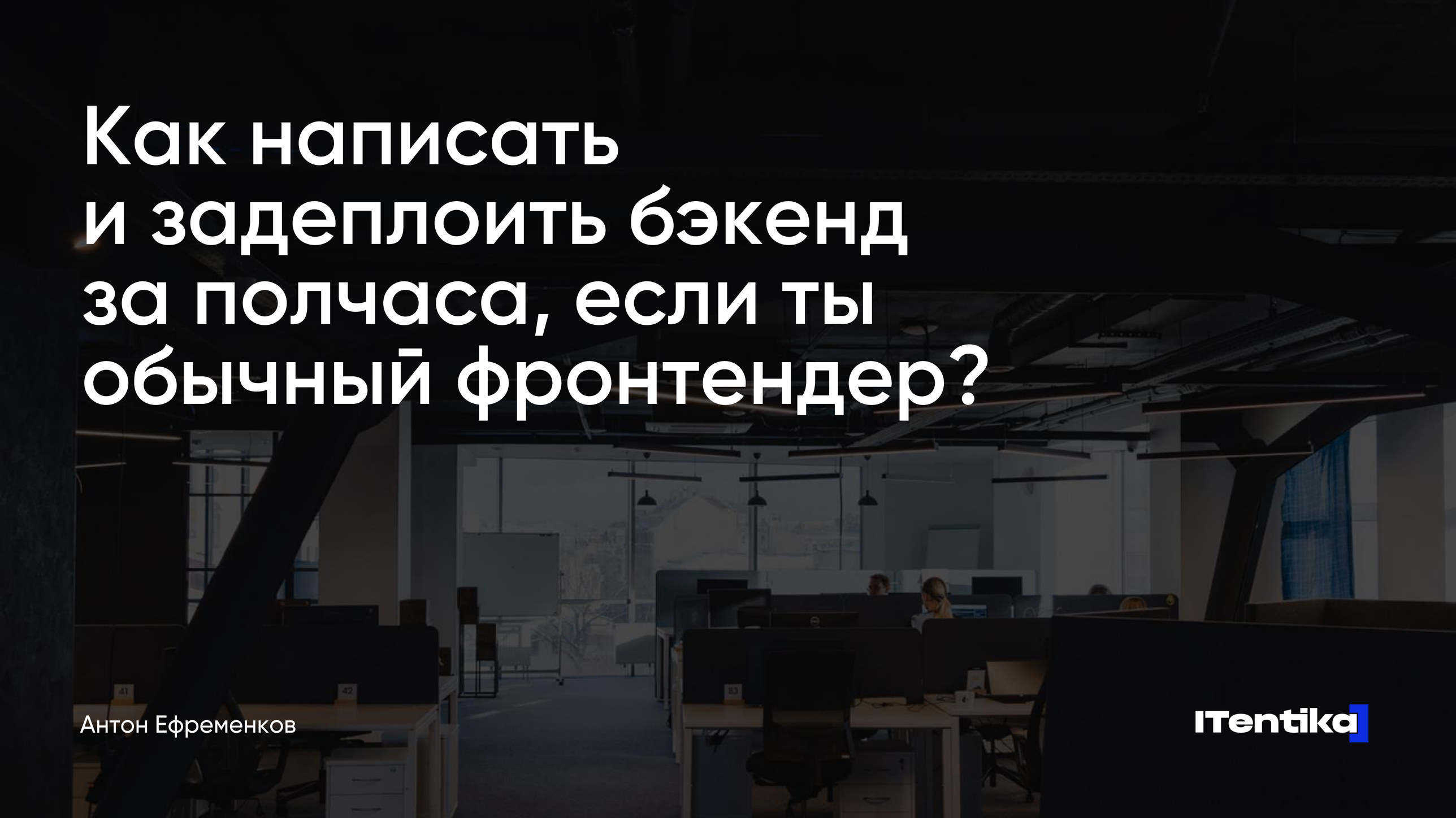
Как написать и задеплоить бэкенд за полчаса, если ты обычный фронтендер?



Антон Ефременков

ITentika

Обо мне

Антон Ефременков

- 13+ лет в IT
- Всё ещё не надоело
- Проектирую, пишу, преподаю
- Променял == на ===

О чём поговорим

- что такое serverless-архитектура

О чём поговорим

- что такое serverless-архитектура
- небольшой интерактив

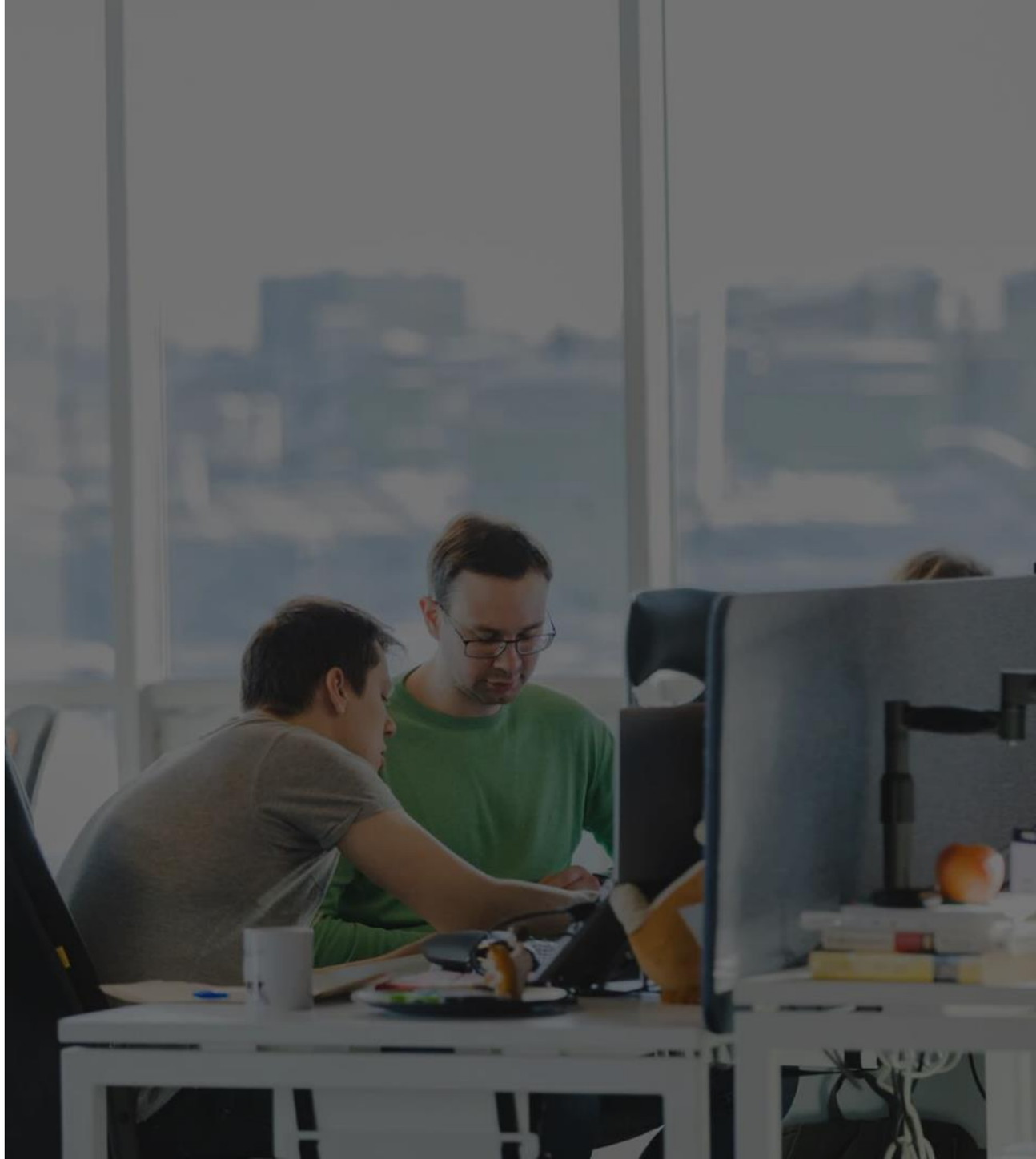
О чём поговорим

- что такое serverless-архитектура
- небольшой интерактив
- serverless на JS в действии

О чём поговорим

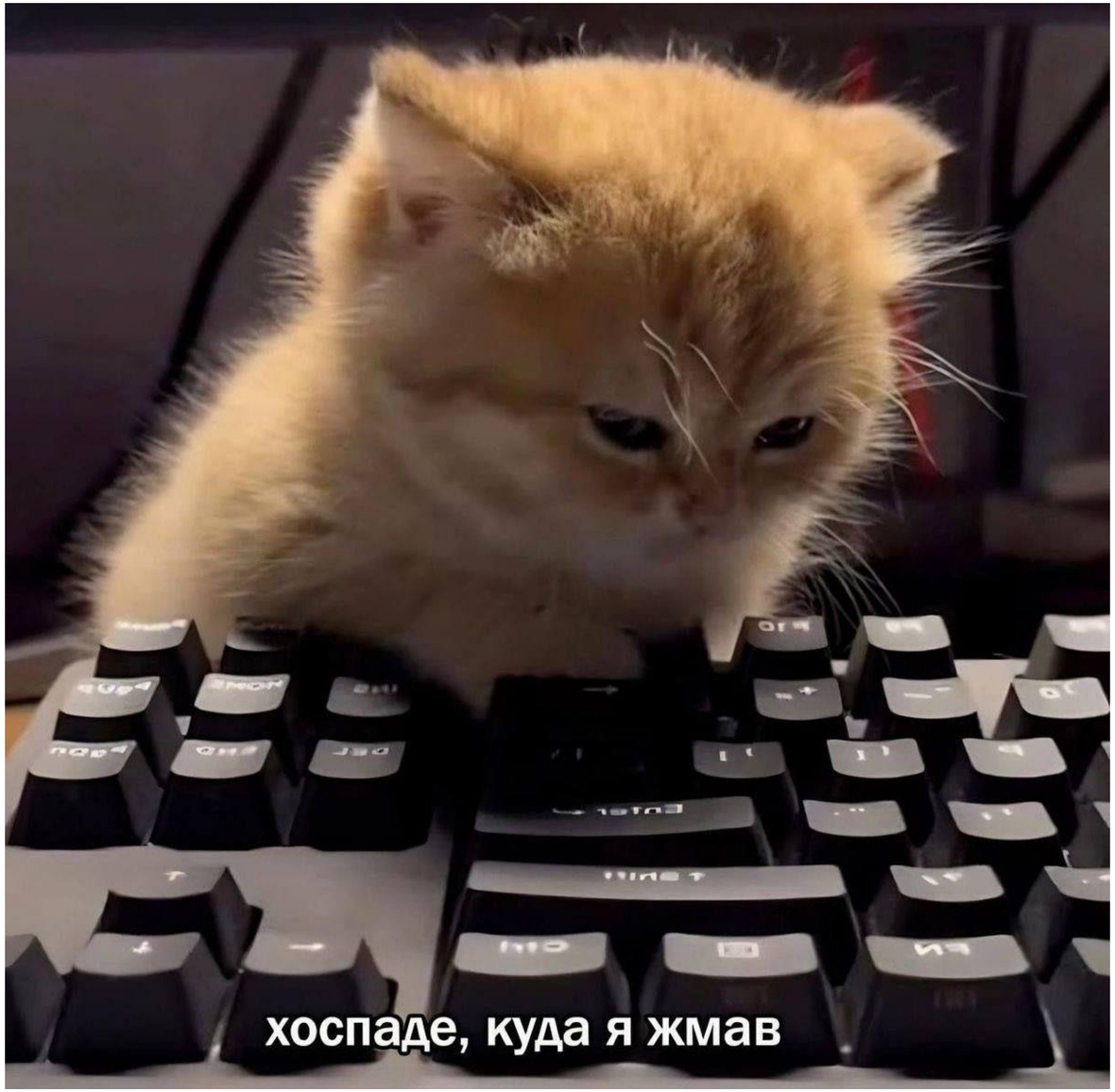
- что такое serverless-архитектура
- небольшой интерактив
- serverless на JS в действии
- выводы

Поговорим про serverless



Проблема?

Я фронтендер, но нужно написать бэк.

