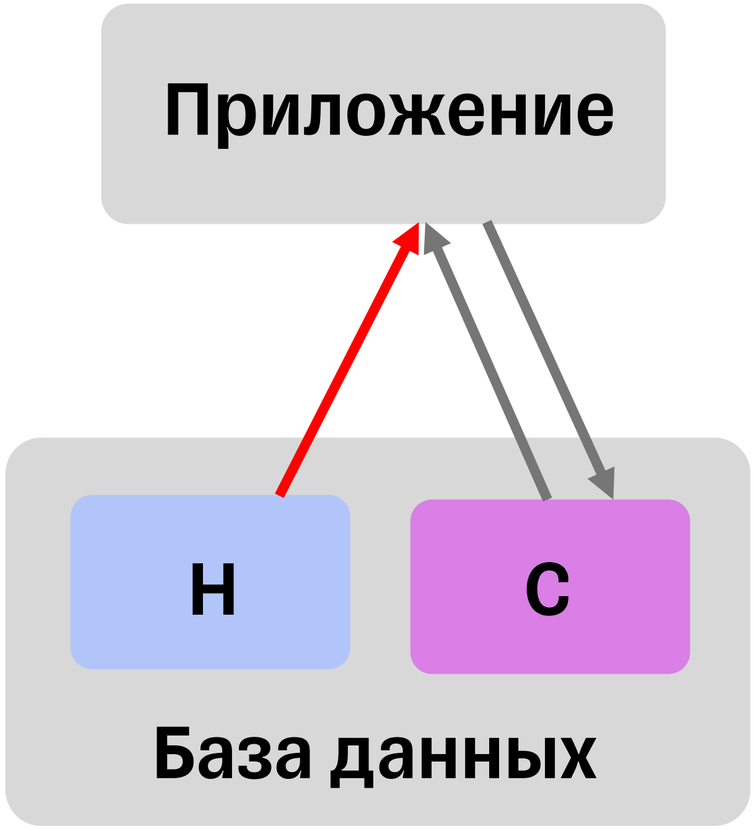
Миграция схемы и данных: релиз 4

```
ALTER TABLE categories DROP COLUMN deleted_at;
```

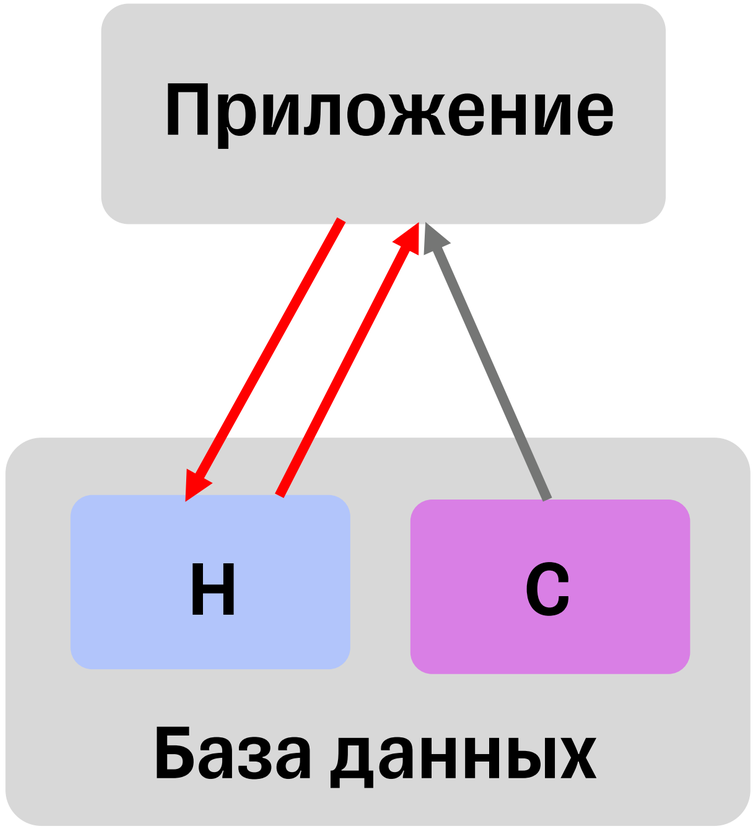
Старая версия



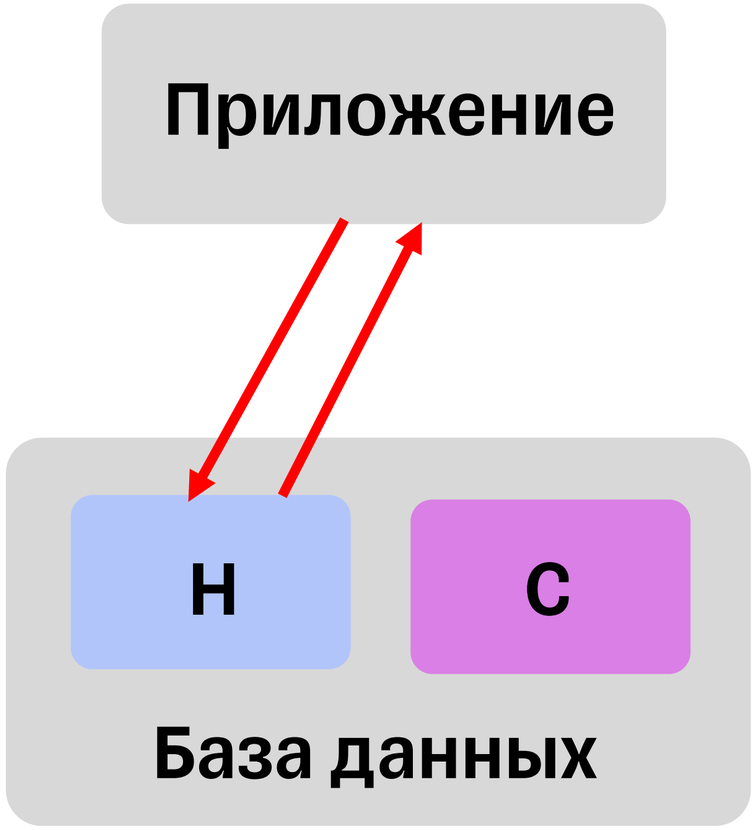
Релиз 1



Релиз 2



Релиз 3



Переименование таблицы - 1

Создать новую таблицу

Переименование таблицы - 1

Создать новую таблицу

Мигрировать старые данные

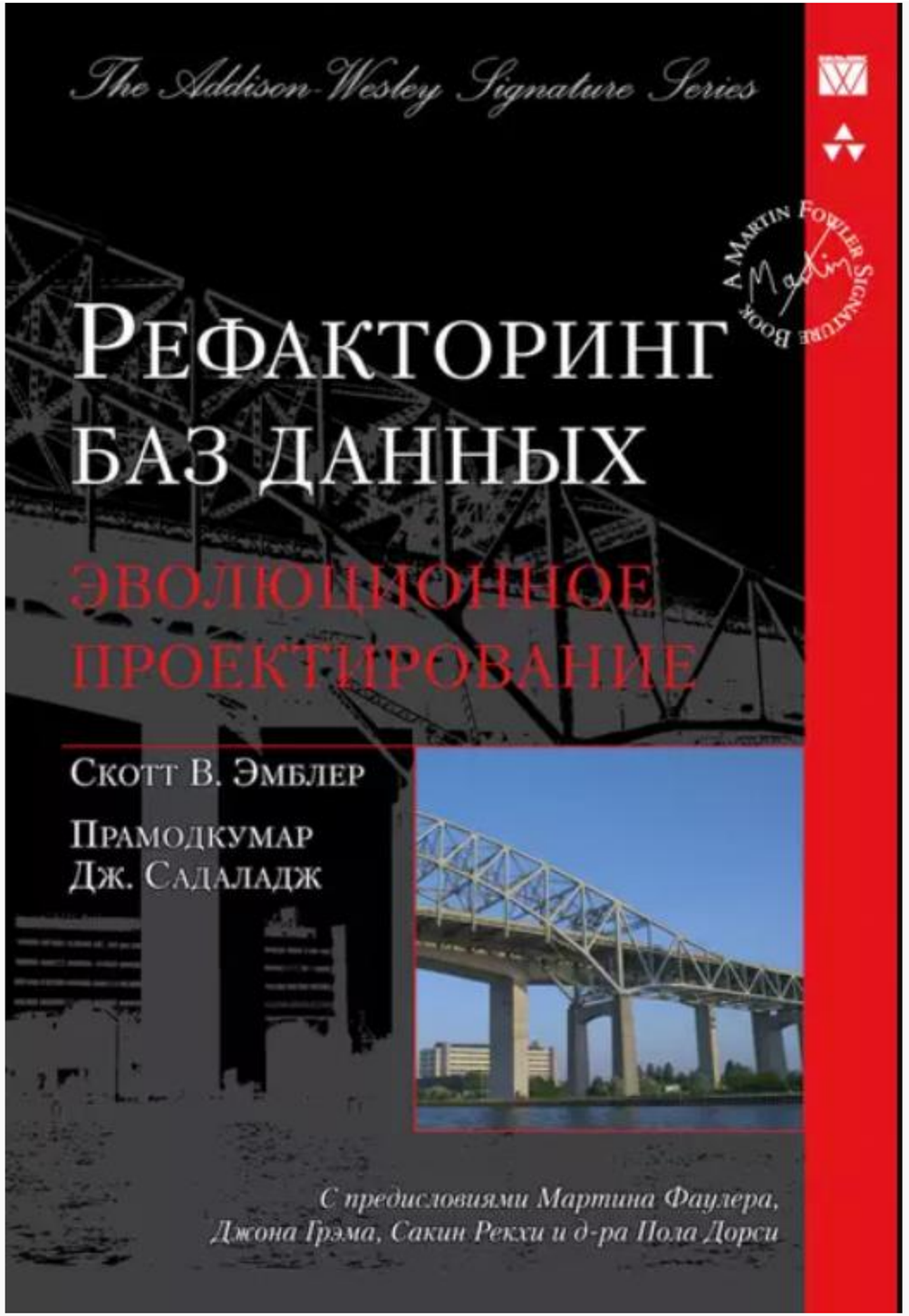
Переименование таблицы - 1

Создать новую таблицу

Мигрировать старые данные

Синхронизировать новые данные между таблицами (здравствуй триггеры)


И ТАК СОЙДЕТ!


