Путь в тысячу микросервисов начинается с одного

Smartway | Входит в группу компаний 1С

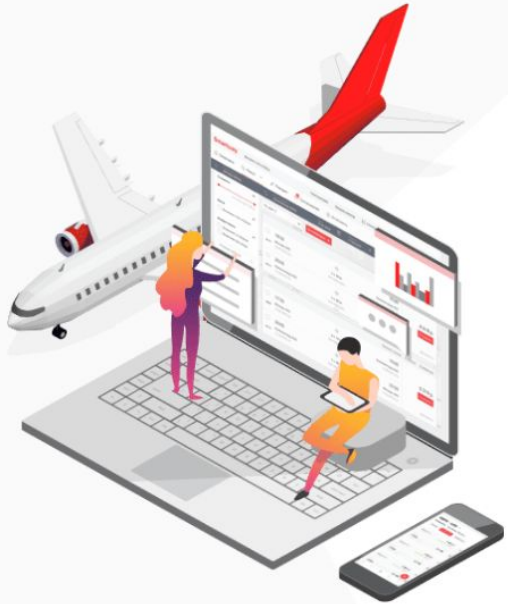
Для кого | Возможности | О компании | Контакты

Войти | Получить доступ

Удобные командировки без переплат

Бронируйте билеты и отели для сотрудников без наценок. Все документы сформируются автоматически.

Получить доступ



Без сервисных сборов

Мы работаем напрямую с глобальными поставщиками и продаём билеты без комиссии

Интеграция с 1С:

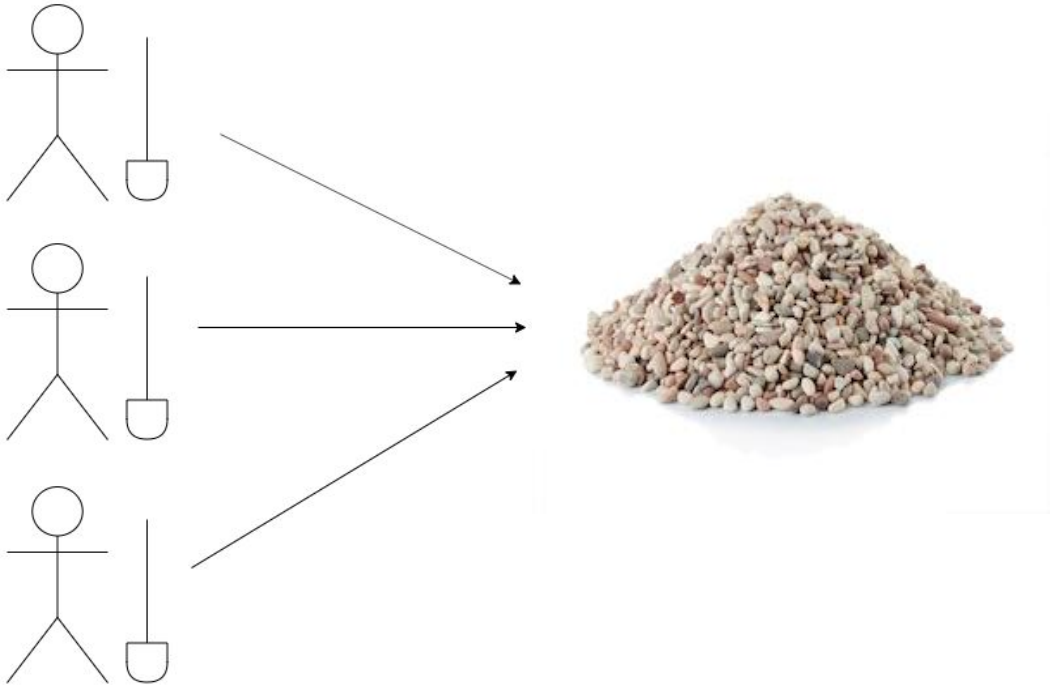
Данные о поездках больше не нужно вручную загружать в «1С» Они попадут туда автоматически