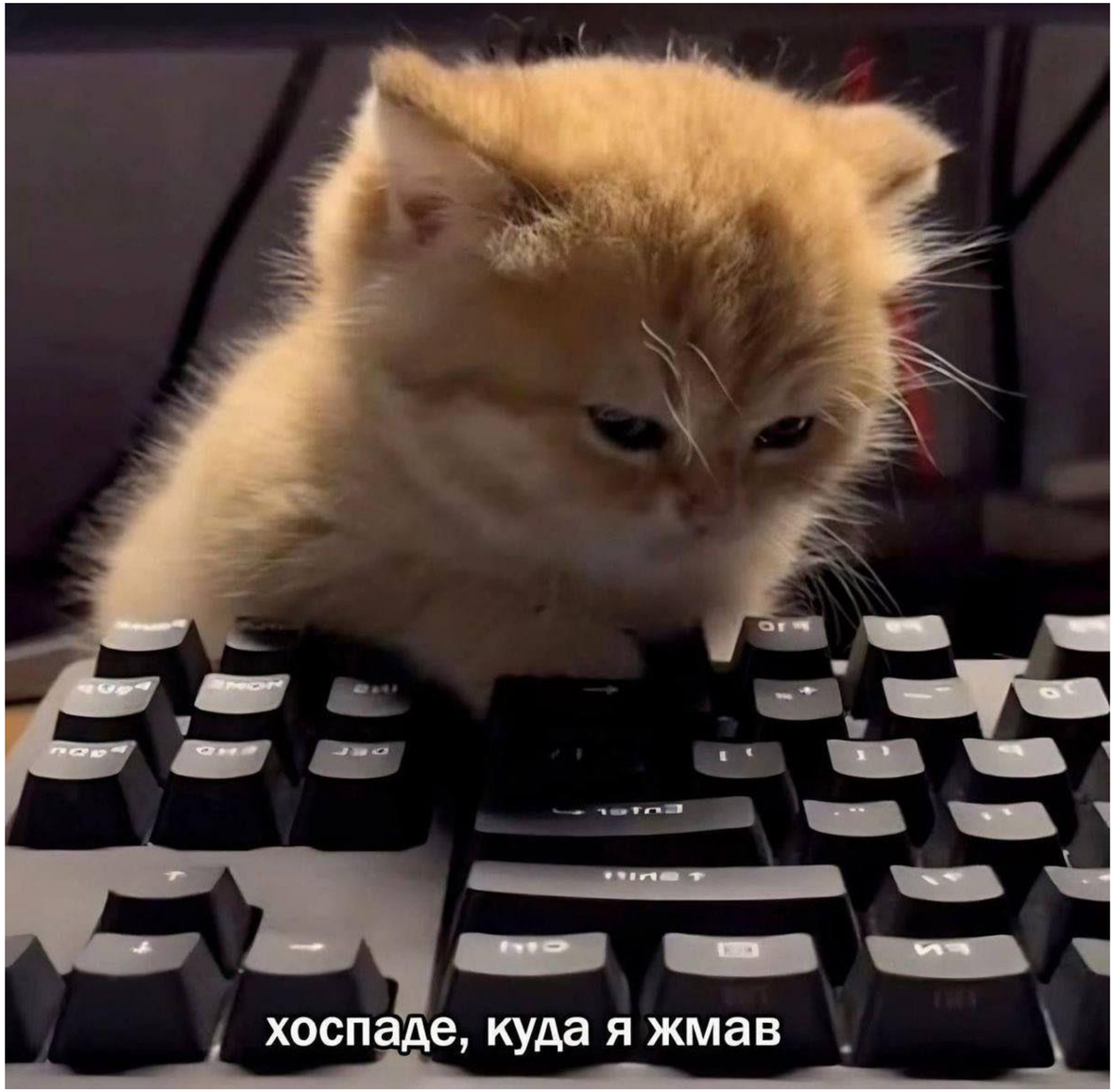


Проблема?

Я фронтендер, но нужно написать бэк.

Зачем?

- проверить гипотезу
- накидать PoC
- микросервисное решение
- написать чат-бот



На чём писать?

Ок, а где хранить данные?

MySQL, PostgreSQL и другие реляционные БД

- жёсткая схема данных
- сложность SQL
- управление связями
- требования к типам данных

Ок, а где хранить данные?

MySQL, PostgreSQL и другие реляционные БД

- жёсткая схема данных
- сложность SQL
- управление связями
- требования к типам данных

Ок, а где хранить данные?

MongoDB

- гибкая схема данных
- хорошая интеграция с JS
- понятный интерфейс запросов
- легко хранить связанные данные
- хорошая документация и сообщество

Ок, а где хранить данные?

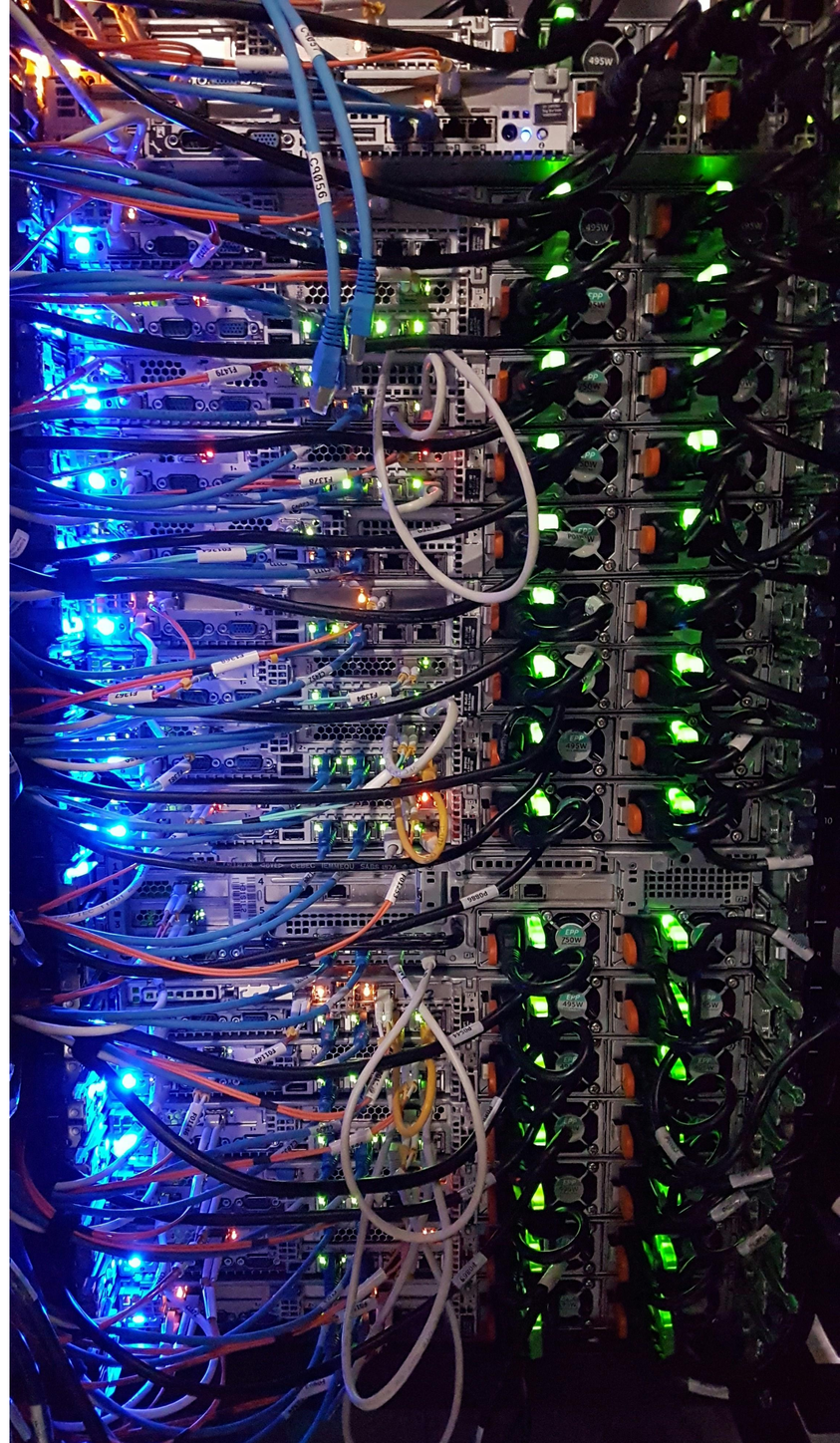
MongoDB

- гибкая схема данных
- хорошая интеграция с JS
- понятный интерфейс запросов
- легко хранить связанные данные
- хорошая документация и сообщество

Чиназес! Сюда!!!

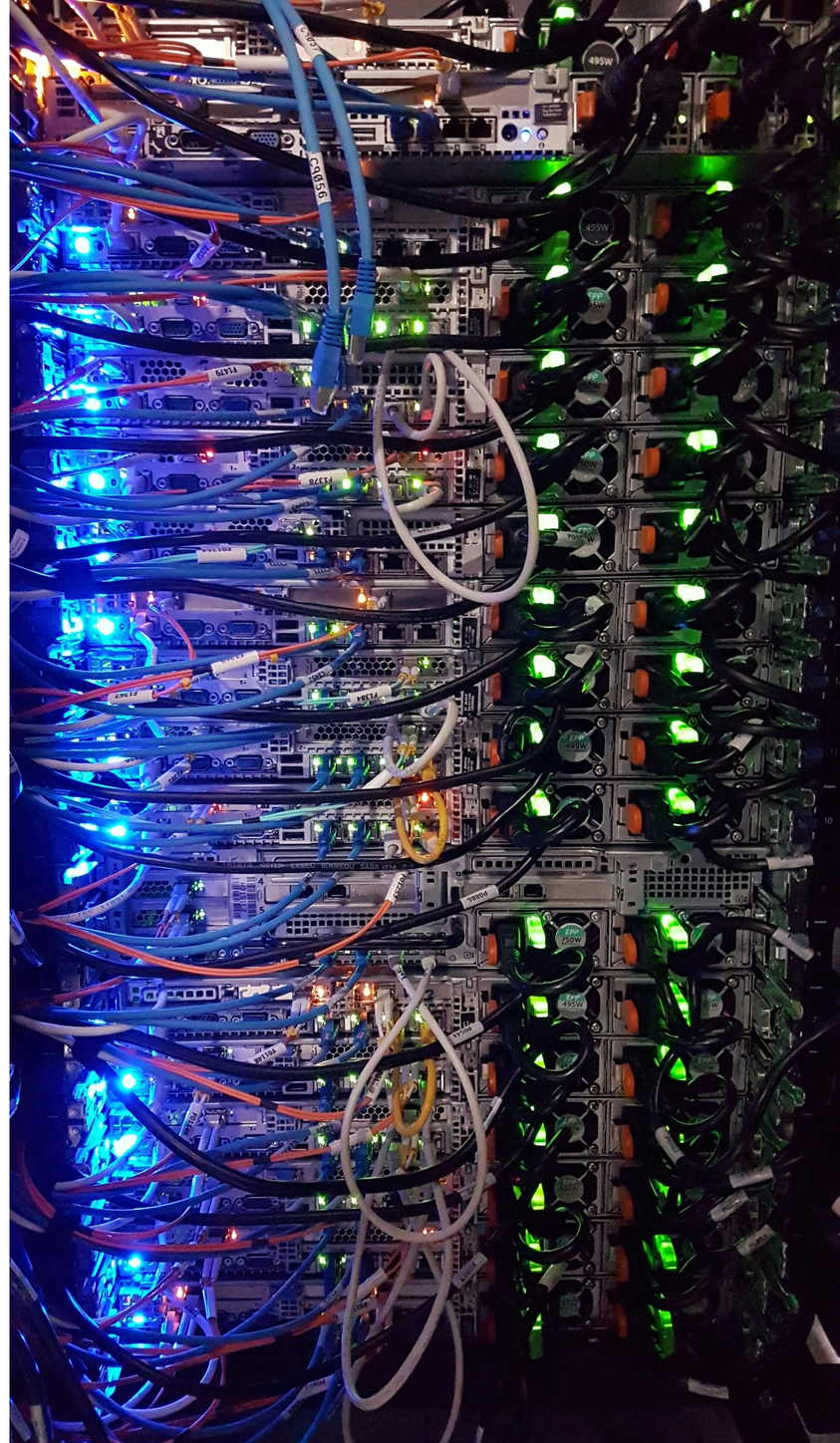
Бэк написан. А где деплоить?

- Nginx/Apache - конфигурить??? Ну нет...