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Рефакторинг баз данных
эволюционное проектирование
Скотт В. Эмблер
Прамодкумар Дж. Садаладж
С предисловиями Мартина Фаулера, Джона Грэма, Сакин Рекхи и д-ра Пола Дорси

Переименование таблицы - 2

Переименовать таблицу

Переименование таблицы - 2

Переименовать таблицу

Создаем обновляемое представление (view)

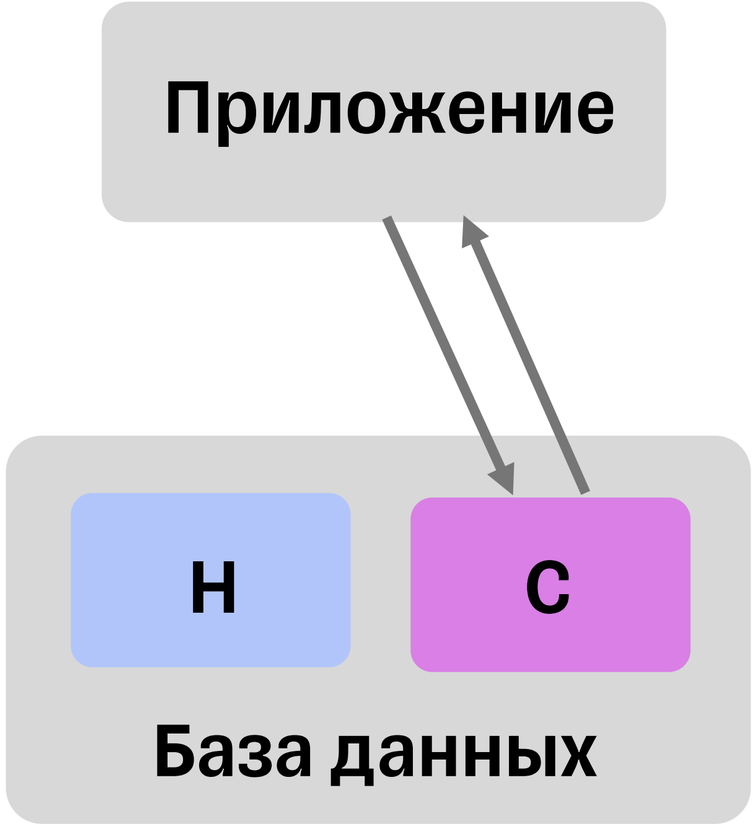
Переименование таблицы - 2

Переименовать таблицу

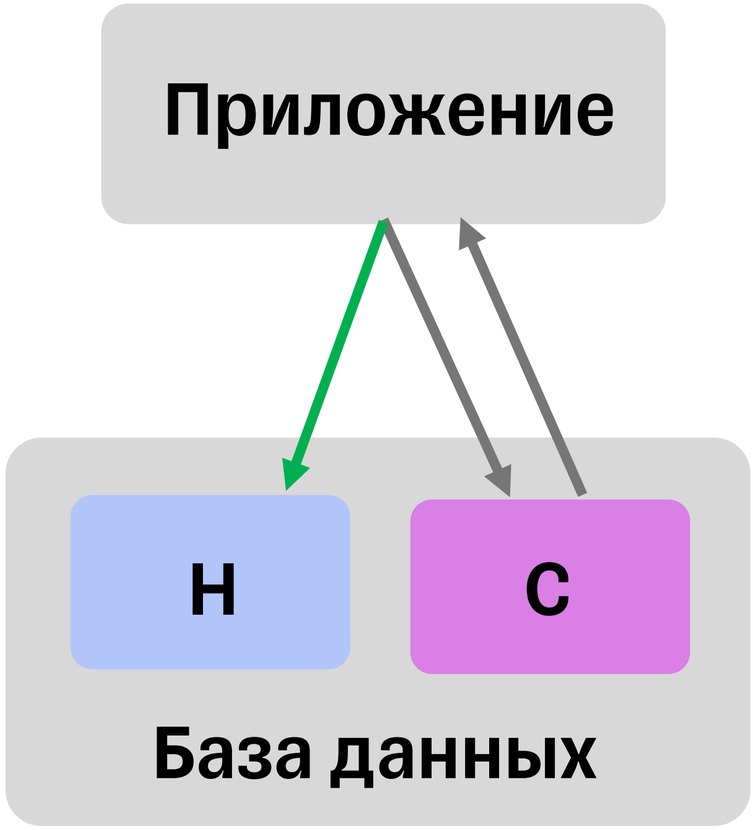
Создаем обновляемое представление (view)

Не нужно мигрировать данные и синхронизировать их на время деплоя и работы!