Поддержка 24/7

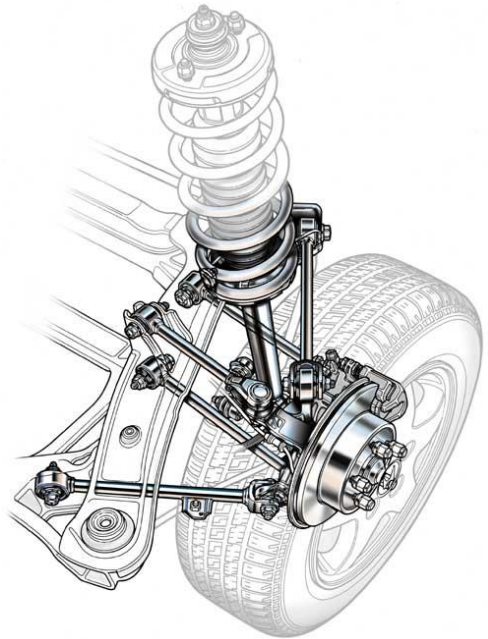
Ответим на любые вопросы, поможем обменять и вернуть билеты или изменить бронь

1. С чего все началось и как мы к этому пришли
2. Оркестрация
    2.1. Docker swarm
    2.2. Маршрутизация
    2.3. CI и environment
3. Окружения и БД
    3.1. DNS и стенды
    3.2. Namespace
    3.3. Миграции БД и драйверы
4. Логирование, мониторинг

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг



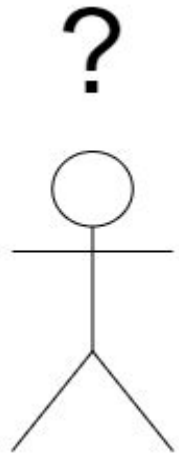
1. С чего все началось и как мы к этому пришли
2. Оркестрация
    2.1. Docker swarm
    2.2. Маршрутизация
    2.3. CI и environment
3. Окружения и БД
    3.1. DNS и стенды
    3.2. Namespace
    3.3. Миграции БД и драйверы
4. Логирование, мониторинг

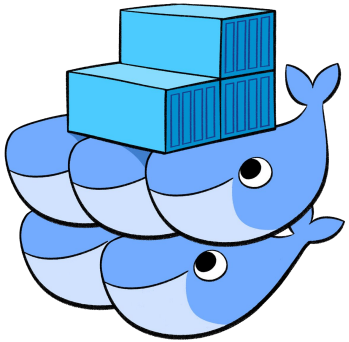


1. С чего все началось и как мы к этому пришли
2. Оркестрация
    2.1. Docker swarm
    2.2. Маршрутизация
    2.3. CI и environment
3. Окружения и БД
    3.1. DNS и стенды
    3.2. Namespace
    3.3. Миграции БД и драйверы
4. Логирование, мониторинг

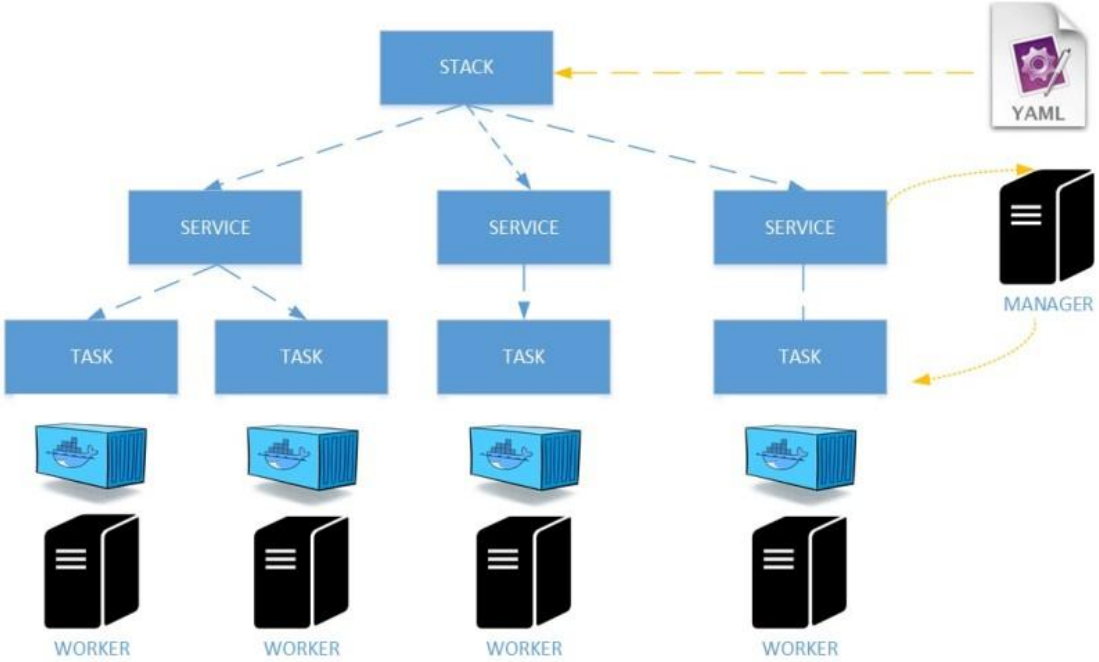
Начни такой проект сейчас - я бы сразу проектировал микросервисную архитектуру