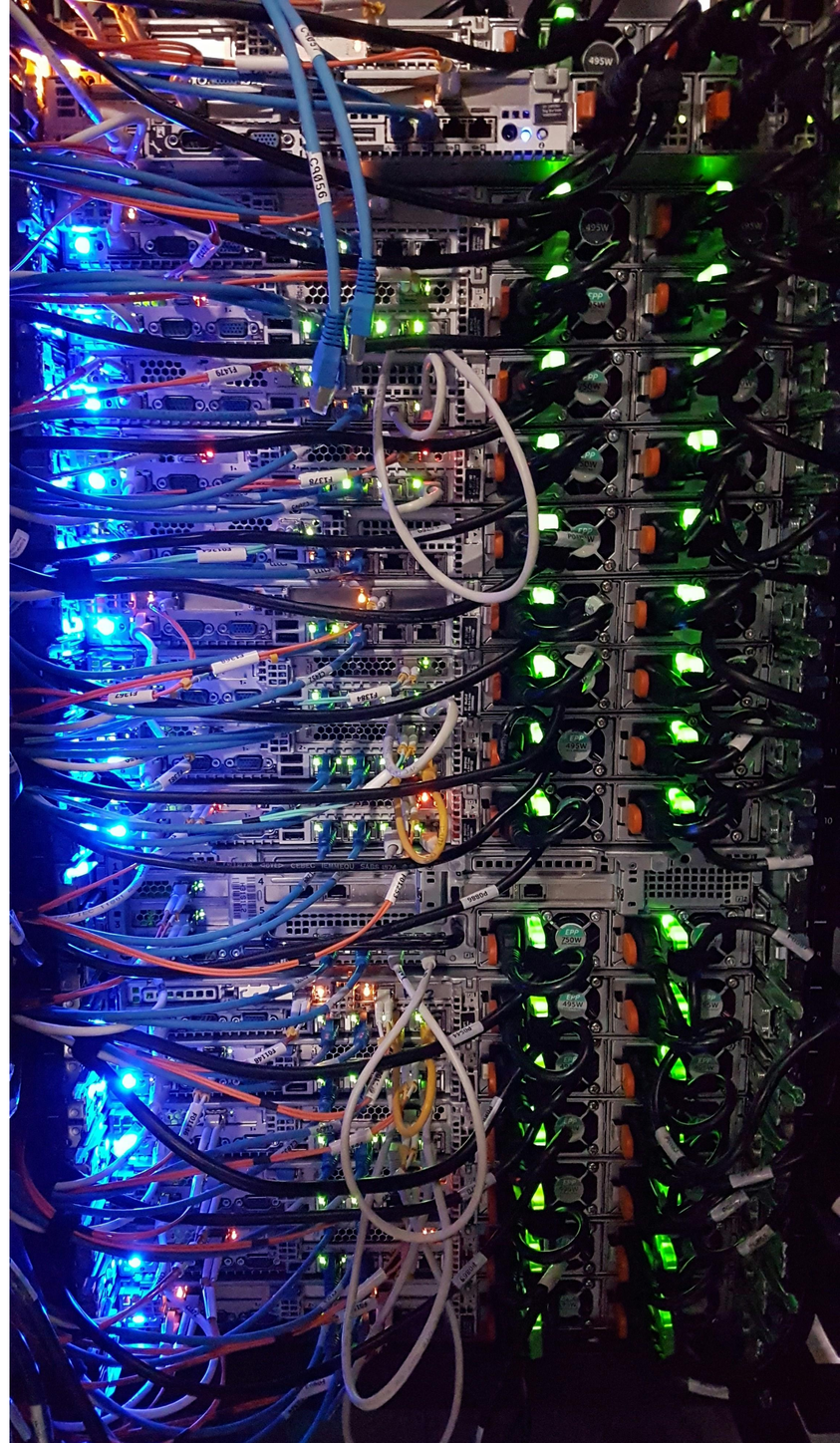
Бэк написан. А где деплоить?

- Nginx/Apache - конфигурить??? Ну нет...
- Heroku - жаль, но в РФ нет (



Бэк написан. А где деплоить?

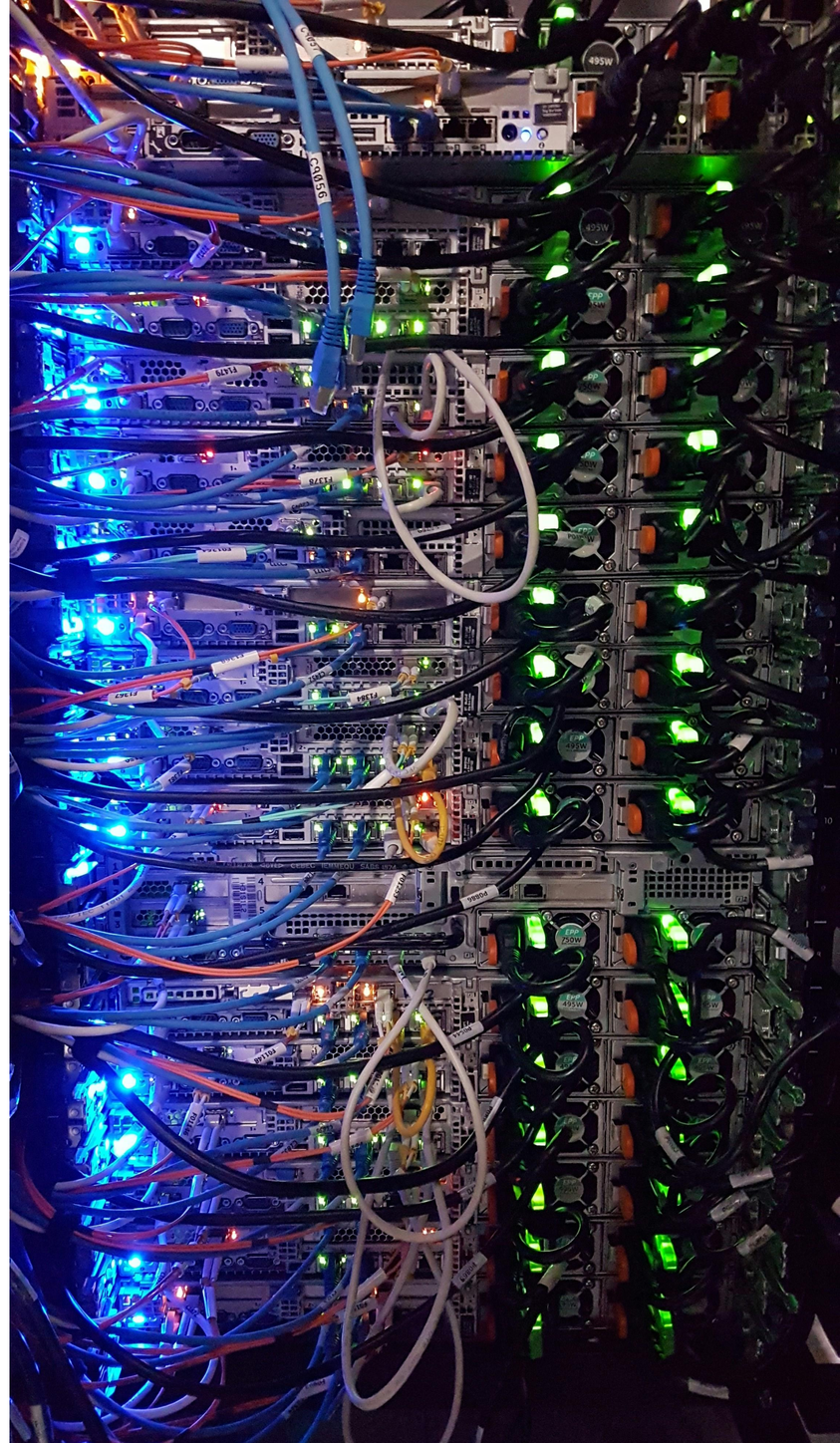
- Nginx/Apache - конфигурить??? Ну нет...
- Heroku - жаль, но в РФ нет (
- IIS - спасибо, но нет...



Бэк написан. А где деплоить?

- Nginx/Apache - конфигурить??? Ну нет...
- Heroku - жаль, но в РФ нет (
- IIS - спасибо, но нет...

А может, ну их эти серверы?..

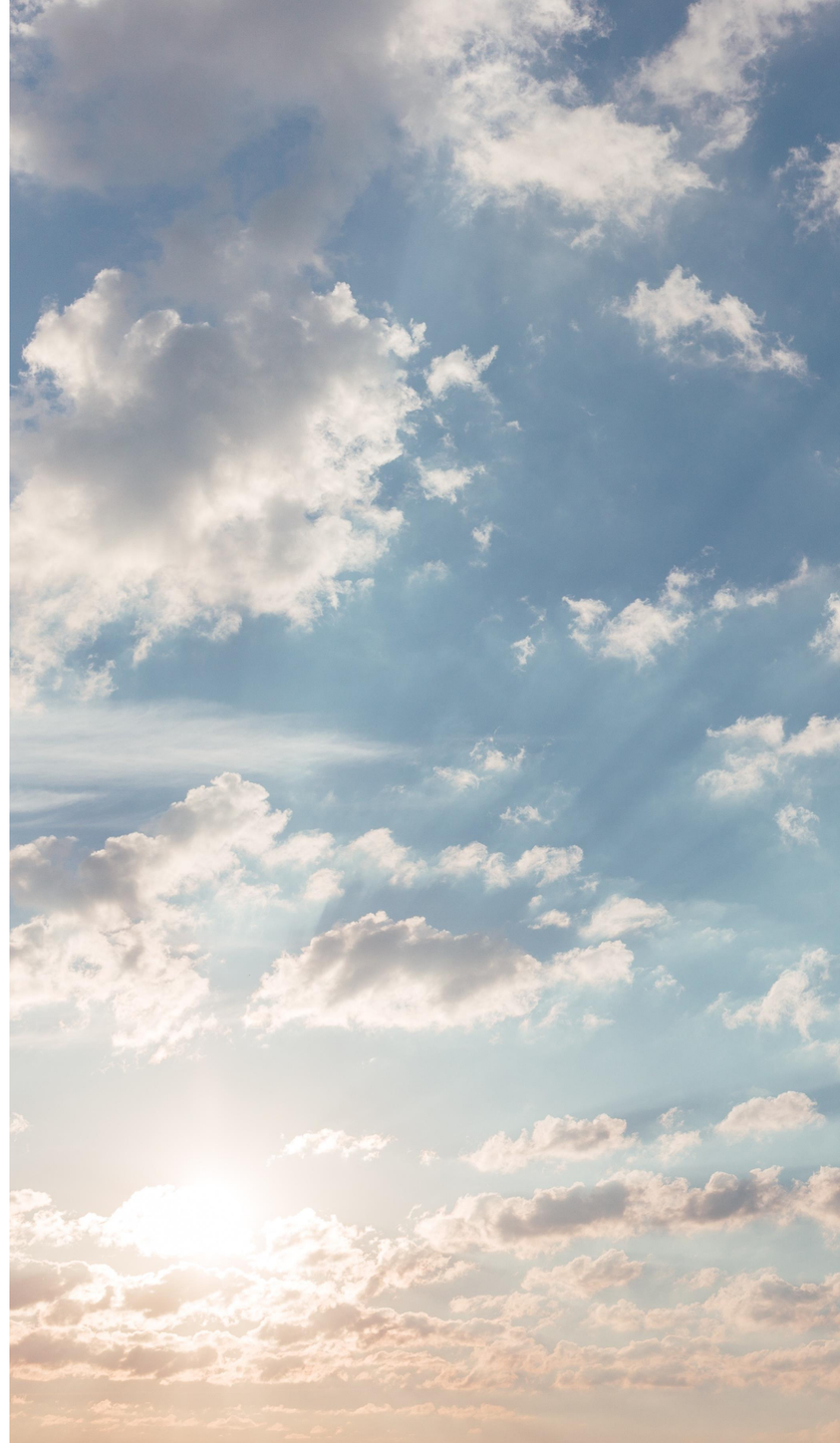


А может, ну эти серверы?

- Serverless, нет серверов нет проблем

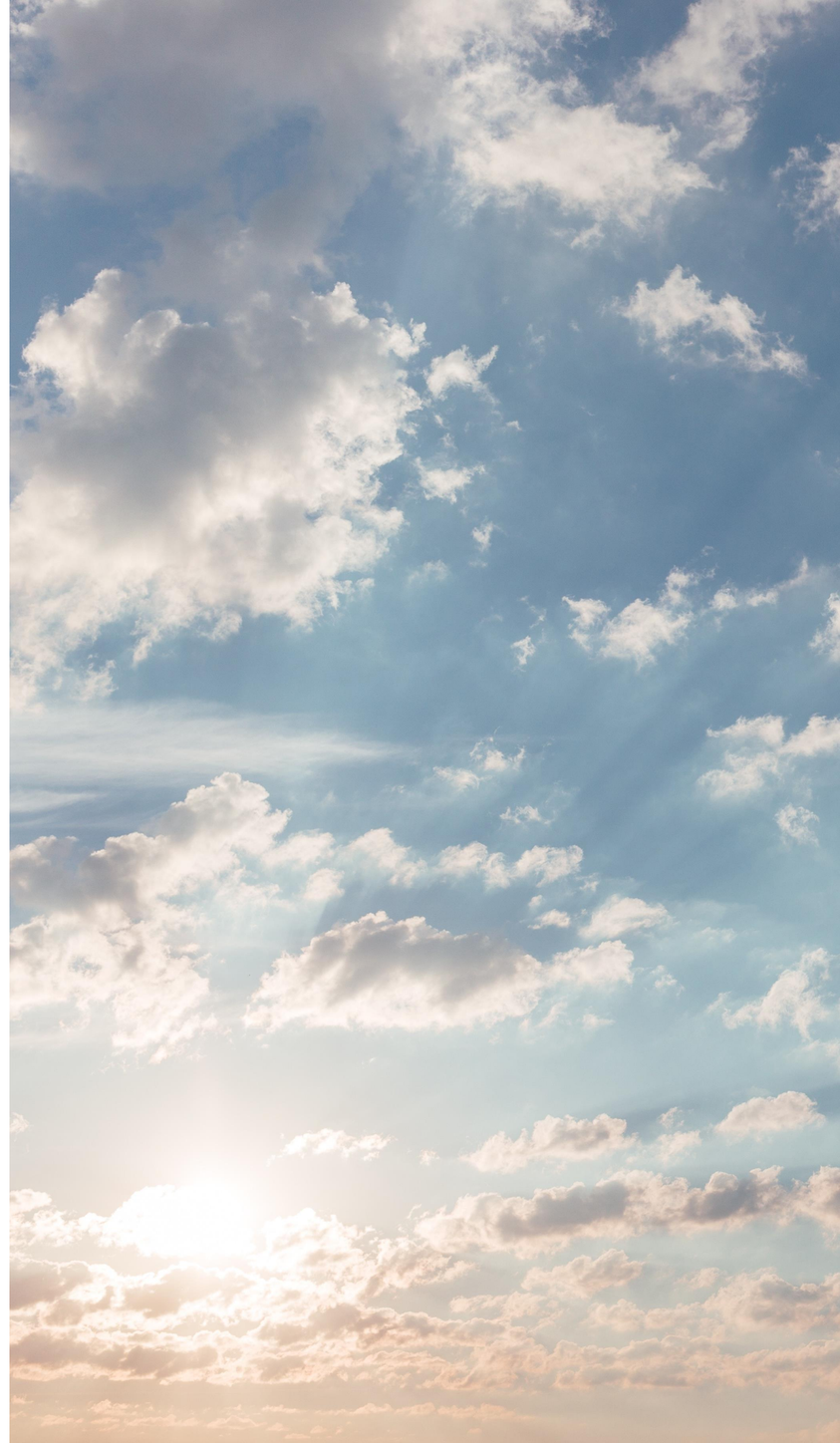
А может, ну эти серверы?