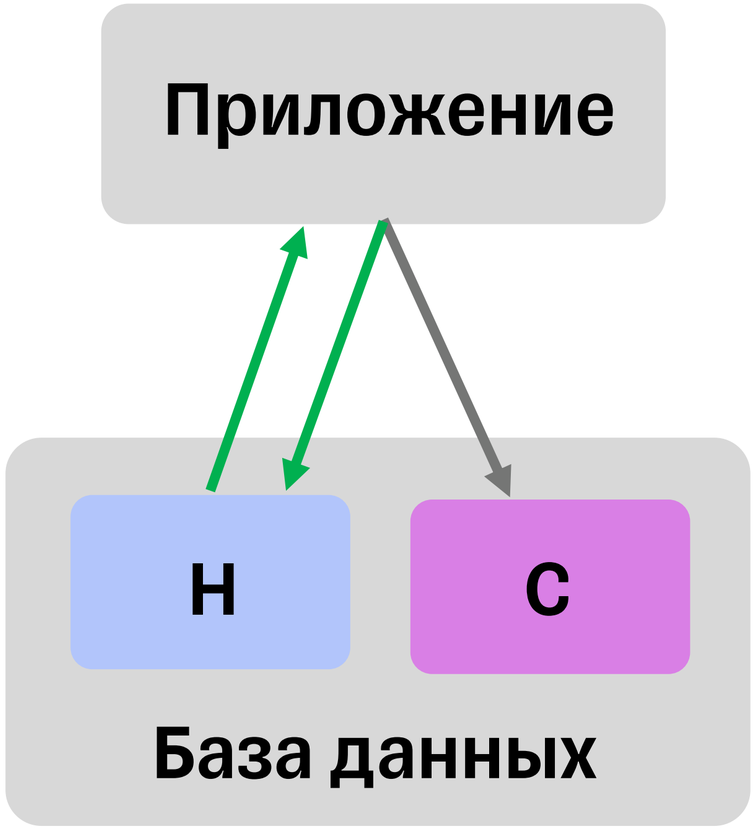
Старая версия



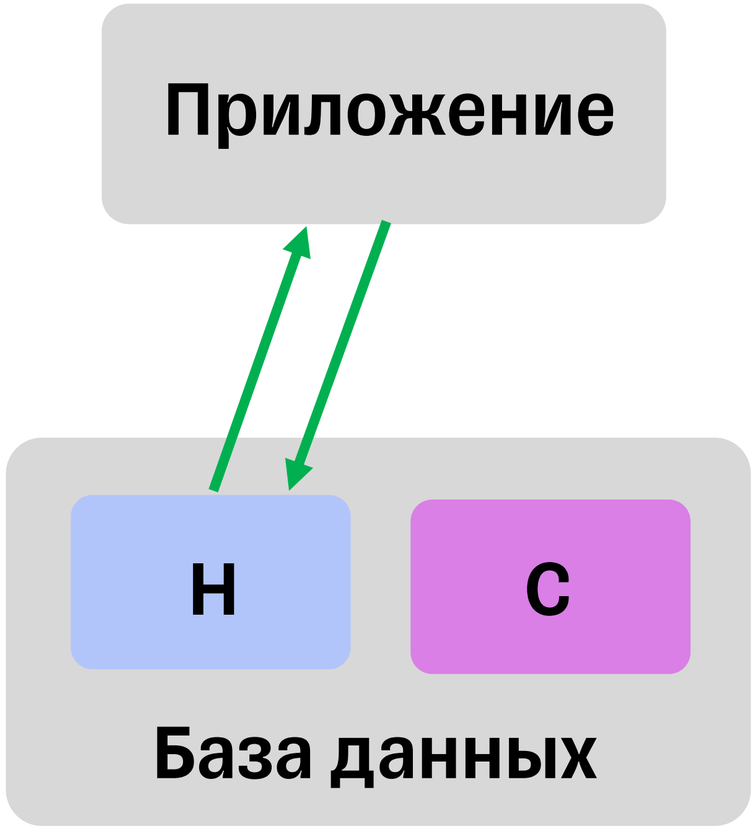
Релиз 1



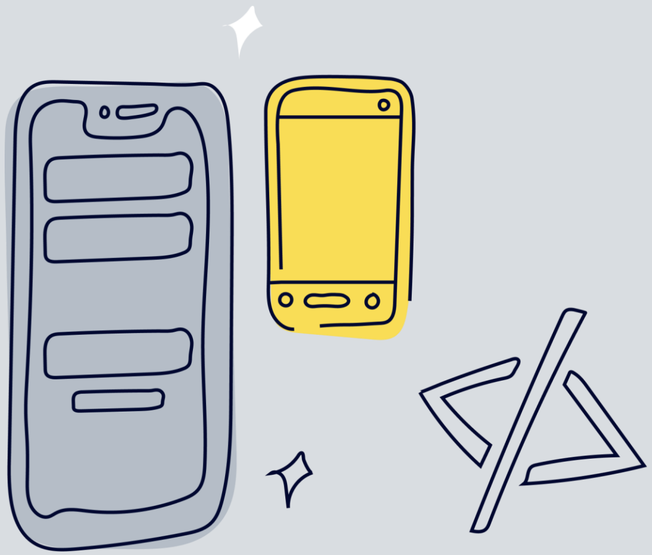
Релиз 2



Релиз 3

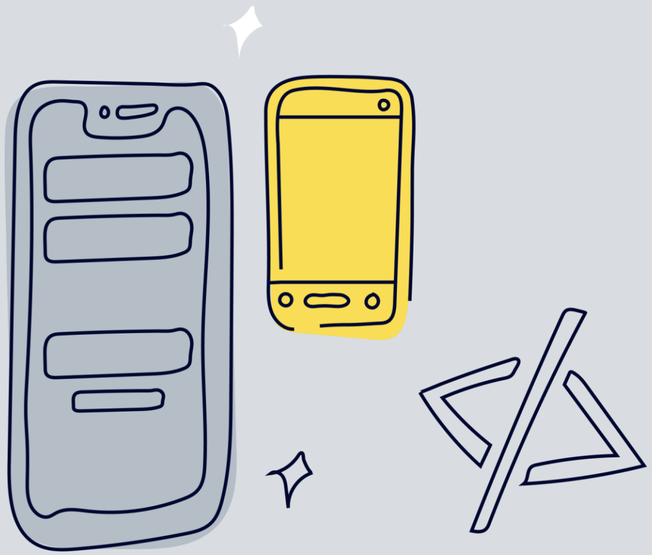


Итоги по несовместимым операциям:



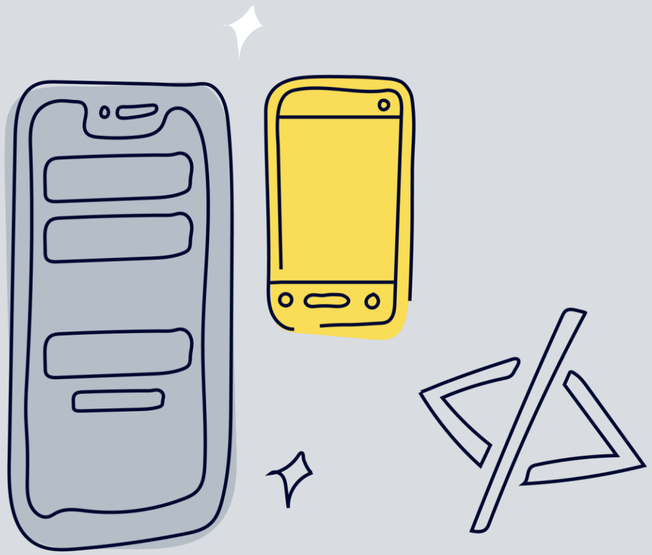
- Преобразуем несовместимую операцию в последовательность совместимых

Итоги по несовместимым операциям:



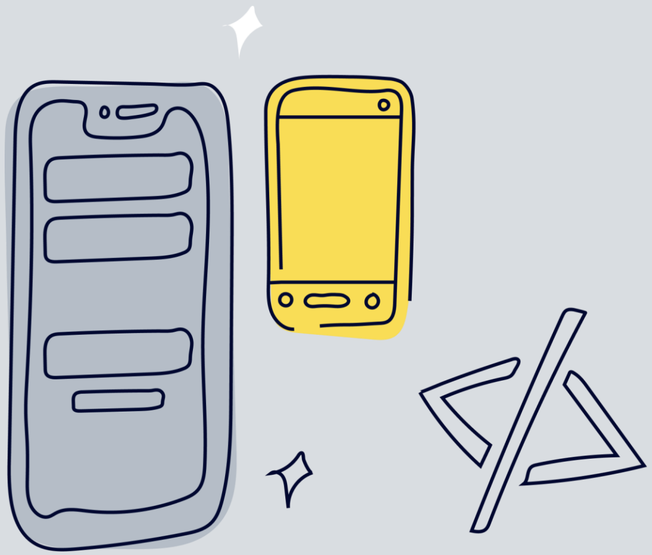
- Преобразуем несовместимую операцию в последовательность совместимых
- Совместимость нужна как по схеме так и по данным

Итоги по несовместимым операциям:



- Преобразуем несовместимую операцию в последовательность совместимых
- Совместимость нужна как по схеме так и по данным
- Много маленьких релизов и миграций

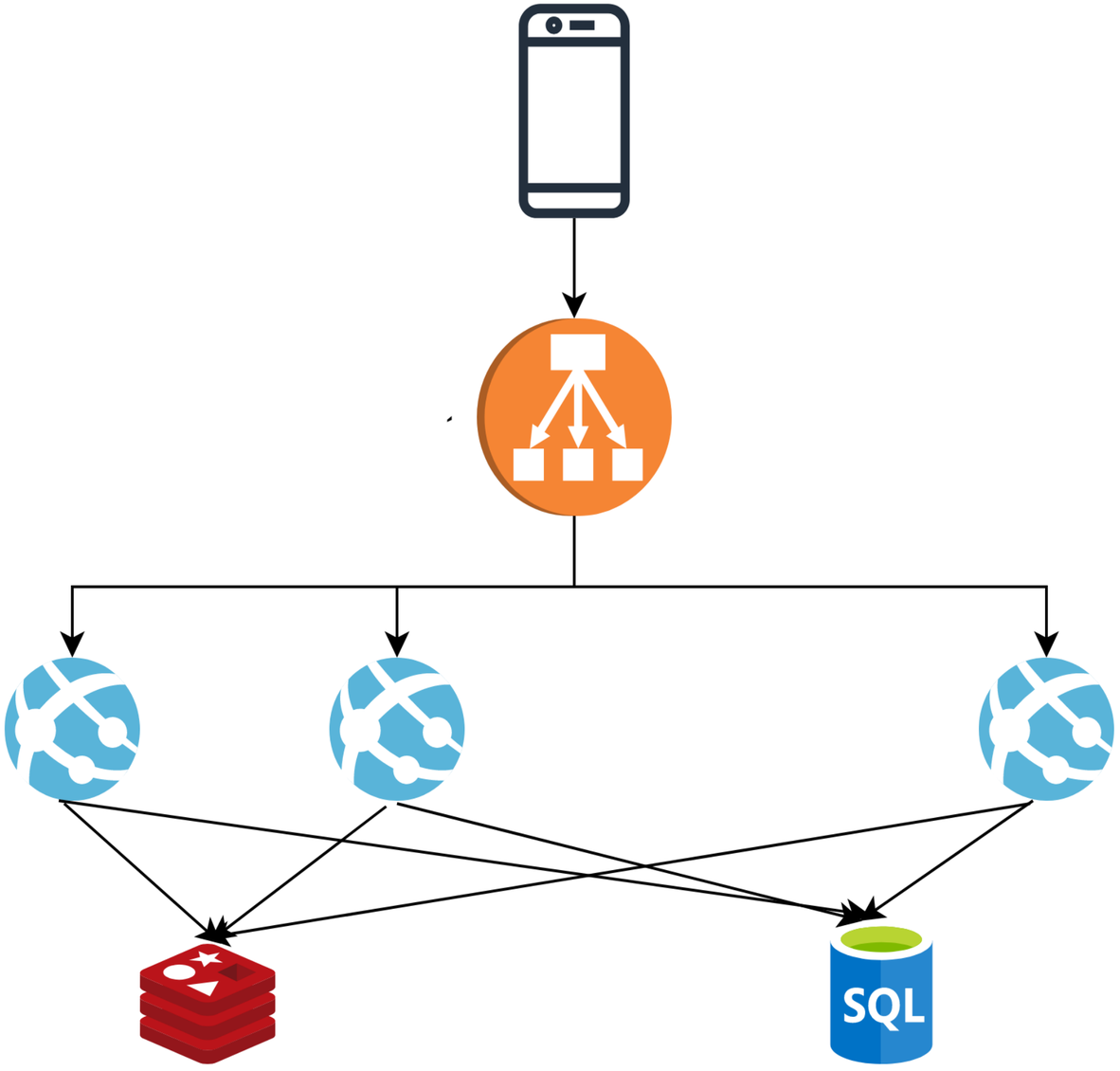
Итоги по несовместимым операциям:



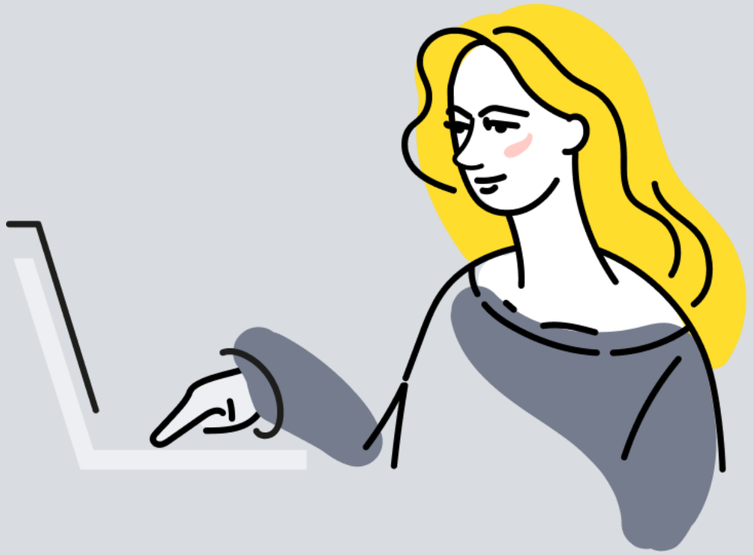
- Преобразуем несовместимую операцию в последовательность совместимых
- Совместимость нужна как по схеме так и по данным
- Много маленьких релизов и миграций
- Миграции EF Core придется исправлять