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг





1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

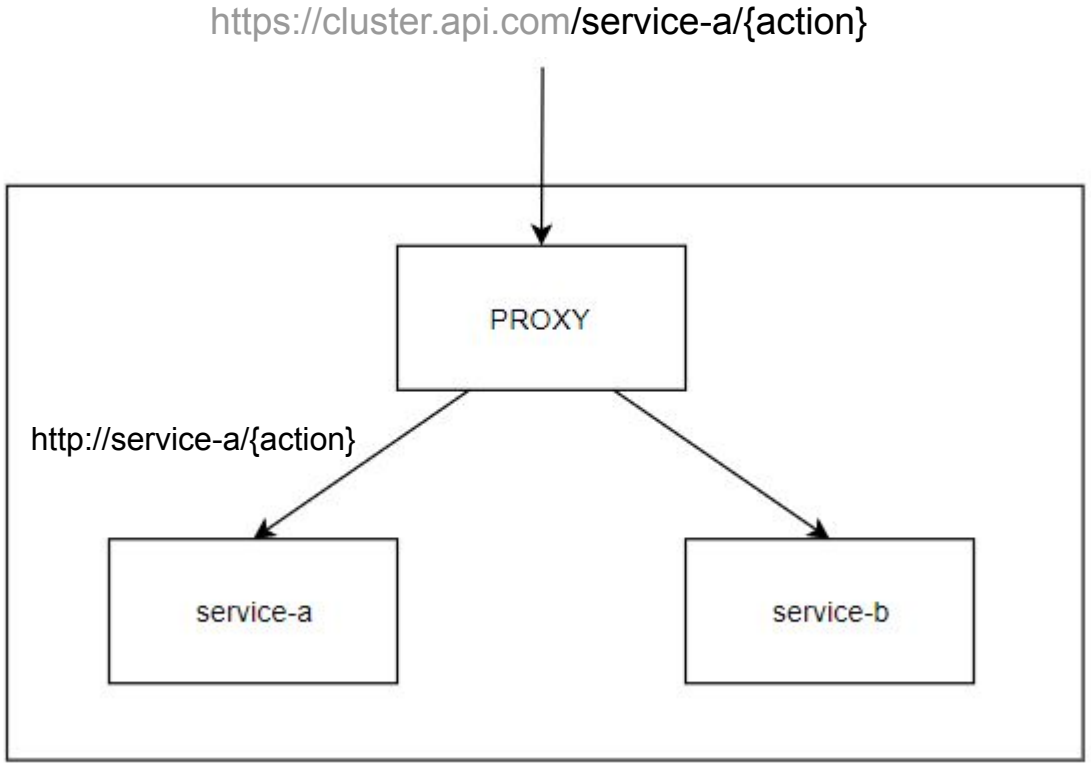


1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

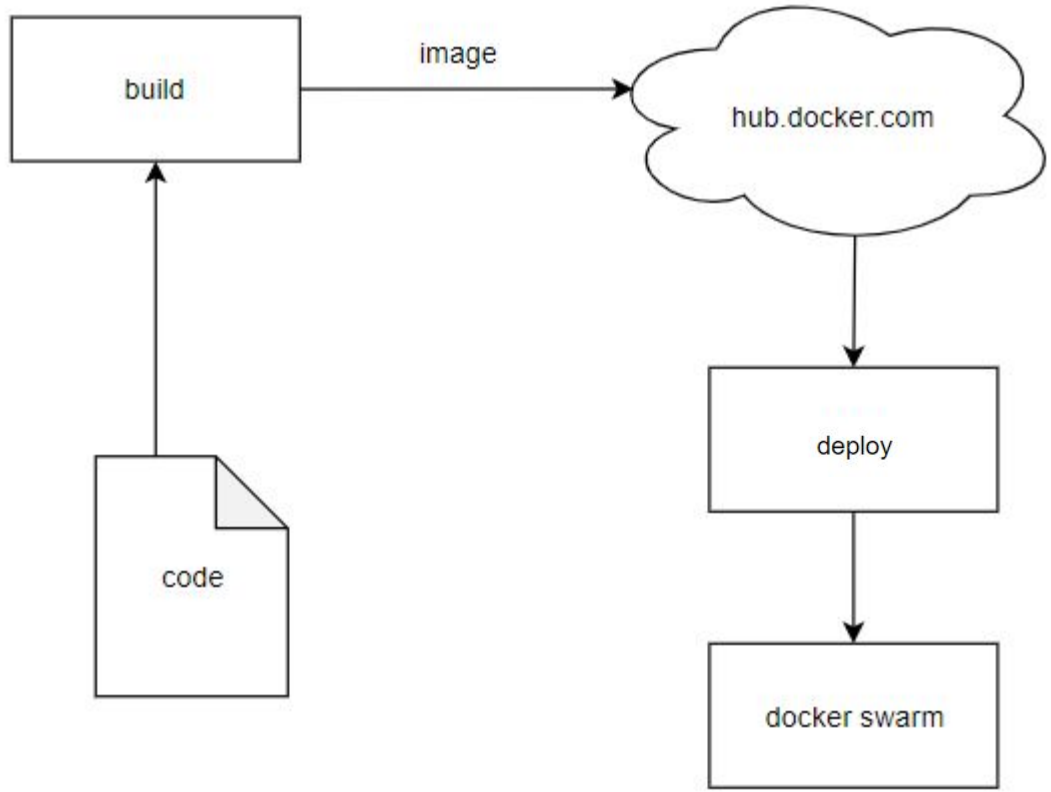
```
version: "3.3"
networks:
  external:
  internal:
  auth:
services:
  proxy:
    image: smartwaytoday/proxy:latest.!BRANCH!
    deploy:
      replicas: !REP!
      resources:
        limits:
          memory: 300M
        reservations:
          memory: 20M
    networks:
      - external
    ports:
      - "8001:8001"
  auth-gateway:
    image: smartwaytoday/auth-gateway:latest.!BRANCH!
    deploy:
      replicas: !REP!
      resources:
        limits:
          memory: 300M
        reservations:
          memory: 20M
    networks:
      - external
      - auth
    environment:
      - SW_ENV=!ENV!
```

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   2.1. Docker swarm
   2.2. Маршрутизация
   2.3. CI и environment
3. Окружения и БД
   3.1. DNS и стенды
   3.2. Namespace
   3.3. Миграции БД и драйверы
4. Логирование, мониторинг

docker swarm имеет балансировщик транспортного уровня



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

```
version: "3.3"
networks:
  external:
  internal:
  auth:
services:
  proxy:
    image: smartwaytoday/proxy:latest.!BRANCH!
    deploy:
      replicas: !REP!
      resources:
        limits:
          memory: 300M
        reservations:
          memory: 20M
    networks:
      - external
    ports:
      - "8001:8001"
  auth-gateway:
    image: smartwaytoday/auth-gateway:latest.!BRANCH!
    deploy:
      replicas: !REP!
      resources:
        limits:
          memory: 300M
        reservations:
          memory: 20M
    networks:
      - external
      - auth
    environment:
      - SW_ENV=!ENV!
```