- Serverless, нет серверов нет проблем
- FaaS



А может, ну эти серверы?

- Serverless, нет серверов нет проблем
- FaaS
- Автоматический maintenance



А может, ну эти серверы?

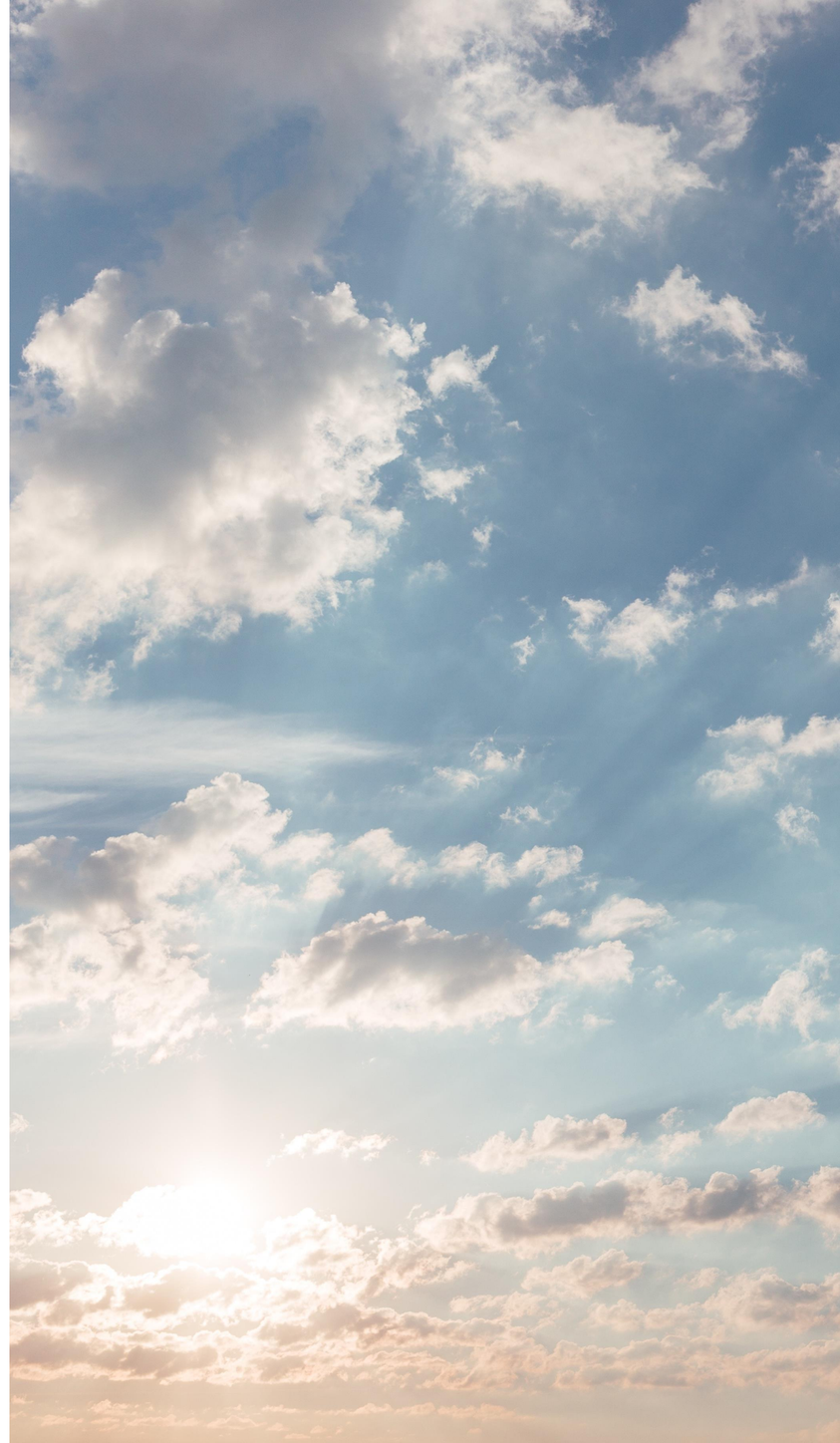
- Serverless, нет серверов нет проблем
- FaaS
- Автоматический maintenance
- Автоматическое управление ресурсами

А может, ну эти серверы?

- Serverless, нет серверов нет проблем
- FaaS
- Автоматический maintenance
- Автоматическое управление ресурсами
- Микросервисная архитектура

А может, ну эти серверы?

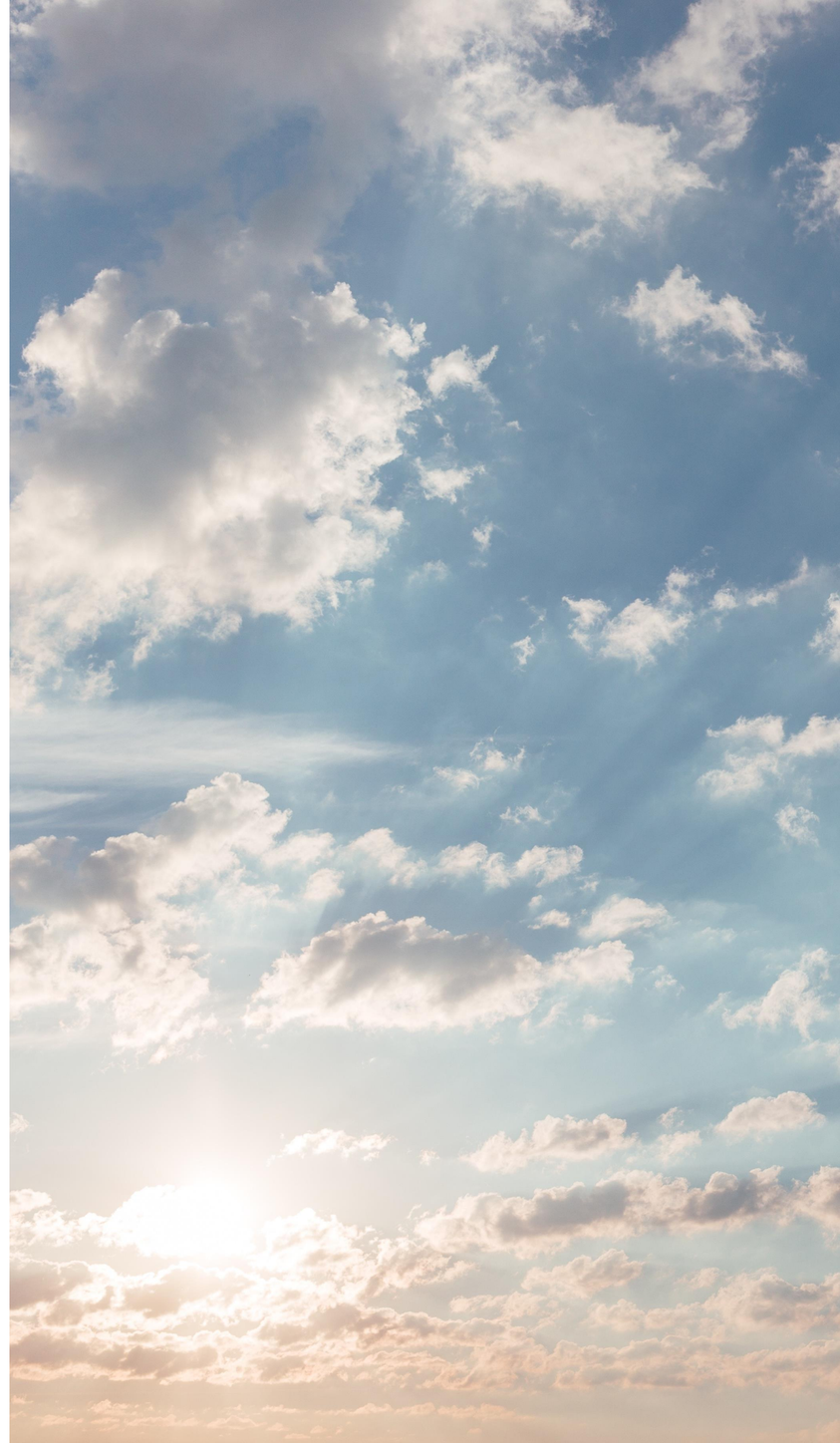
- Serverless, нет серверов нет проблем
- FaaS
- Автоматический maintenance
- Автоматическое управление ресурсами
- Микросервисная архитектура
- Динамический скейлинг, в пределах квот



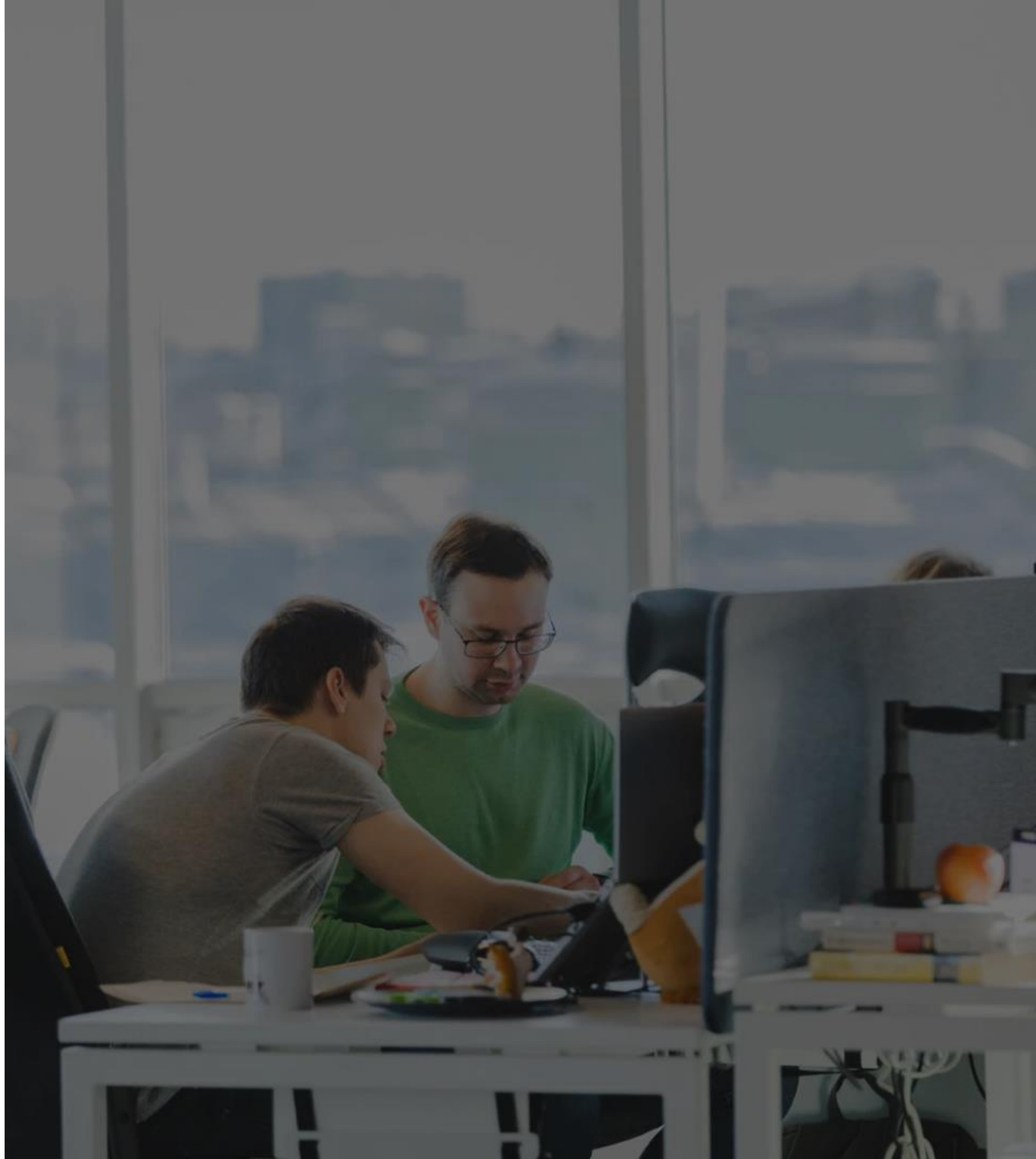
А может, ну эти серверы?