5. NoSql in action

Схема системы

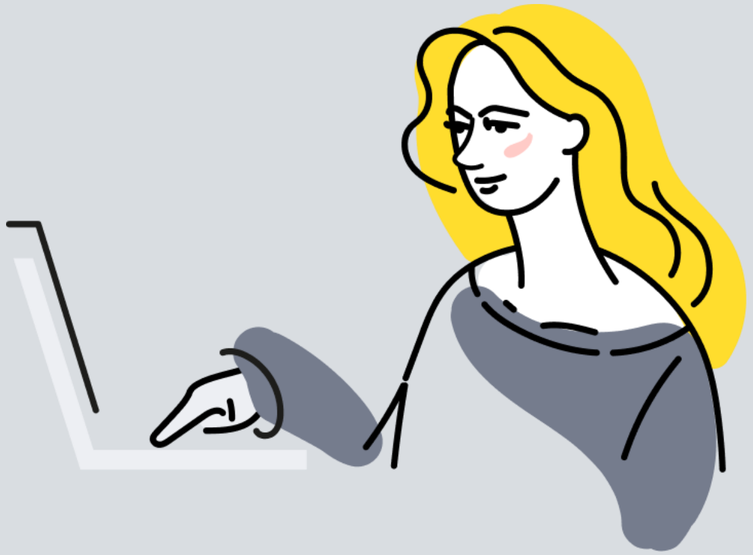


Релиз новой версии



- Сделали релиз новой версии сервиса

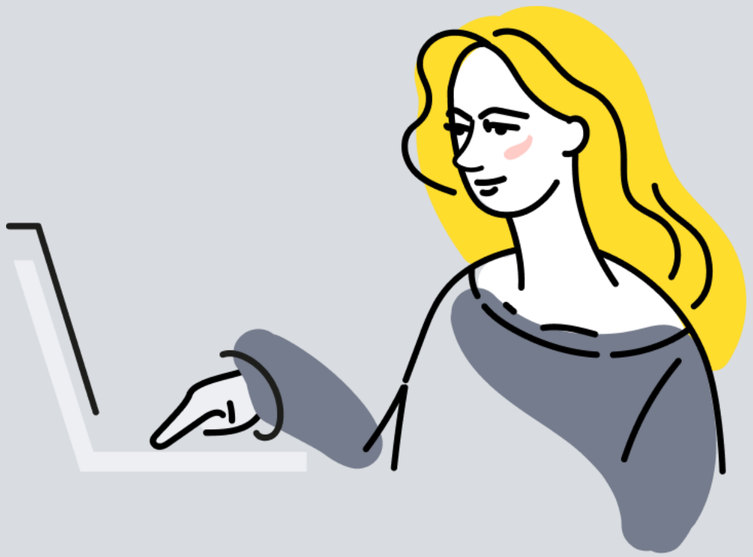
Релиз новой версии



Сделали релиз новой версии сервиса

Через несколько минут появляются ошибки

Релиз новой версии

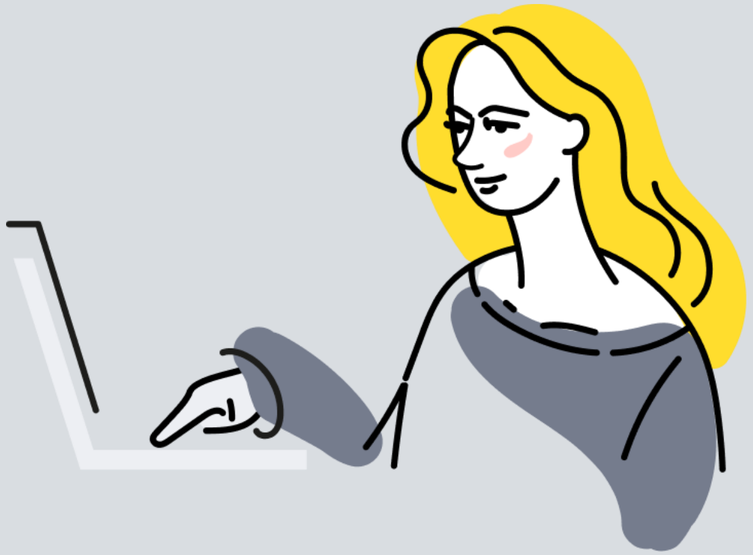


Сделали релиз новой версии сервиса

Через несколько минут появляются ошибки

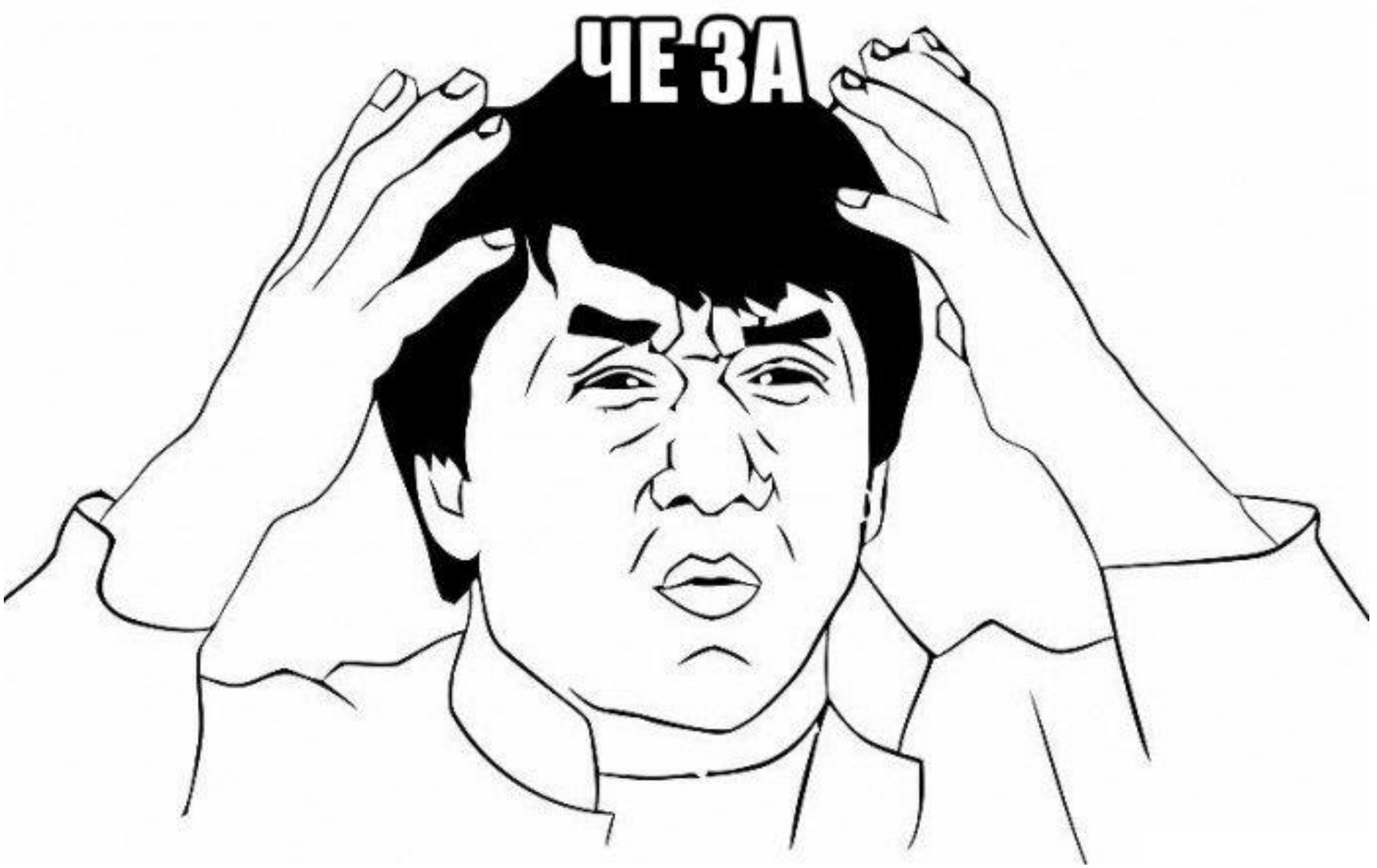
Причину ошибок пока не выяснили. Откатываемся на прошлую версию

Релиз новой версии



- Сделали релиз новой версии сервиса
- Через несколько минут появляются ошибки
- Причину ошибок пока не выяснили. Откатываемся на прошлую версию
- После отката количество ошибок уменьшается, но они не пропадают полностью

Что-то пошло не так



Что делать?

Смотреть в мониторинг!

Что делать?

Смотреть в мониторинг!

Логи, метрики, трейсы...

Делаем!