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

```
$ docker stack deploy --compose-file docker-compose.yml vossibility
```

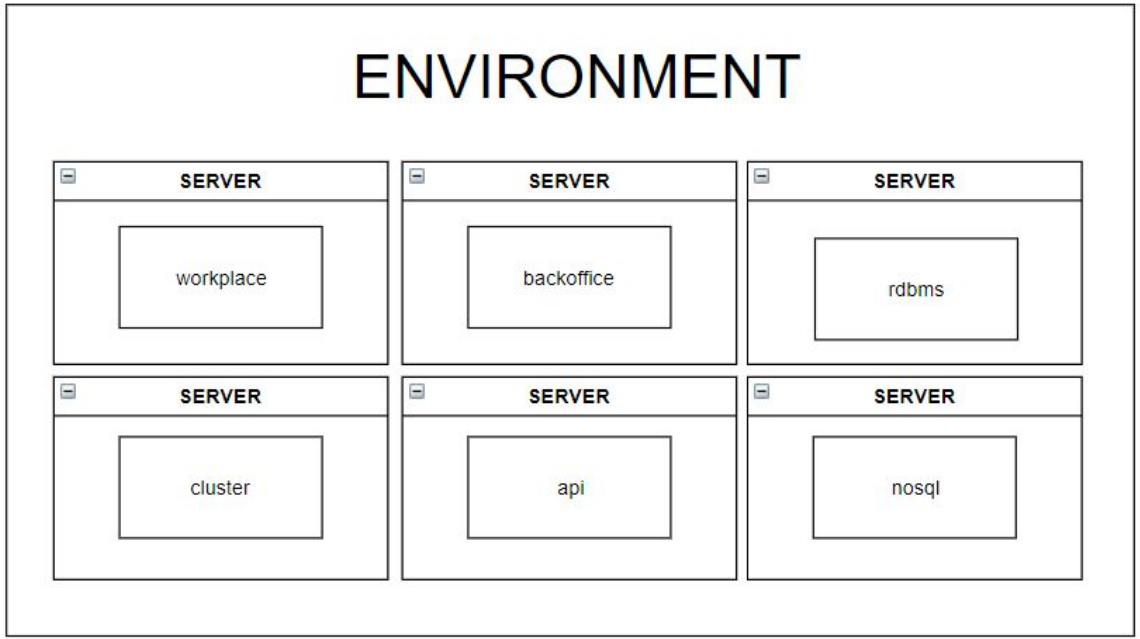
```
docker service create \
  --replicas ${COUNT} \
  --hostname ${SERVICE_NAME} \
  --env SW_ENV=${ENV}
```

```
docker service update \
  ${Rollback_} \
  ${SERVICE_NAME}
```