- Serverless, нет серверов нет проблем
- FaaS
- Автоматический maintenance
- Автоматическое управление ресурсами
- Микросервисная архитектура
- Динамический скейлинг, в пределах квот

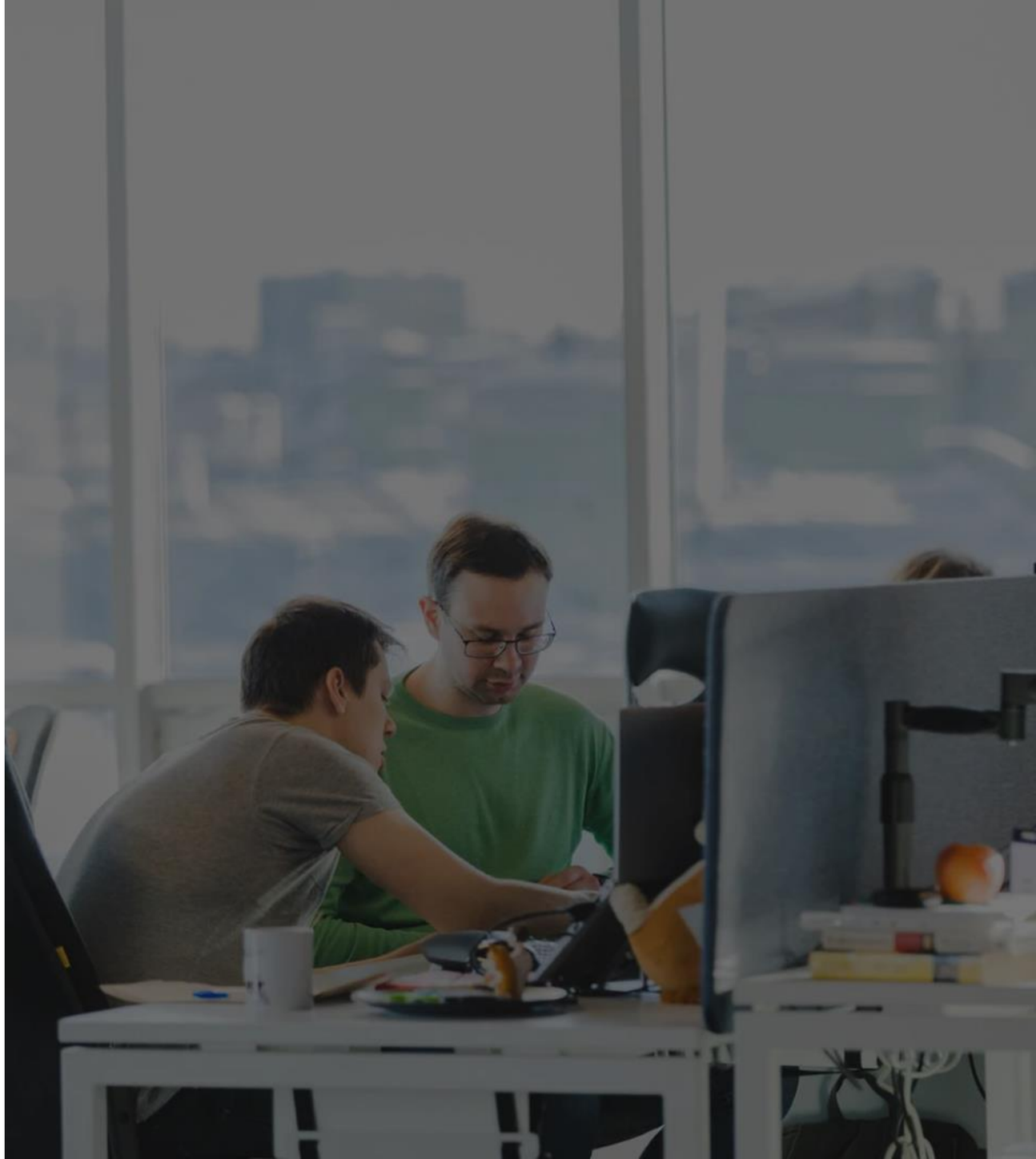
```
module.exports.handler = async function (event, context) {
    return {
        statusCode: 200,
        body: 'Hello World!',
    };
};
```



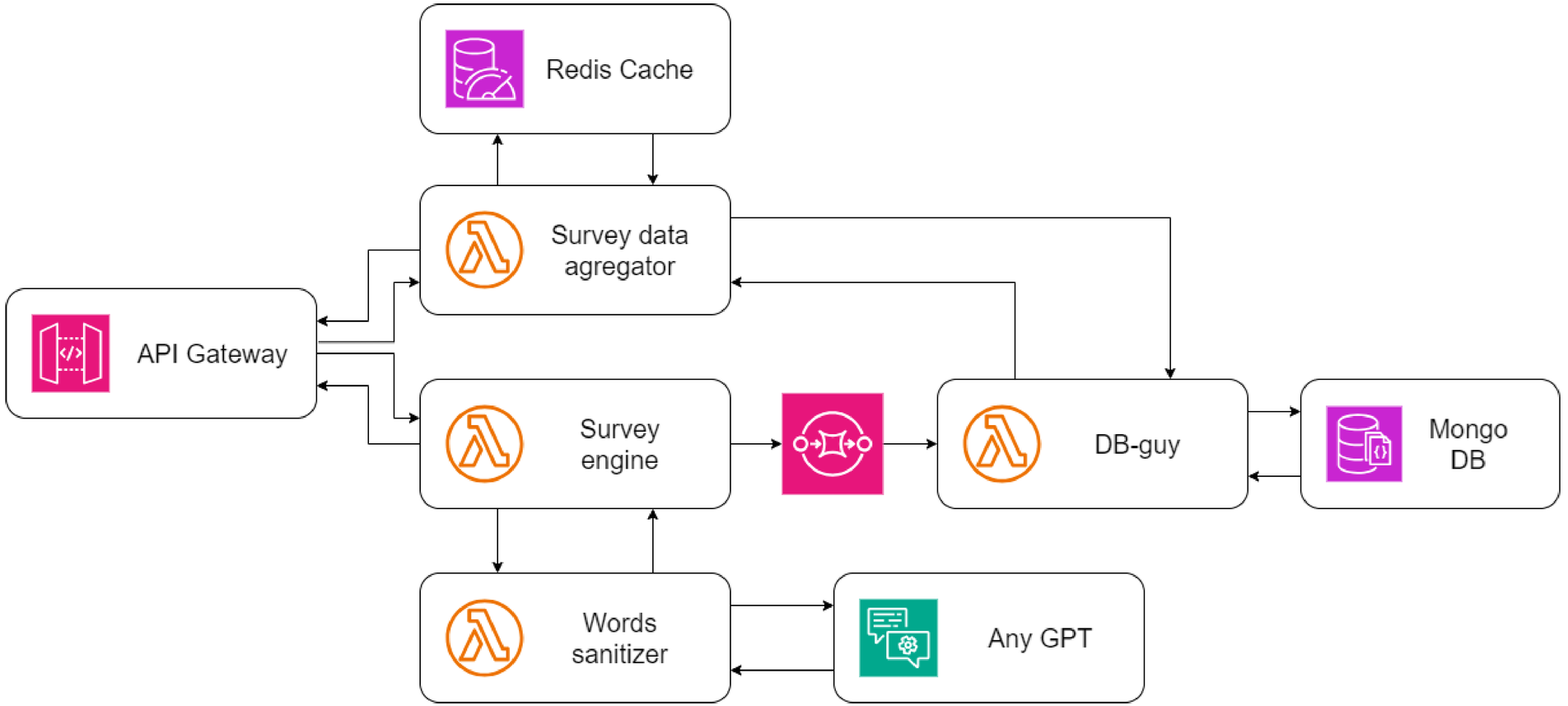
Время для интерактива



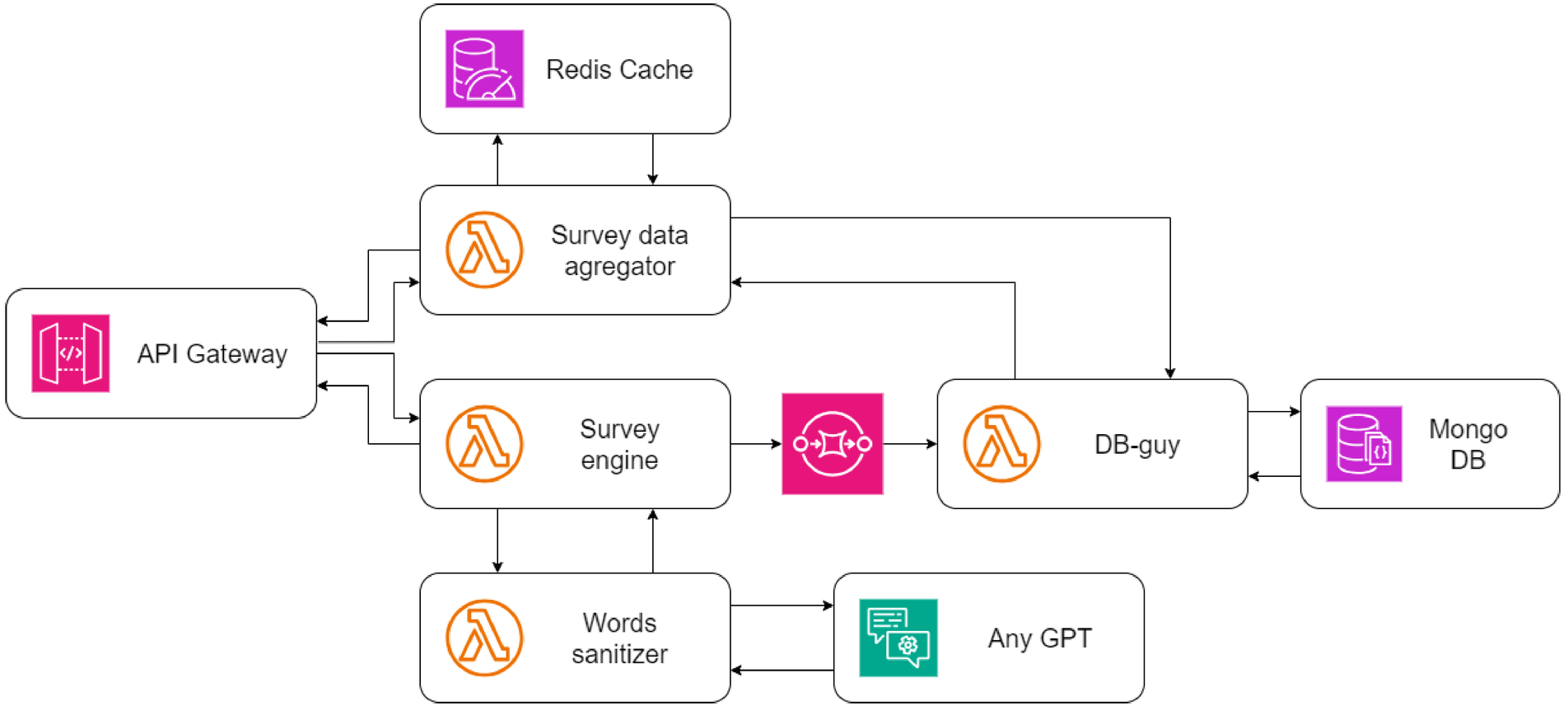
А что под капотом?