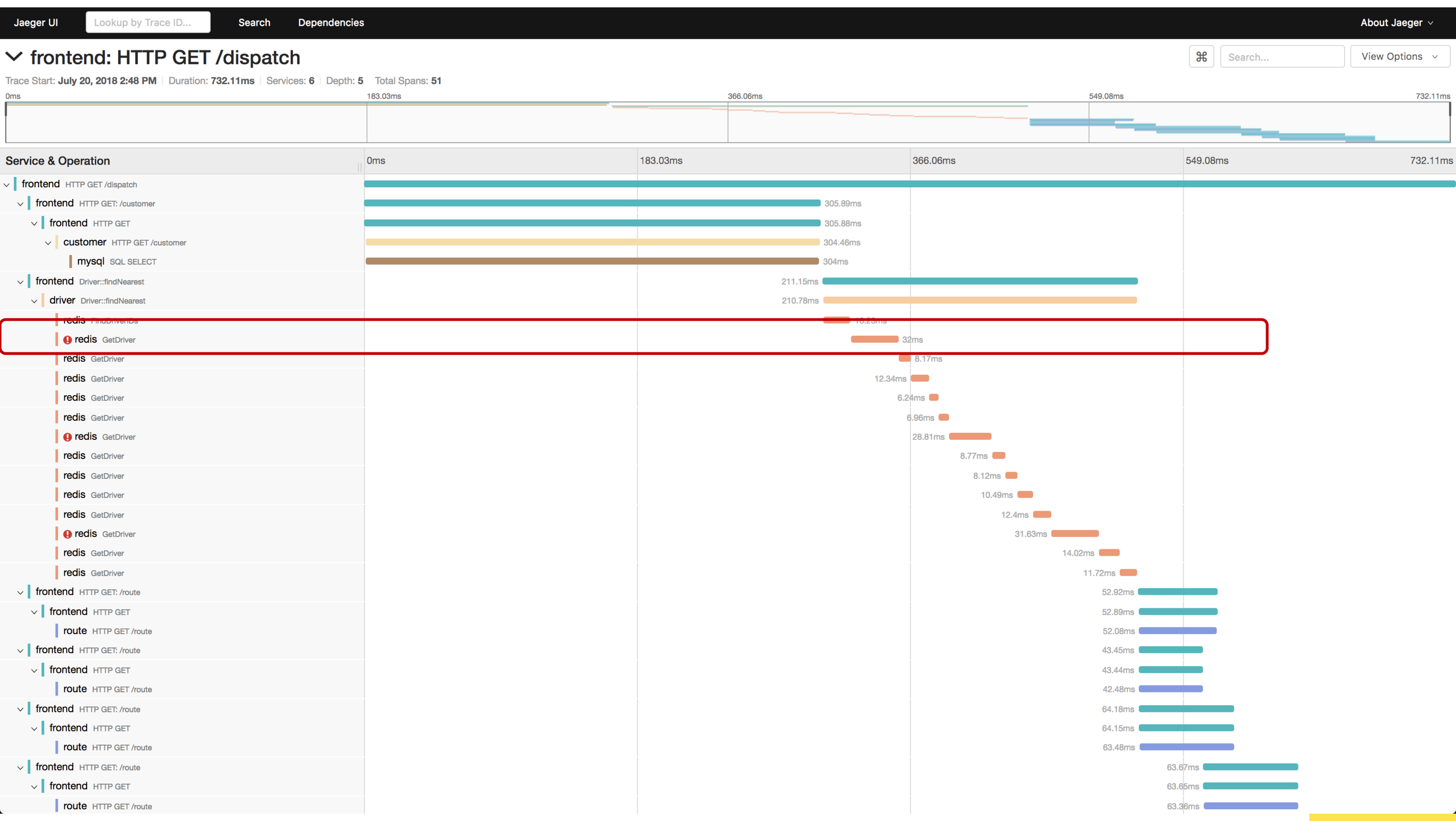
В логах ищем упавший запрос level = ERROR

```
System.Text.Json.JsonException: The JSON value could not be converted to ModelV1.
Path: $.Category | LineNumber: 0 | BytePositionInLine: 27.
 ---> System.InvalidOperationException: Cannot get the value of a token type 'String' as a number.
```

Делаем!

В логах ищем упавший запрос level = ERROR

По trace_id смотрим подробную информацию


Jaeger UI
Lookup by Trace ID...
Search
Dependencies
About Jaeger
frontend: HTTP GET /dispatch
Search...
View Options
Trace Start: July 20, 2018 2:48 PM | Duration: 732.11ms | Services: 6 | Depth: 5 | Total Spans: 51
0ms
183.03ms
366.06ms
549.08ms
732.11ms
Service & Operation
frontend HTTP GET /dispatch
frontend HTTP GET: /customer
305.89ms
frontend HTTP GET
305.88ms
customer HTTP GET /customer
304.46ms
mysql SQL SELECT
304ms
frontend Driver::findNearest
211.15ms
driver Driver::findNearest
210.78ms
redis GetDriver
32ms
redis GetDriver
8.17ms
redis GetDriver
12.34ms
redis GetDriver
6.24ms
redis GetDriver
6.96ms
redis GetDriver
28.81ms
redis GetDriver
8.77ms
redis GetDriver
8.12ms
redis GetDriver
10.49ms
redis GetDriver
12.4ms
redis GetDriver
31.63ms
redis GetDriver
14.02ms
redis GetDriver
11.72ms
frontend HTTP GET: /route
52.92ms
frontend HTTP GET
52.89ms
route HTTP GET /route
52.08ms
frontend HTTP GET: /route
43.45ms
frontend HTTP GET
43.44ms
route HTTP GET /route
42.48ms
frontend HTTP GET: /route
64.18ms
frontend HTTP GET
64.15ms
route HTTP GET /route
63.48ms
frontend HTTP GET: /route
63.67ms
frontend HTTP GET
63.65ms
route HTTP GET /route
63.36ms

Что произошло:

- В старой модели было поле типа int а в новой – стало типа string

Что произошло:

В старой модели было поле типа int а в новой – стало типа string

Старое значение в кэше не читается новой версией, а новое - не читается старой версией

Что произошло:

В старой модели было поле типа int а в новой – стало типа string

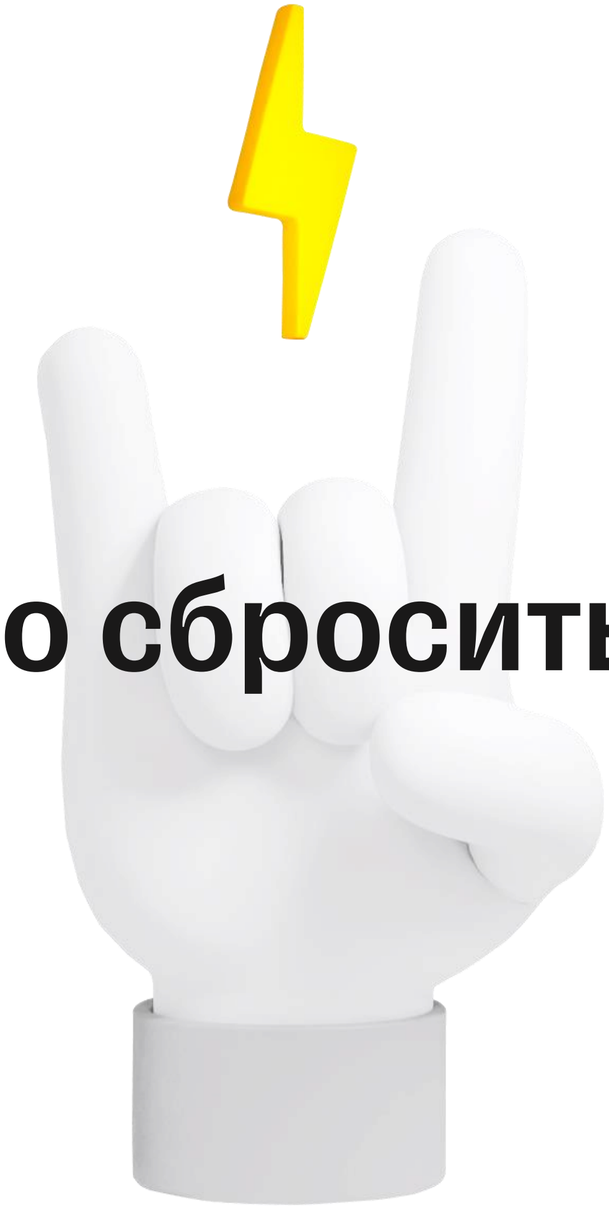
Старое значение в кэше не читается новой версией, а новое - не читается старой версией

В релизе изменилась схема кэша!

Что произошло:

```
public record ModelV1(int Category);
public record ModelV2(string Category);

public class Cache
{
    private readonly IDistributedCache _distributedCache;

    public ModelV1? GetModel(string key)
    {
        byte[]? bytes = _distributedCache.Get(key);

        if (bytes == null)
            return null;

        string json = Encoding.UTF8.GetString(bytes, 0, bytes.Length);

        return JsonSerializer.Deserialize<ModelV1>(json);
    }
}
```

Как чинить?



Нужно сбросить кэш

Продолжаем разговор

- Посыпались таймауты

Продолжаем разговор

- Посыпались таймауты
- Потребление CPU на базе подскакивает до 100%

Продолжаем разговор

- Посыпались таймауты
- Потребление CPU на базе подскакивает до 100%
- Или растет нагрузка на используемые сервисы

Продолжаем разговор

- Посыпались таймауты
- Потребление CPU на базе подскакивает до 100%
- Или растет нагрузка на используемые сервисы
- Или растет потребление ресурсов на инстансах

Приложение может работать без кэша ?

Что делать-то:

- Запустить подготовленный скрипт для прогрева кэша

Что делать-то:

- Запустить подготовленный скрипт для прогрева кэша
- Шаманство, например:
  - уменьшить кол-во подключений к базе
  - подождать пока кэш прогреется
  - вернуть кол-во подключений к базе на место

Что делать-то:

Запустить подготовленный скрипт для прогрева кэша

Шаманство, например:
- уменьшить кол-во подключений к базе
- подождать пока кэш прогреется
- вернуть кол-во подключений к базе на место

Как предотвратить такие ошибки?

Кэш это тоже данные!

А значит тоже нужно думать об обратной совместимости

Кэш это тоже данные!

А значит тоже нужно думать об обратной совместимости

Миграции для кэша писать не имеет смысла

Кэш это тоже данные!