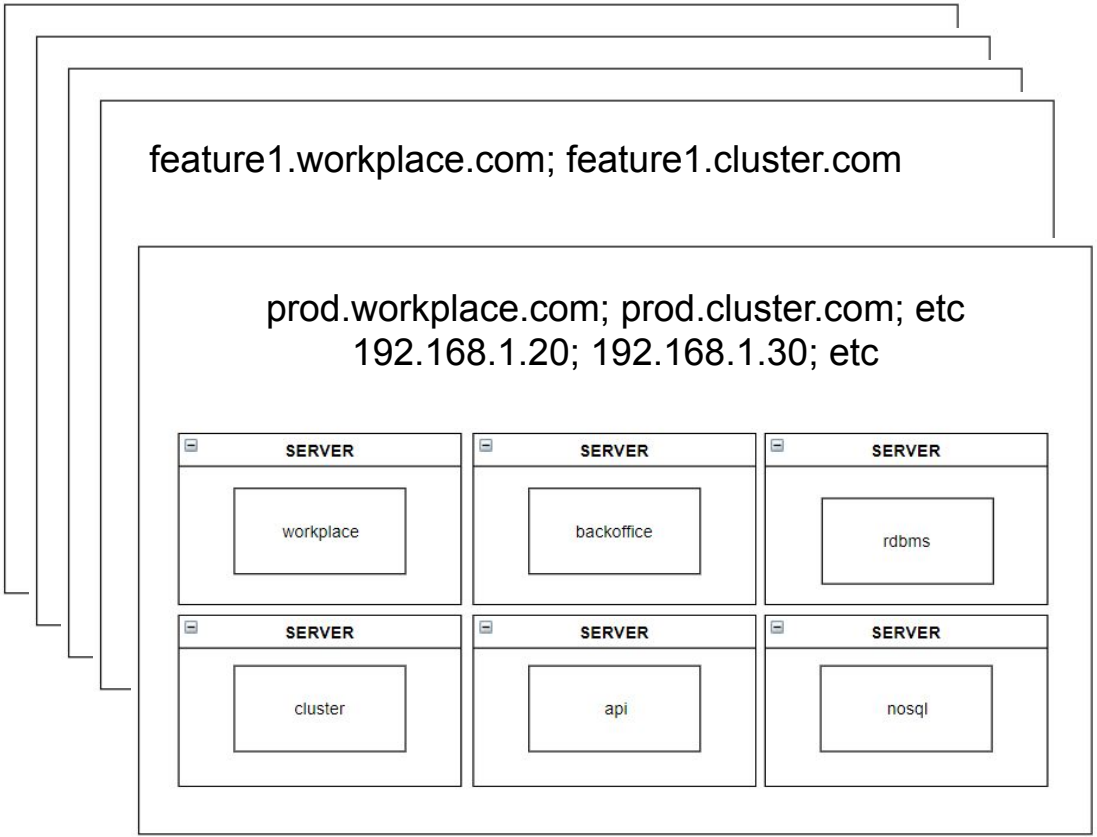
1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

Выбор средства оркестрации был обусловлен стоимостью его дальнейшей поддержки

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

А жирно не будет столько серверов держать?!

Дано: 100 микросервисов, 10 окружений и best practice в виде одной БД на каждый микросервис (да, такой паттерн имеет место быть)

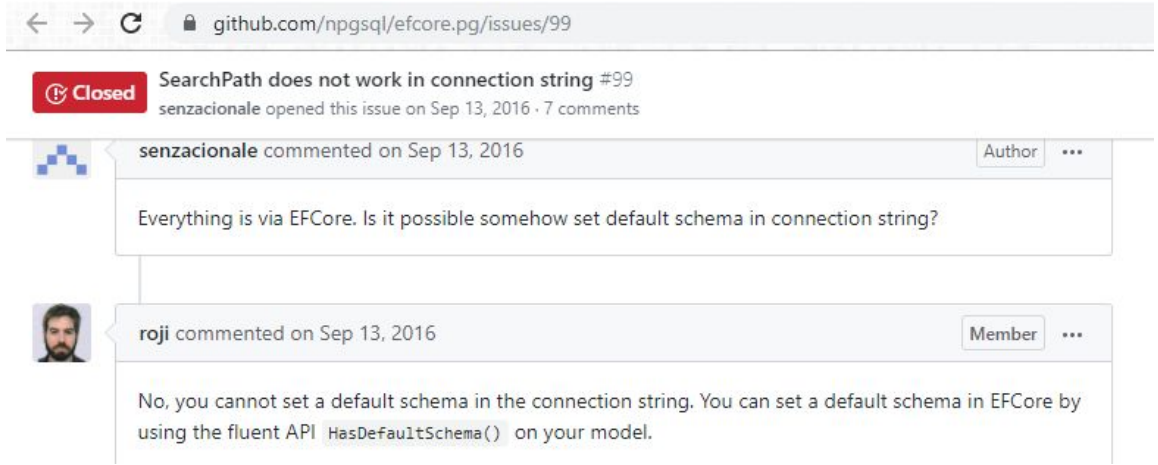
Итого: 1000 БД.

prod-users.public.user;
feature1-users.public.user;
feature2-users.public.user;
…….