Архитектура

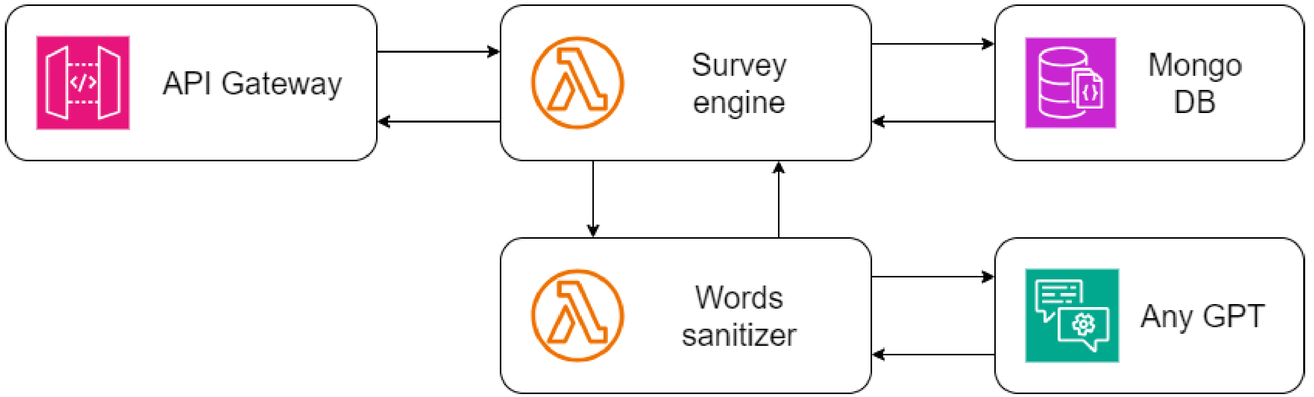


Архитектура



... за полчаса?..

Архитектура



За полчаса!

Архитектура

Gateway (шлюз)

- один из облачных сервисов
- инфраструктура недоступна извне
- осуществляет маршрутизацию

в пределах одной/нескольких функций

- можно прицепить домен

```
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://<your_gateway_id>.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: <your_function_id>
        tag: $latest
        service_account_id: <your_service_account_id>
components:
  schemas:
    JsonData:
      type: object
```

Архитектура

Gateway (шлюз)

- один из облачных сервисов
- инфраструктура недоступна извне
- осуществляет маршрутизацию

в пределах одной/нескольких функций

- можно прицепить домен

```
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://<your_gateway_id>.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: <your_function_id>
        tag: $latest
        service_account_id: <your_service_account_id>
components:
  schemas:
    JsonData:
      type: object
```

Архитектура

Gateway (шлюз)

- один из облачных сервисов
- инфраструктура недоступна извне
- осуществляет маршрутизацию

в пределах одной/нескольких функций

- можно прицепить домен

```
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://<your_gateway_id>.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: <your_function_id>
        tag: $latest
        service_account_id: <your_service_account_id>
components:
  schemas:
    JsonData:
      type: object
```

Архитектура

Gateway (шлюз)

- один из облачных сервисов
- инфраструктура недоступна извне
- осуществляет маршрутизацию

в пределах одной/нескольких функций

- можно прицепить домен

```
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://<your_gateway_id>.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: <your_function_id>
        tag: $latest
        service_account_id: <your_service_account_id>
components:
  schemas:
    JsonData:
      type: object
```

Архитектура

Gateway (шлюз)

- один из облачных сервисов
- инфраструктура недоступна извне
- осуществляет маршрутизацию

в пределах одной/нескольких функций

- можно прицепить домен

```
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://<your_gateway_id>.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: <your_function_id>
        tag: $latest
        service_account_id: <your_service_account_id>
components:
  schemas:
    JsonData:
      type: object
```

Архитектура

Gateway (шлюз)

```yaml
openapi: 3.0.0
info:
  title: HolySurvey API
  version: 1.0.0
servers:
- url: https://[illegible]pjf3c5.apigw.yandexcloud.net
paths:
  /api/{path+}:
    x-yc-apigateway-any-method:
      parameters:
        - name: path
          in: path
          required: false
          schema:
            type: string
      x-yc-apigateway-integration:
        type: cloud_functions
        function_id: d4e6i[illegible]66s911fj
        tag: $latest
        service_account_id: ajeo[illegible]8o42ce
  /:
    get:
      x-yc-apigateway-integration:
        type: dummy
        content:
```

Архитектура

Survey engine (функция)

Нам нужны:

- express
- serverless-http
- mongodb

Других зависимостей нет.

```
import { mongoUrl } from "./consts.js";
import { addMiddleware } from "./middleware.js";
import { countValues, countTravelValues } from './logic.js';
import { MongoClient } from 'mongodb';

import express from 'express';
import serverless from 'serverless-http';

const app = express();
addMiddleware(app);

const client = new MongoClient(mongoUrl);
```

Архитектура

Survey engine (функция)

```
app.post('/api/survey', async (req, res) => {
  try {
    validateRequest(req);
    await sanitizeValues(req);

    const conn = await client.connect();
    const db = conn.db("HolySurvey")
        .collection("results")
        .insertOne(req.body);

    res.status(200).json(result);
  } catch (err) {
    res.status(500).json(
      { message: 'Ошибка сервера', error: err }
    );
  }
})
```

Архитектура

Survey engine (функция)

1. Описываем эндпоинт

```
app.post('/api/survey', async (req, res) => {
  try {
    validateRequest(req);
    await sanitizeValues(req);

    const conn = await client.connect();
    const db = conn.db("HolySurvey")
        .collection("results")
        .insertOne(req.body);

    res.status(200).json(result);
  } catch (err) {
    res.status(500).json(
      { message: 'Ошибка сервера', error: err }
    );
  }
})
```

Архитектура

Survey engine (функция)

1. Описываем эндпоинт
2. Проверяем данные

```
app.post('/api/survey', async (req, res) => {
  try {
    validateRequest(req);
    await sanitizeValues(req);

    const conn = await client.connect();
    const db = conn.db("HolySurvey")
        .collection("results")
        .insertOne(req.body);

    res.status(200).json(result);
  } catch (err) {
    res.status(500).json(
      { message: 'Ошибка сервера', error: err }
    );
  }
})
```

Архитектура

Survey engine (функция)

1. Описываем эндпоинт
2. Проверяем данные
3. Развлекаемся с БД

```
app.post('/api/survey', async (req, res) => {
  try {
    validateRequest(req);
    await sanitizeValues(req);

    const conn = await client.connect();
    const db = conn.db("HolySurvey")
        .collection("results")
        .insertOne(req.body);

    res.status(200).json(result);
  } catch (err) {
    res.status(500).json(
      { message: 'Ошибка сервера', error: err }
    );
  }
})
```

Архитектура

Survey engine (функция)

1. Описываем эндпоинт
2. Проверяем данные
3. Развлекаемся с БД
4. Отдаём результат

```
app.post('/api/survey', async (req, res) => {
  try {
    validateRequest(req);
    await sanitizeValues(req);

    const conn = await client.connect();
    const db = conn.db("HolySurvey")
        .collection("results")
        .insertOne(req.body);

   res.status(200).json(result);
  } catch (err) {
    res.status(500).json(
      { message: 'Ошибка сервера', error: err }
    );
  }
})
```

Архитектура

Survey engine (функция)

И да, нужен небольшой враппер над app

```
export const handler = (event, context) => {
    const patchedEvent = {
        ...event,
        path: event.url,
        originalPath: event.path
    }
    return serverless(app)(patchedEvent, context);
}
```

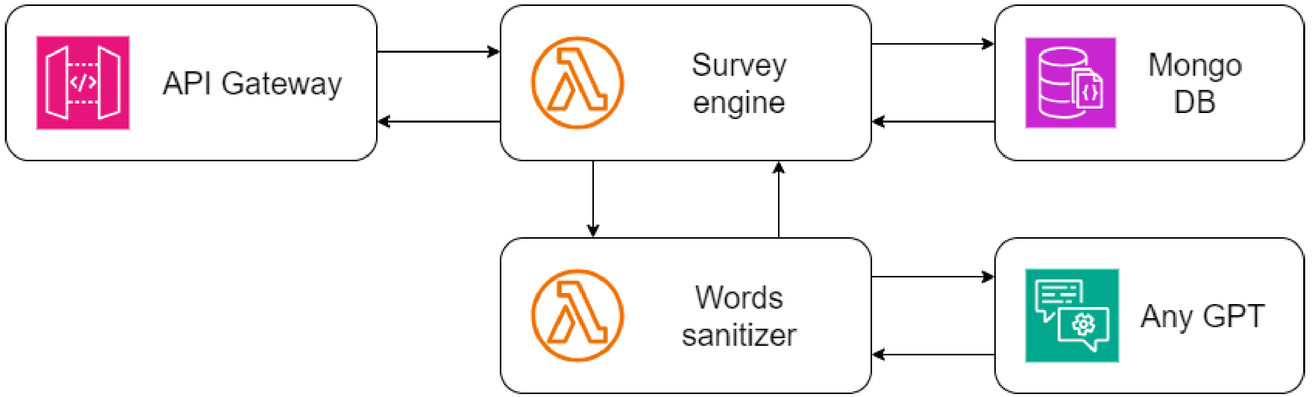

Архитектура

Words sanitizer (тоже функция)

1. Принимаем запрос
2. Направляем в чат
3. Верим в способности нейросетки =)

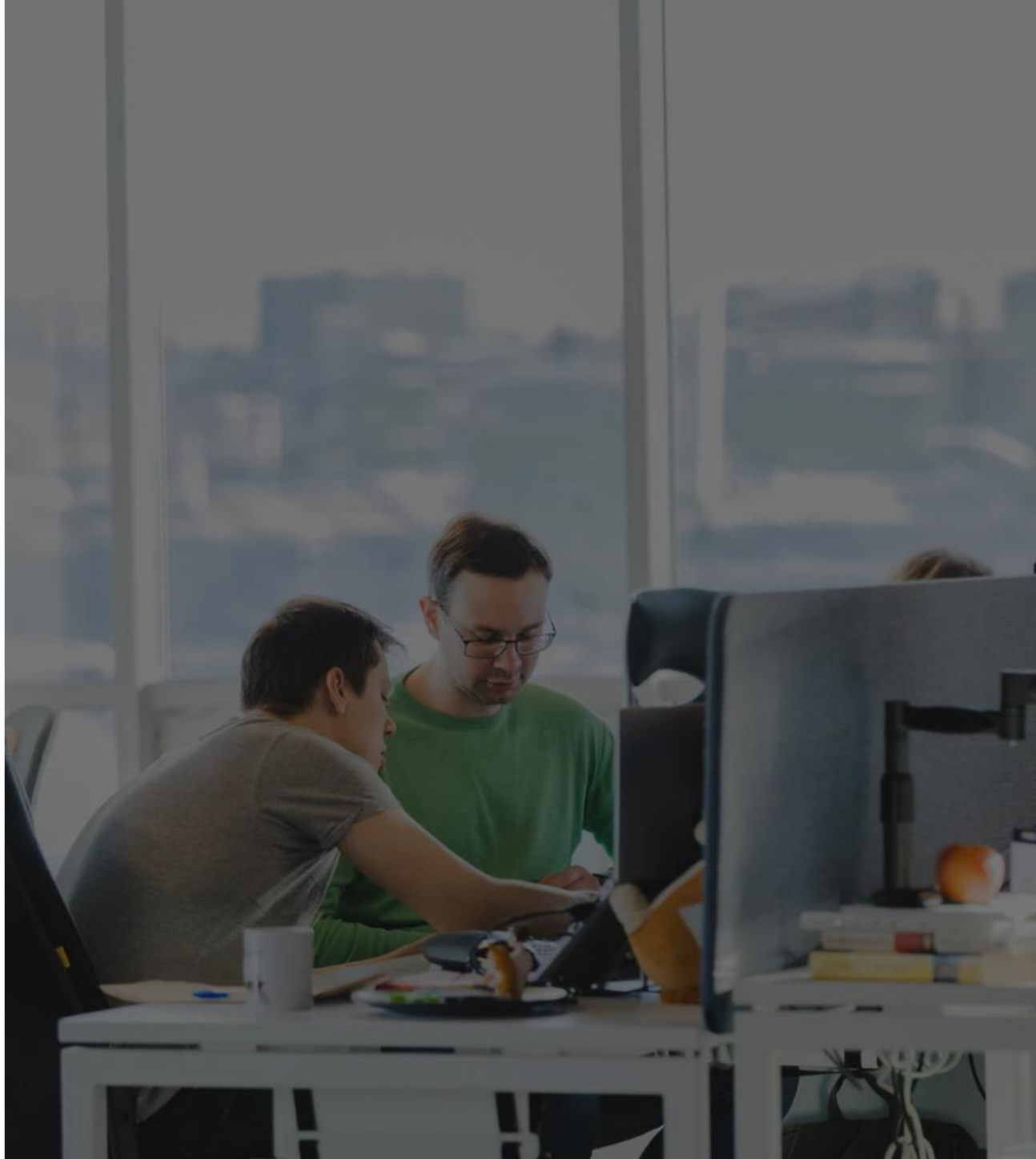
```
const magic = require('./magic.js');

module.exports.handler = async function (event) {
    const phrase = JSON.parse(event.body);

    const payload = {
        message: magic.getPrompt(phrase),
        api_key: magic.apiKey
    }

    const response = await fetch(
        magic.url,
        { method: 'POST', body: JSON.stringify(payload)}
    );

    const result = await response.json();

    return {
        statusCode: 200,
        body: result.response,
    };
};
```

Архитектура



За полчаса!

Что с деплоем?



Чьё облако использовать?