А значит тоже нужно думать об обратной совместимости

Миграции для кэша писать не имеет смысла

Ключ кэширования должен зависеть от версии сервиса!
1.1.0-mymodel-1
1.2.0-mymodel-2

Добавили версию приложения в ключ:

```
private static string Version => "1.1.0";

public ModelV1? GetModel(string id)
{
    string cacheKey = $"{Version}_model_{id}";
    byte[]? bytes = _distributedCache.Get(cacheKey);

    if (bytes == null)
        return null;

    string json = Encoding.UTF8.GetString(bytes, 0, bytes.Length);

    return JsonSerializer.Deserialize<ModelV1>(json);
}
```

Что получили

Плюсы

Просто и безопасно

Работает во время релиза

Можно откатиться на старую версию

Что получили

Плюсы

Просто и безопасно

Работает во время релиза

Можно откатиться на старую версию

Минусы

Увеличение размера кэша в 2 раза на время релиза

Снова проблема холодного старта

Что делать с холодным стартом?

- Ничего

Что делать с холодным стартом?

- Ничего
- Не допускать сброс кэша

Что делать с холодным стартом?

- Ничего
- Не допускать сброс кэша
- Скрипт для прогрева (можно встроить в pipeline)

Что делать с холодным стартом?

- Ничего
- Не допускать сброс кэша
- Скрипт для прогрева (можно встроить в pipeline)
- Плавное включение нового функционала (можно использовать фича флаги)

Версия приложения в ключе:

```
private static string Version => "1.1.0";

public Category? GetCategory(string categoryId)
{
    string cacheKey = $"{Version}_categories_{categoryId}";
    //
}

public Product? GetProduct(int productId)
{
    string cacheKey = $"{Version}_products_{productId}";
    //
}
```

Версия схемы в ключе:

```
public CategoryV1? GetCategoryV1(string categoryId)
{
    string cacheKey = $"v1_categories_{categoryId}";
    //
}
public CategoryV2? GetCategoryV2(string categoryId)
{
    string cacheKey = $"v2_categories_{categoryId}";
    //
}
public Product? GetProduct(int productId)
{
    string cacheKey = $"v1_products_{productId}";
    //
}
```

Что получили

Плюсы

Работает во время релиза

Можно откатиться на старую версию

Нет проблемы холодного старта

Нет увеличения размера кэша во время релиза

Можно использовать старую и новую версию одновременно

Что получили

Плюсы

Работает во время релиза

Можно откатиться на старую версию

Нет проблемы холодного старта

Нет увеличения размера кэша во время релиза

Можно использовать старую и новую версию одновременно

Минусы

Нужно следить за актуальность схемы

Конвертация старой версии в новую:

```
public CategoryV2? GetCategory(string categoryId)
{
    var categoryV2 = GetCategoryV2(categoryId);

    if (categoryV2 != null)
        return categoryV2;

    var categoryV1 = GetCategoryV1(categoryId);

    if (categoryV1 != null)
        return categoryV1.ConvertToV2();

    return null;
}
```

Использование двух версий в логике:

```
string categoryId = "";

var categoryV2 = _cache.GetCategoryV2(categoryId);

if (categoryV2 != null)
    return RunCategoryV2RelatedCode(categoryV2);

var categoryV1 = _cache.GetCategoryV1(categoryId);

if (categoryV1 != null)
    return RunCategoryV1RelatedCode(categoryV1);

// Load categoryV2 from the DB ...
```

А ведь могло быть и хуже...

В новой версии удалили обязательное поле

А ведь могло быть и хуже...

В новой версии удалили обязательное поле

И после отката на старую версию – Deserialize работает!

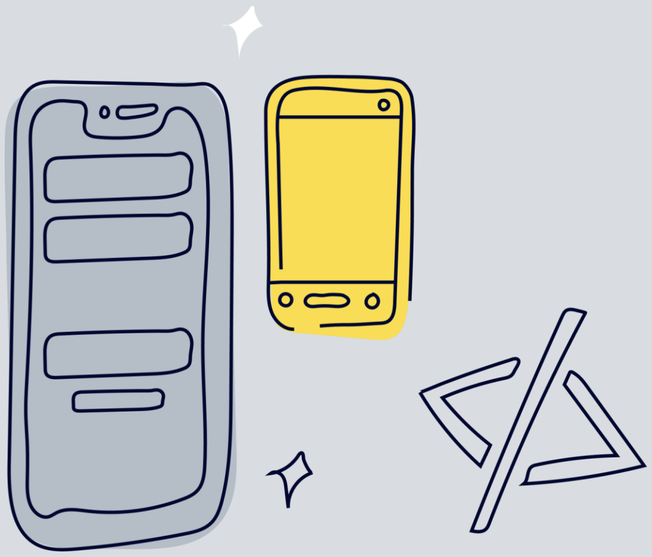
А ведь могло быть и хуже...

В новой версии удалили обязательное поле

И после отката на старую версию – Deserialize работает!

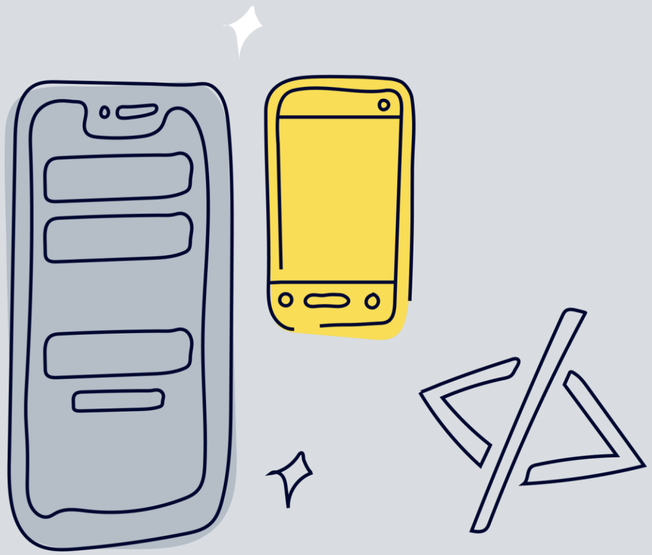
Но начинаются ошибки в логике и понять причину намного сложнее

Итоги по кэшу:



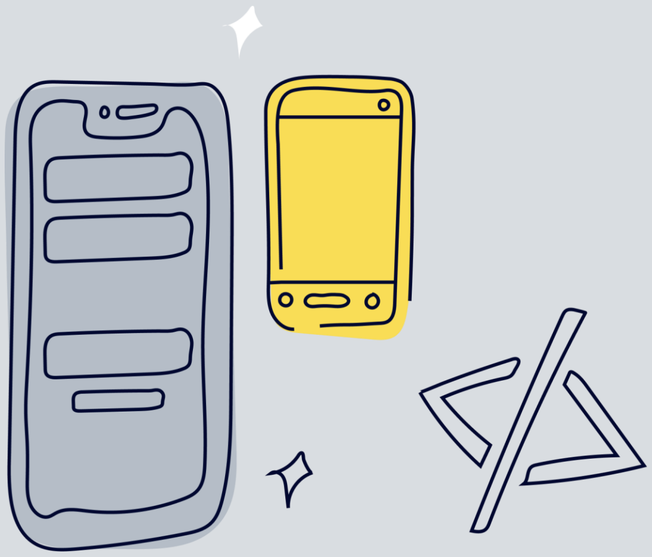
- Есть неявная схема данных на уровне приложения

Итоги по кэшу:



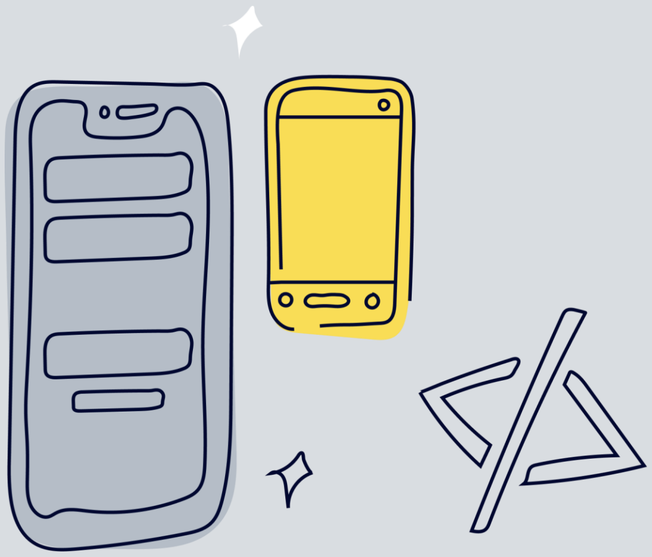
- Есть неявная схема данных на уровне приложения
- Нужна обратная совместимость

Итоги по кэшу:



- Есть неявная схема данных на уровне приложения
- Нужна обратная совместимость
- Можно использовать версию приложения для всех ключей

Итоги по кэшу:



- Есть неявная схема данных на уровне приложения
- Нужна обратная совместимость
- Можно использовать версию приложения для всех ключей
- Или явно версионировать схему данных

А что насчет Elasticsearch?