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

users.prod.user;
users.feature1.user;
users.feature2.user;
…….

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

```
4 usages  murzaev
public static string GetEnv()
{
    return Environment.GetEnvironmentVariable("SW_ENV") ?? "local";
}
```

```
public partial class AddHistory : Migration
{
    private readonly string _schema = Common.Common.GetEnv();

    murzaev
    protected override void Up( [NotNull] MigrationBuilder migrationBuilder)
    {
        migrationBuilder.AddColumn<DateTime>(
            name: "add_date",
            schema: _schema,
            table: "codes",
            nullable: false,
            defaultValue: DateTime.UtcNow);

        migrationBuilder.CreateTable(
            name: "codes_history",
            schema: _schema,
            columns: table => new
            {...},
            constraints: table =>
            {...});

        migrationBuilder.CreateIndex(
            name: "IX_codes_history_code_id",
            schema: _schema,
            table: "codes_history",
            column: "code_id");
    }

    murzaev
    protected override void Down( [NotNull] MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropTable(
            name: "codes_history",
            schema: _schema);

        migrationBuilder.DropColumn(
            name: "add_date",
            schema: _schema,
            table: "codes");
    }
}
```

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

Учитывайте потребности в производственных окружениях и окружениях разработки

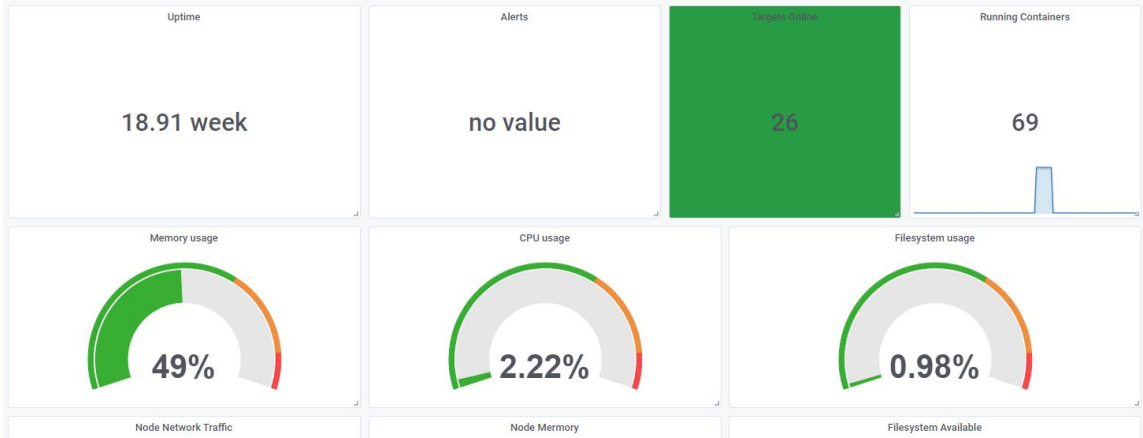
1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

¯_(ツ)_/¯

Console.WriteLine(“wtf?!”);

1. С чего все началось и как мы к этому пришли
2. Оркестрация
   - 2.1. Docker swarm
   - 2.2. Маршрутизация
   - 2.3. CI и environment
3. Окружения и БД
   - 3.1. DNS и стенды
   - 3.2. Namespace
   - 3.3. Миграции БД и драйверы
4. Логирование, мониторинг

grafana + prometheus = profit



1. С чего все началось и как мы к этому пришли
2. Оркестрация
   2.1. Docker swarm
   2.2. Маршрутизация
   2.3. CI и environment
3. Окружения и БД
   3.1. DNS и стенды
   3.2. Namespace
   3.3. Миграции БД и драйверы
4. Логирование, мониторинг

1. Логи и мониторинг лишними не бывают.
2. Бывает куча проблем, если вы не знаете и/или не понимаете что происходит в сервисе

1. Монолит - это не приговор. Необходимо думать о масштабировании.
2. Средства масштабирования необходимо выбирать по затратам на обслуживание и удобству использования
3. Используйте все средства, предлагаемые платформой/технологией для грамотного проектирования