Если вы:

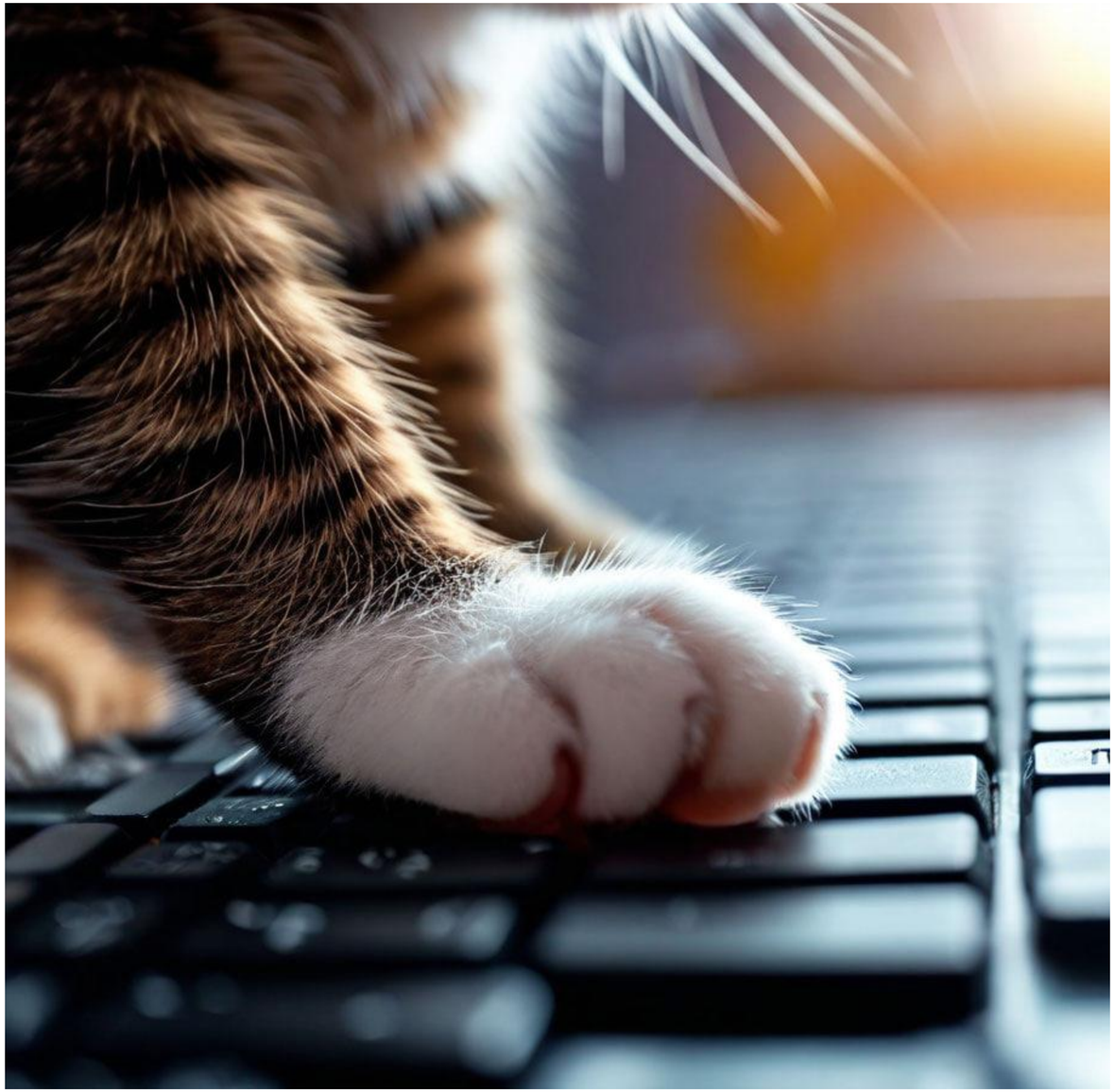
- не в РФ
- или есть карта
- и не боитесь санкций

Чьё облако использовать?

В РФ варианты тоже есть!

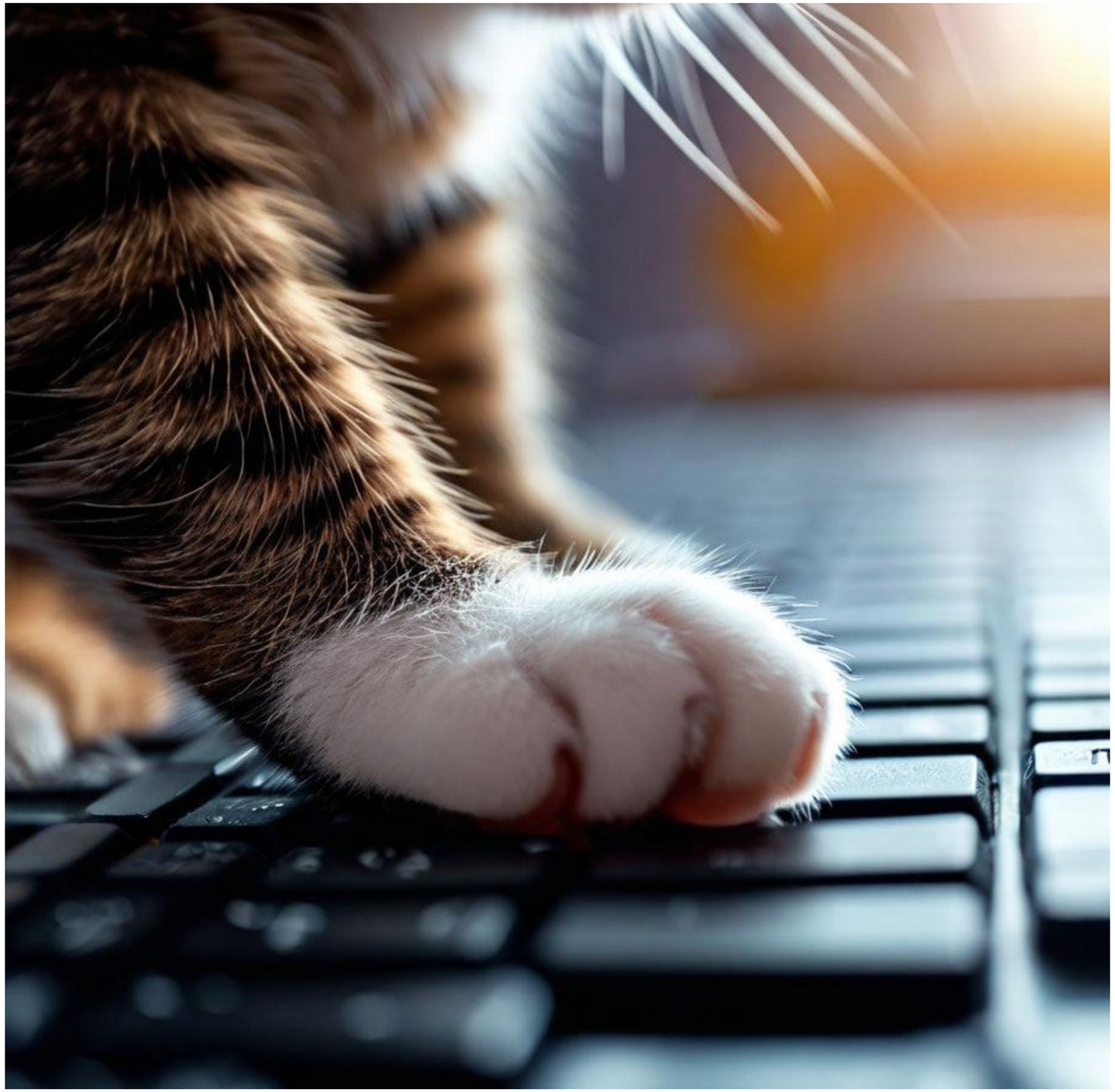
Что у нас с CI/CD?

- Ctrl+C, Ctrl+V



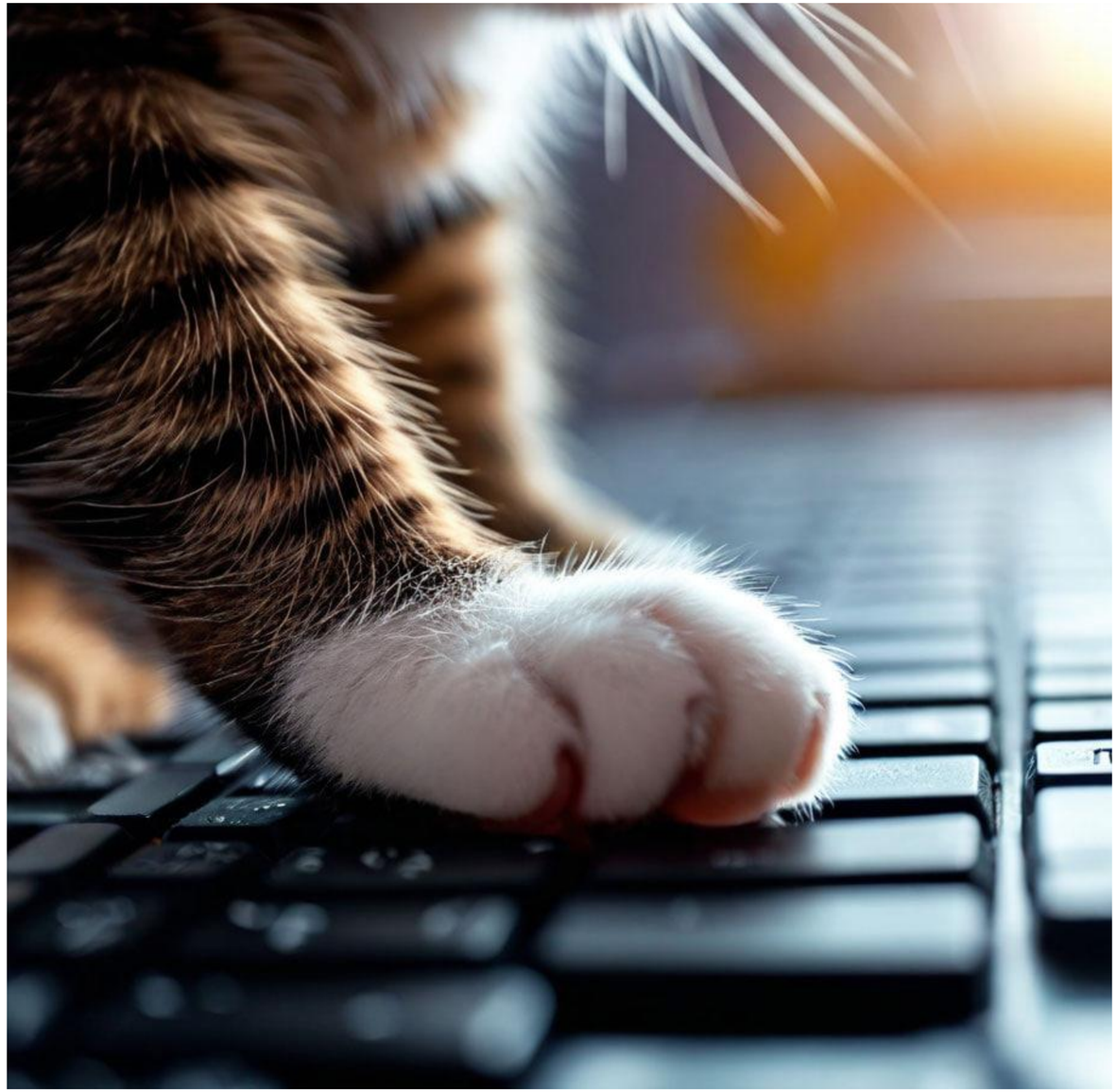
Что у нас с CI/CD?