Elasticsearch

Table -> Index

Elasticsearch

Table -> Index

Row –> Document

Elasticsearch

- Table -> Index
- Row –> Document
- Тип поля в документе определяется по первому вхождению

```
{
  IntValue: 10
}
```

Rollover

Как партиции в postgres

Rollover

Как партиции в postgres

Таблица – alias, партиция - index

Rollover

Как партиции в postgres

Таблица – alias, партиция - index

alias = my-index

my-index-2024.11.0**5**-000001
my-index-2024.11.0**6**-000001
my-index-2024.11.0**7**-000001

Структурные логи

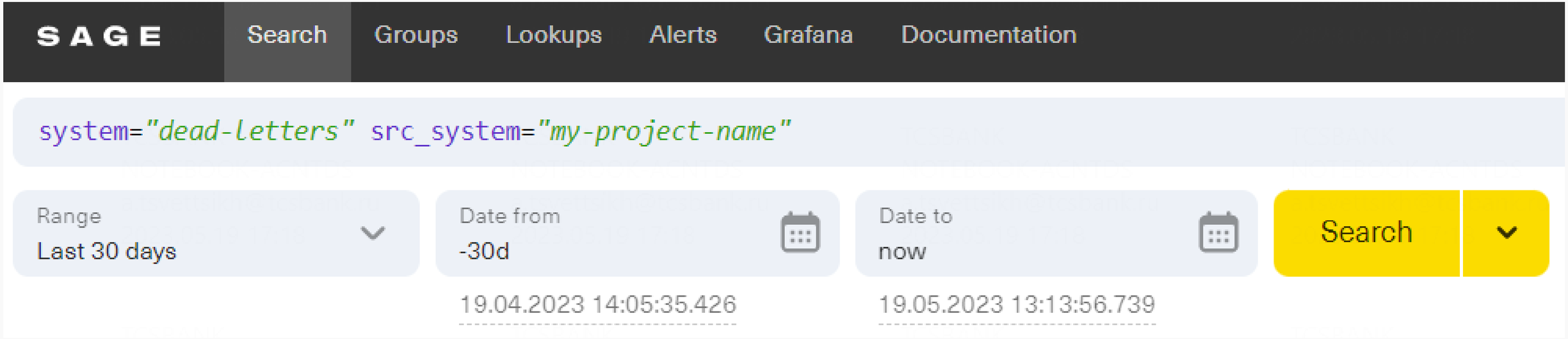
```
_logger.LogInformation("Item number:{Num}", item.IntValue);
```

Если тип Number изменится с int на string?

```
_logger.LogInformation("Item number:{Num}", item.StringValue);
```

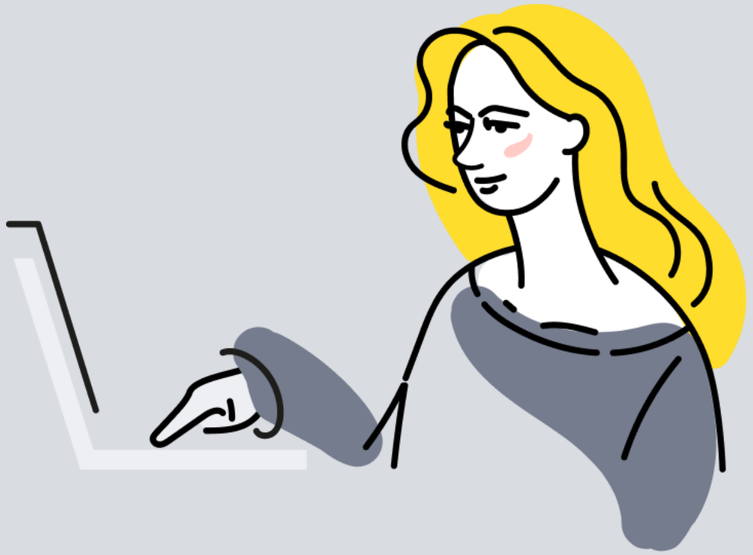
Mapping error detected. Alias <logs.my-project-name>, property name: Num, value: ITEM-123543, type: **String**, expected: **Number**

Dead letters



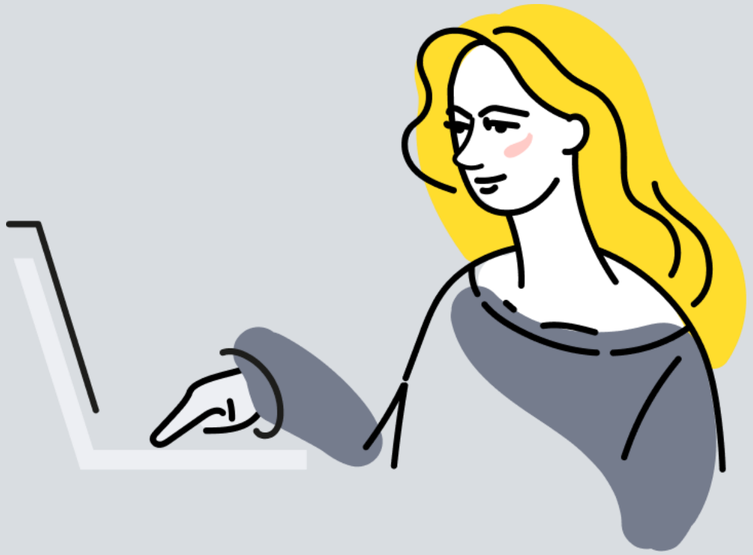
- Нужен alert, если в dead-letters по вашему сервису что-то попало

Что делать с dead-letters



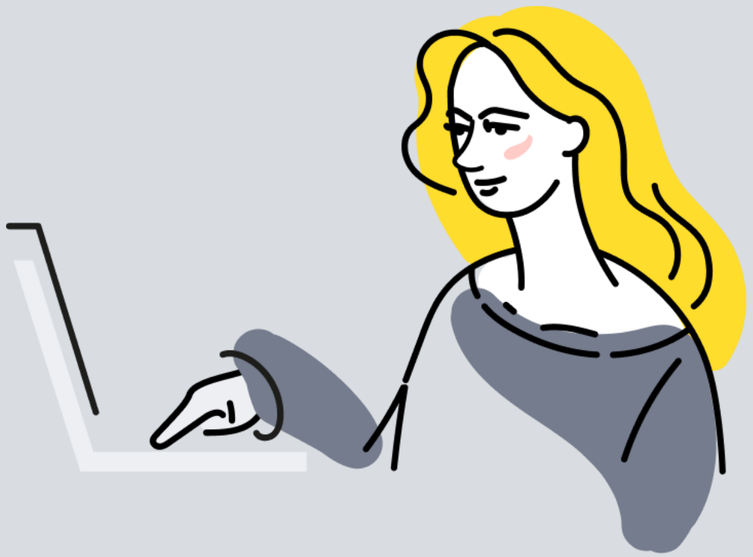
- Можно исправить код и задеплоить

Что делать с dead-letters



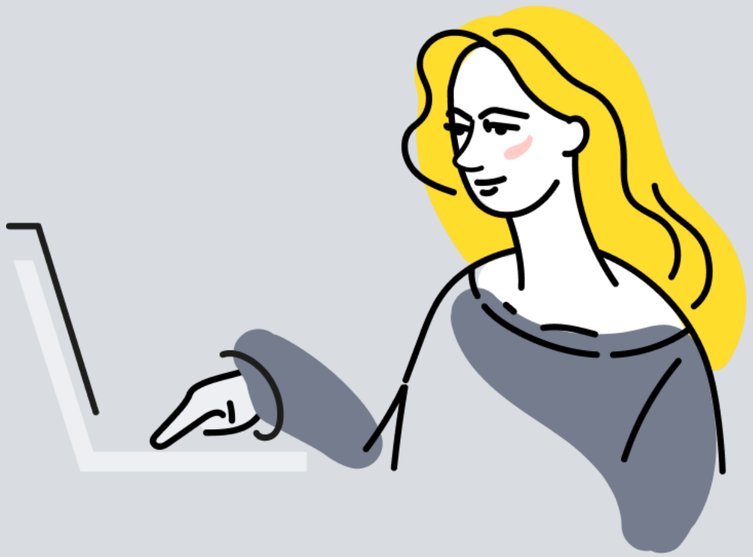
- Можно исправить код и задеплоить
- Ручной rollover

Что делать с dead-letters



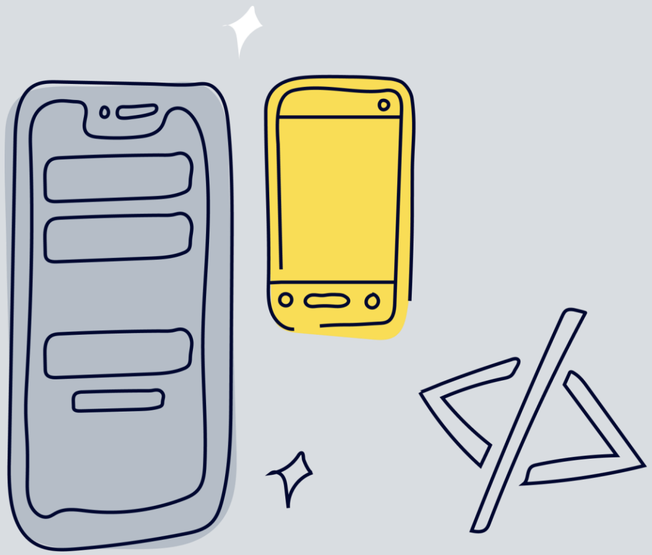
- Можно исправить код и задеплоить
- Ручной rollover
- Ждать автоматического rollover

Что делать с dead-letters



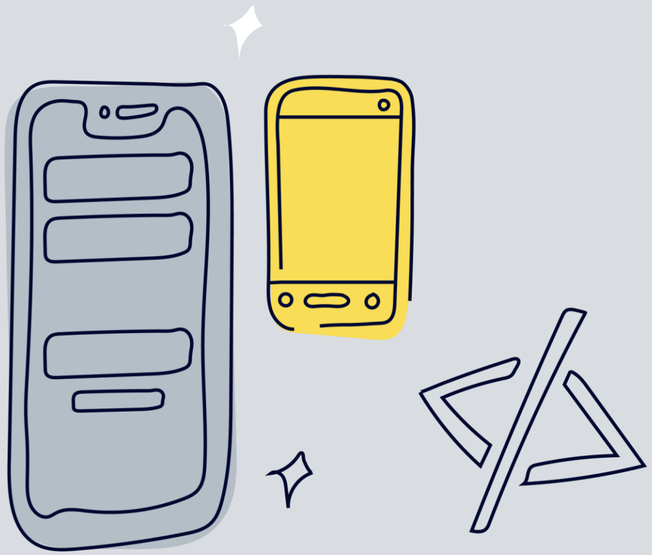
- Можно исправить код и задеплоить
- Ручной rollover
- Ждать автоматического rollover
- А что делать если в dead-letters попали логи от сторонней либы?

Итоги по схеме:



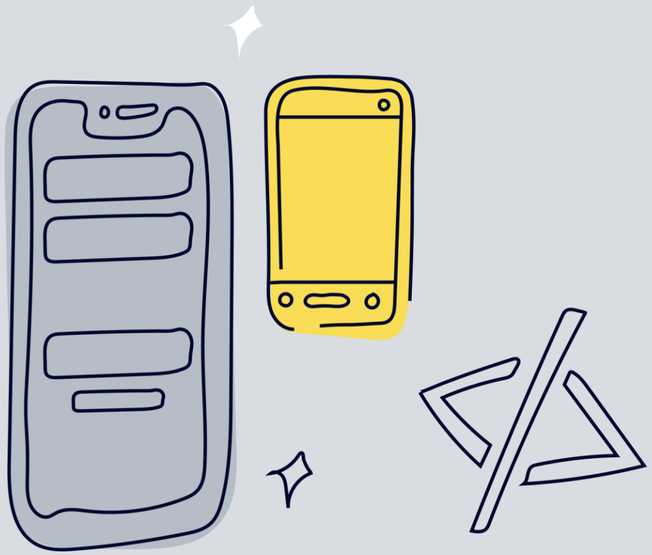
- Postgres – схема на запись

Итоги по схеме:



- Postgres – схема на запись
- Redis – схема на чтение

Итоги по схеме:



- Postgres – схема на запись
- Redis – схема на чтение
- Elasticsearch – что-то среднее

О чем поговорили:

01

Rolling Update/Blue-Green deployment

О чем поговорили:

01

Rolling Update/Blue-Green deployment

02

Запускаем миграции в pipeline

О чем поговорили:

01

Rolling Update/Blue-Green deployment

02

Запускаем миграции в pipeline

03

Мигрируем данные при первом обращении или в фоне

О чем поговорили:

01

Rolling Update/Blue-Green deployment

02

Запускаем миграции в pipeline

03

Мигрируем данные при первом обращении или в фоне

04

Последовательно применяем совместимые по коду и данным операции

О чем поговорили:

01

Rolling Update/Blue-Green deployment

02

Запускаем миграции в pipeline

03

Мигрируем данные при первом обращении или в фоне

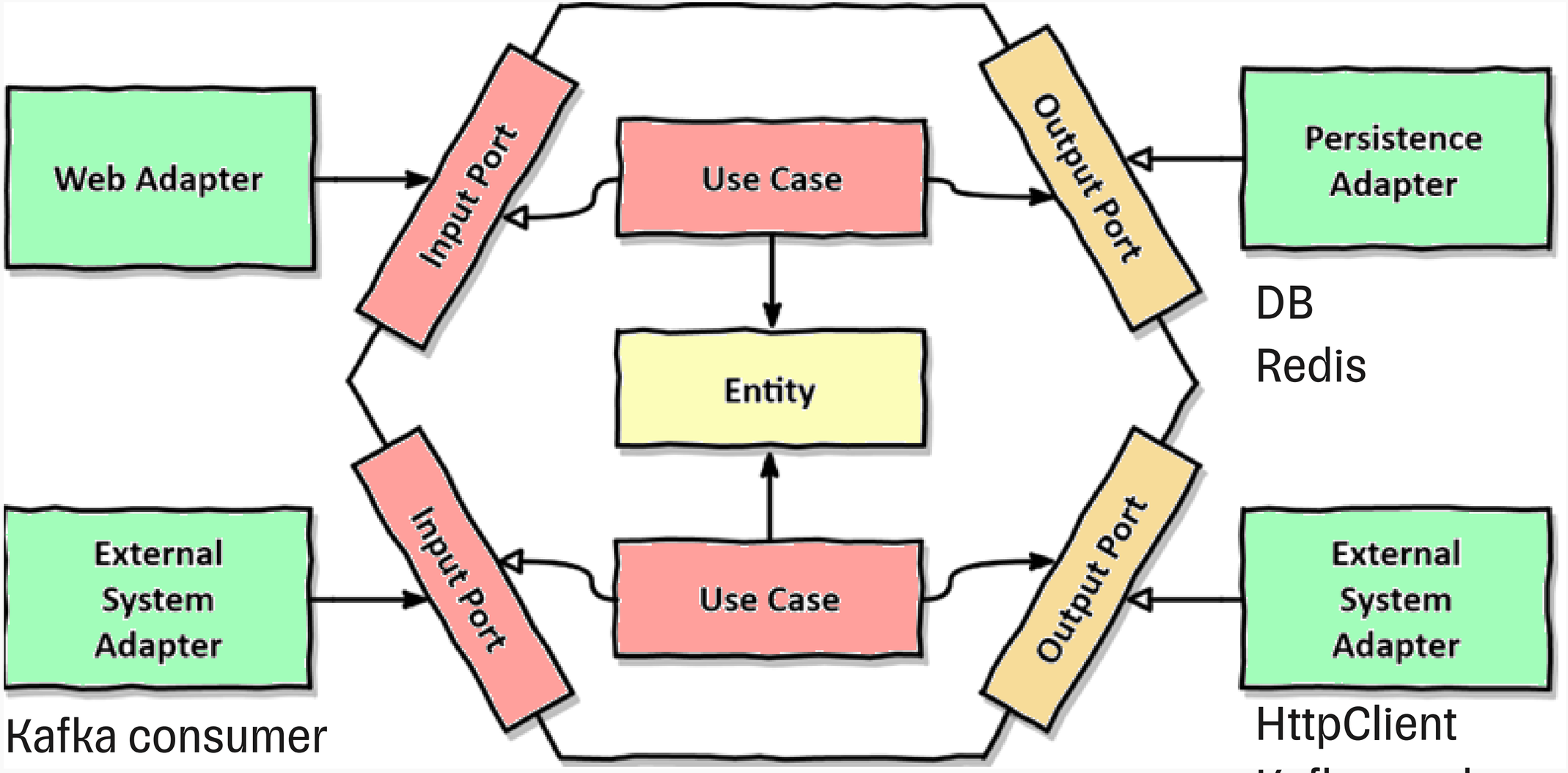
04

Последовательно применяем совместимые по коду и данным операции

05

Версионируем ключи для кэша

Где нужна обратная совместимость



DB
Redis

Kafka consumer

HttpClient
Kafka producer

Background jobs

- Новый job добавляем в код, а потом в базу (триггер для него)

Background jobs

Новый job добавляем в код, а потом в базу (триггер для него)

Удаляем в обратном порядке

Background jobs

Новый job добавляем в код, а потом в базу (триггер для него)

Удаляем в обратном порядке

Обновление = Добавление + Удаление

Очереди сообщений

Интеграционные сообщения – это тоже ваш контракт.

Очереди сообщений

Интеграционные сообщения – это тоже ваш контракт.

Поэтому нельзя удалять обязательные поля

Очереди сообщений

Интеграционные сообщения – это тоже ваш контракт.

Поэтому нельзя удалять обязательные поля

Версионируйте сообщения с самого начала

Материалы по теме

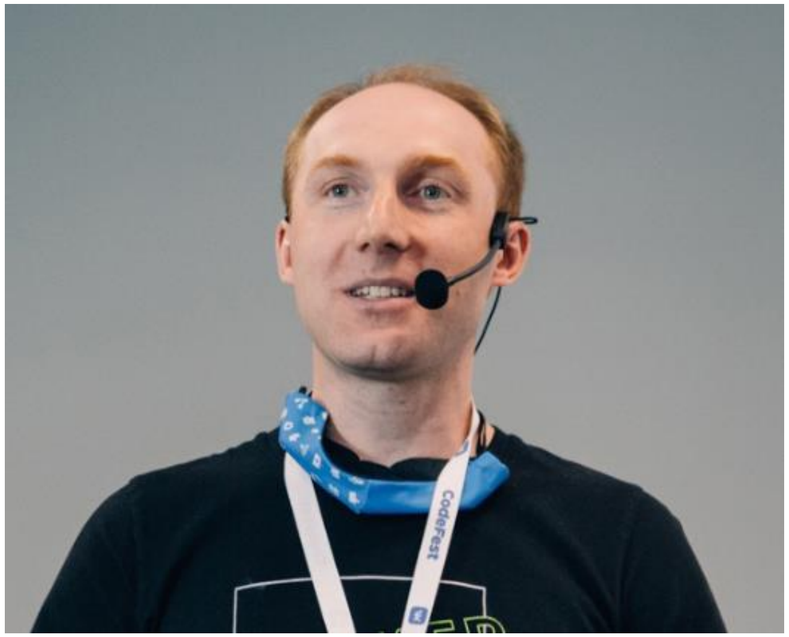
- Refactoring Databases: Evolutionary Database Design
- Распространённые ошибки изменения схемы ... / Николай Самохвалов (Postgres.ai)
- Jeffrey Richter: 'Architecting Distributed Cloud Applications' video series
- Публичные постмортемы
- Большая база данных стала монолитом ... / Евгений Горбачев

Т БАНК

Он такой один

DevBrothers

А нас двое



Спасибо за внимание!

Андрей Цветцих | Ведущий разработчик

Andrew.Tsw@gmail.com @AndrewTsw