- Ctrl+C, Ctrl+V
- Zip-архив



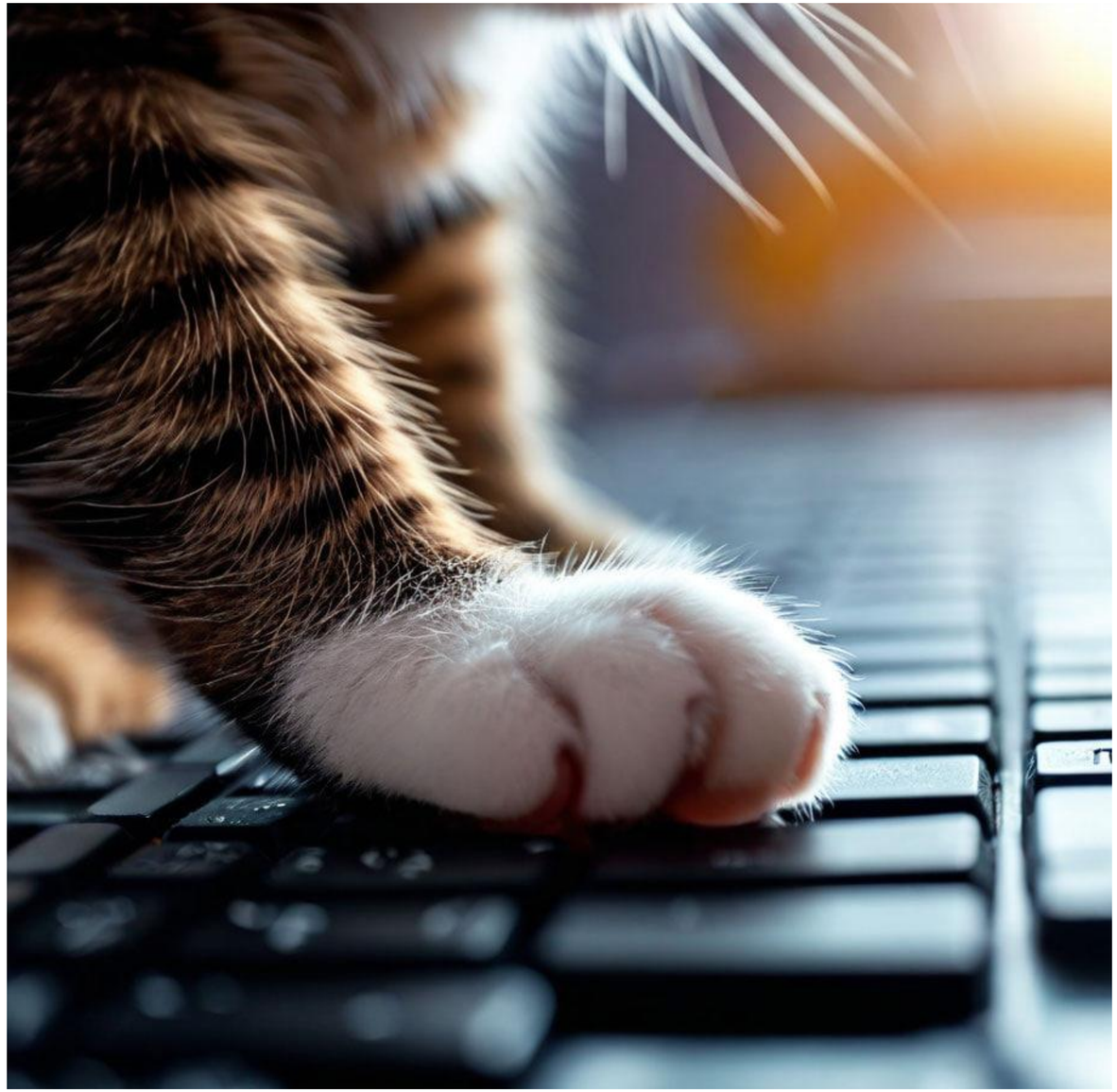
Что у нас с CI/CD?

- Ctrl+C, Ctrl+V
- Zip-архив
- GitHub workflows / GitLab pipelines

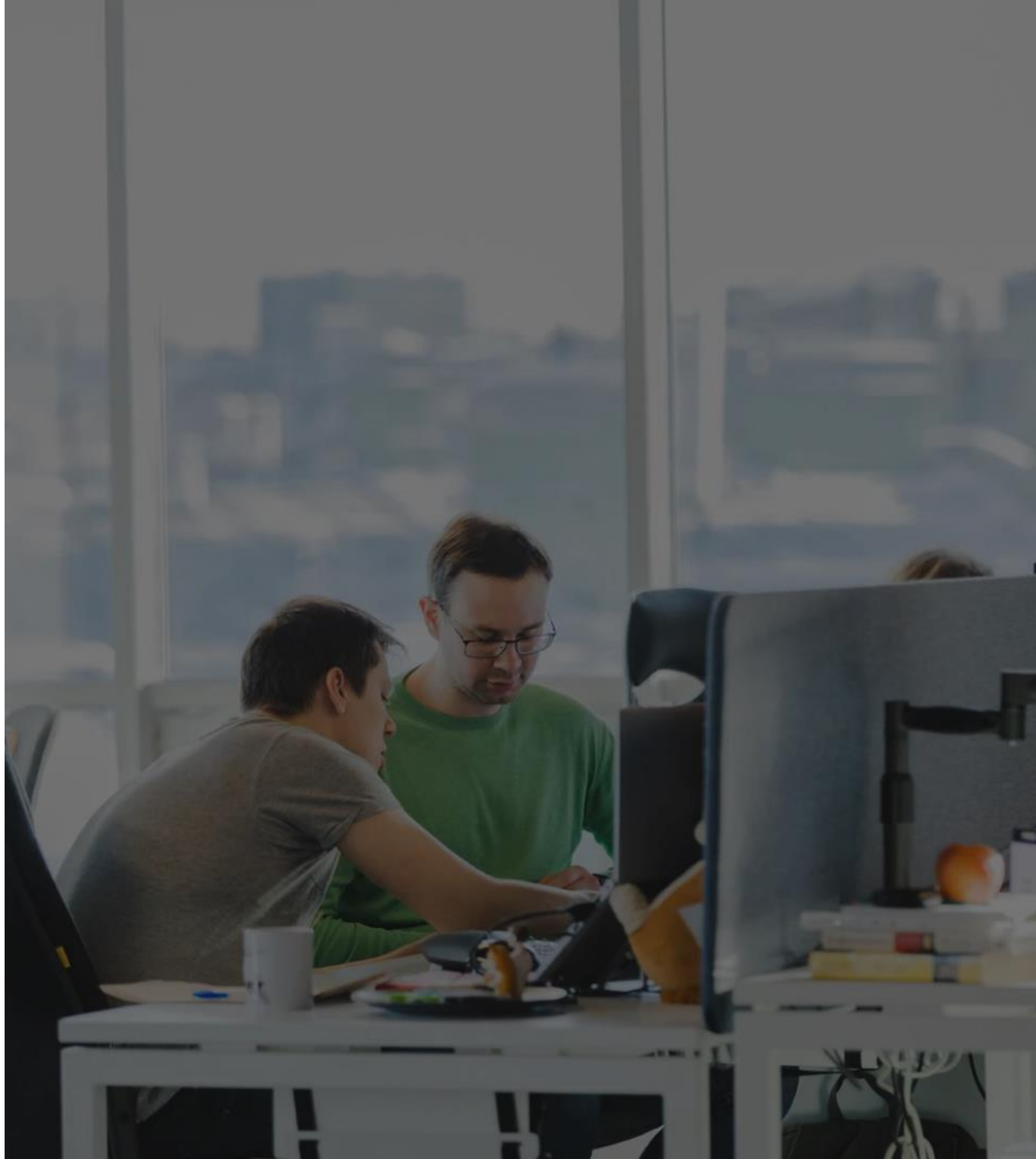


Что у нас с CI/CD?

- Ctrl+C, Ctrl+V
- Zip-архив
- GitHub workflows / GitLab pipelines
- Terraform



Немного о проде



Serverless в проде

Система документооборота

- Централизация отсылки, шаринга и обмена документами
- Инфраструктура AWS
- ~50 сервисов
- JS / Python

Serverless в проде

Система документооборота

- Централизация отсылки, шаринга и обмена документами
- Инфраструктура AWS
- ~50 сервисов
- JS / Python

Система управления IT-компанией

- Учёт времени, ведение проектов, бюджетирование
- Инфраструктура Yandex-cloud
- ~20 сервисов
- TS / JS

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом
- Быстрый rollback

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом
- Быстрый rollback
- Оптимизация расходов

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом
- Быстрый rollback
- Оптимизация расходов
- Редактирование кода в браузере

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом
- Быстрый rollback
- Оптимизация расходов
- Редактирование кода в браузере
- Динамическое масштабирование

Что мы получили

(и нам понравилось)

- Упростилась поддержка инфраструктуры
- Логирование осуществлялось самим клаудом
- Быстрый rollback
- Оптимизация расходов
- Редактирование кода в браузере
- Динамическое масштабирование
- Все преимущества микросервисной архитектуры

Что мы получили

(и нам не понравилось)

- Сложнее дебажить

Что мы получили

(и нам не понравилось)

- Сложнее дебажить
- Редактирование кода в браузере

Что мы получили

(и нам не понравилось)

- Сложнее дебажить
- Редактирование кода в браузере
- Сложнее создать dev-окружение

Что мы получили

(и нам не понравилось)

- Сложнее дебажить
- Редактирование кода в браузере
- Сложнее создать dev-окружение
- Нужно думать о прогреве функций

Что мы получили

(и нам не понравилось)

- Сложнее дебажить
- Редактирование кода в браузере
- Сложнее создать dev-окружение
- Нужно думать о прогреве функций
- Можно упереться в квоты

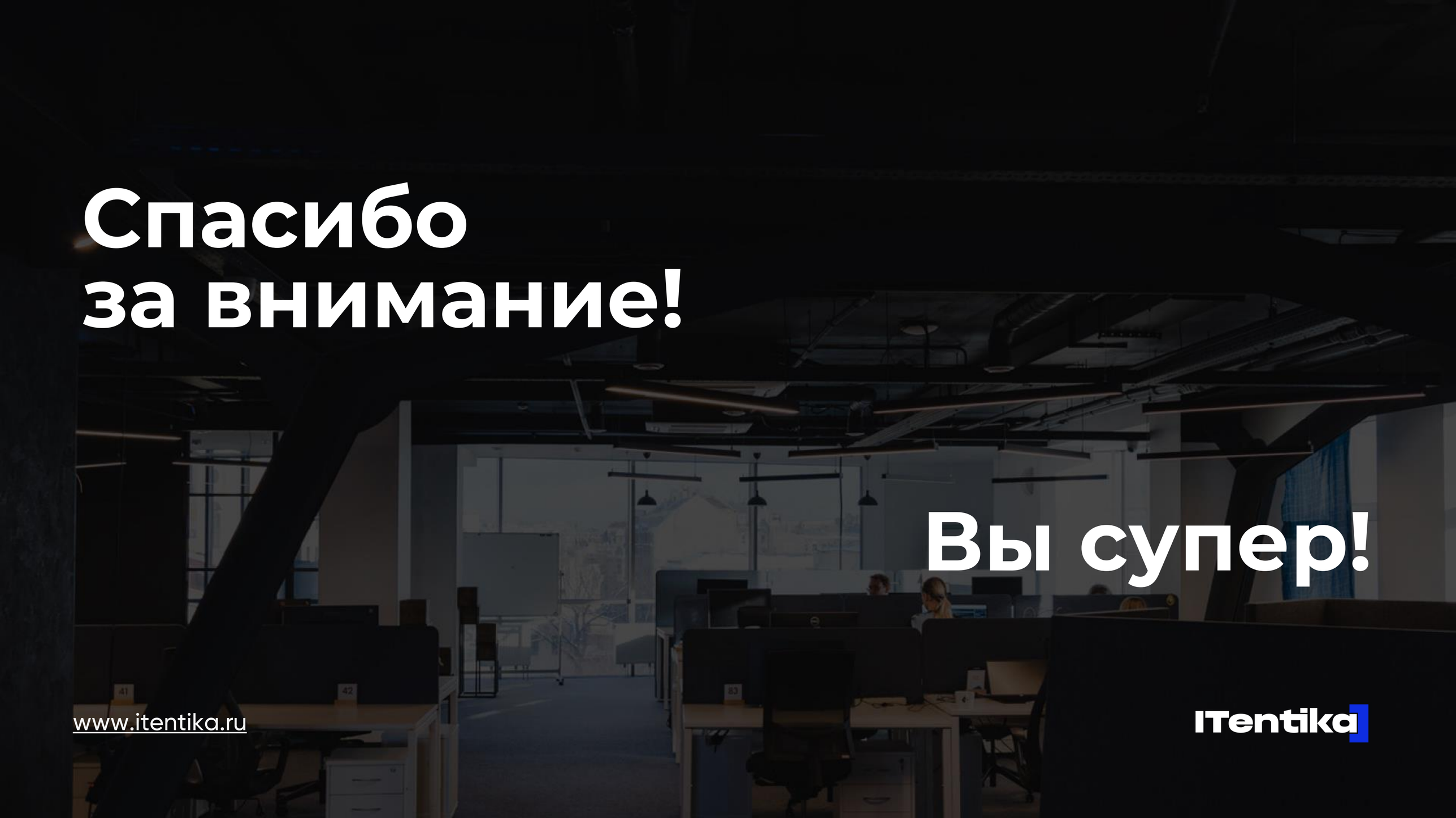
Что мы получили

(и нам не понравилось)

- Сложнее дебажить
- Редактирование кода в браузере
- Сложнее создать dev-окружение
- Нужно думать о прогреве функций
- Можно упереться в квоты
- Все недостатки микросервисной архитектуры

Для кого serverless

- Низкая нагрузка на сервис
- Сильно неоднородный характер нагрузки на сервис
- Внутренние сервисы компаний
- Стартапы (когда каждая копейка на счету)
- Фоновая обработка и задачи по расписанию
- IoT: смарт-термометр, автоматизация дома и др.



Спасибо за внимание!

Вы супер!

www.itentika.ru

ITentika