

# ЭПОПЕЯ МИГРАЦИИ ~400 ПРОЕКТОВ ИЗ SWARM В K8S: ЧЕМУ ОНА НАУЧИЛА

Андрей Важенин

 skyeng

[github.com/ag0n1k](https://github.com/ag0n1k)



# КТО Я?

Инженер в инфраструктуре

Разработчик

Администратор

Devops-евангелист

SRE



# ПОЧЕМУ K8S?







скорость

мобильность

стабильность

безопасность

swarm умирает  
swarm плохо работает с большими кластерами



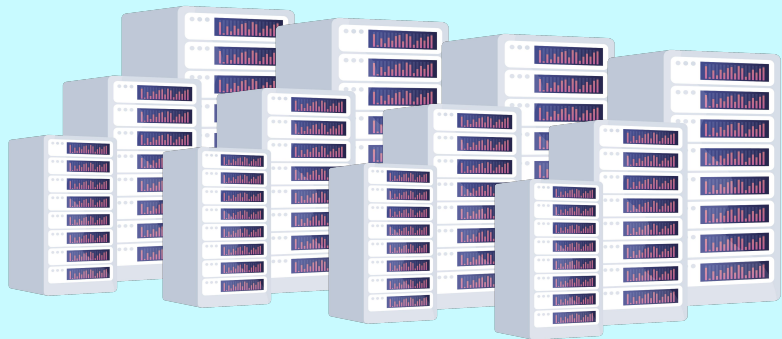
swarm умирает  
swarm плохо работает с большими кластерами



**This repository has been archived by the owner. It is now read-only.**



# ЭКОНОМИЯ



**Docker Swarm**  
40 nodes

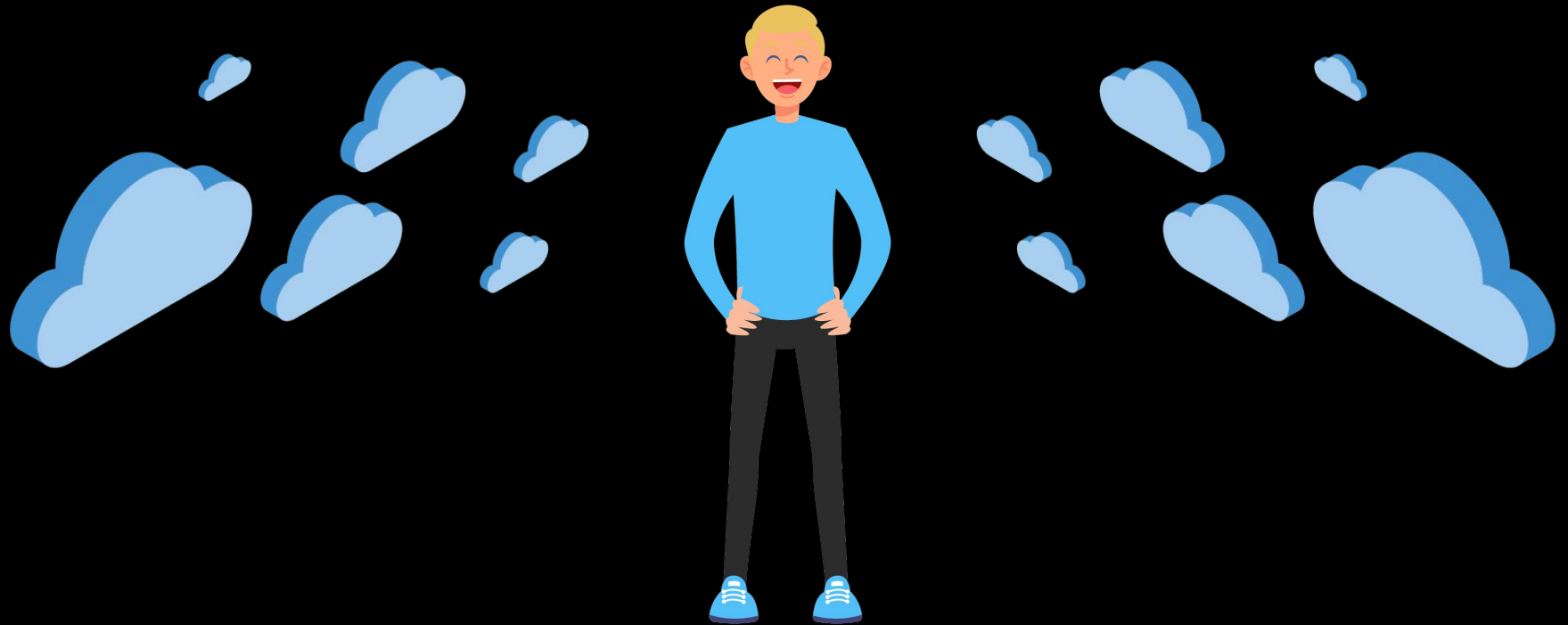


**k8s**  
25 nodes

лучше балансирует нагрузку  
утилизация ресурсов выше 60% → 80%  
**~40% выигрыш**



# ТРЕНДЫ



● **Kubernetes**  
Search term

● **Docker Swarm**  
Search term

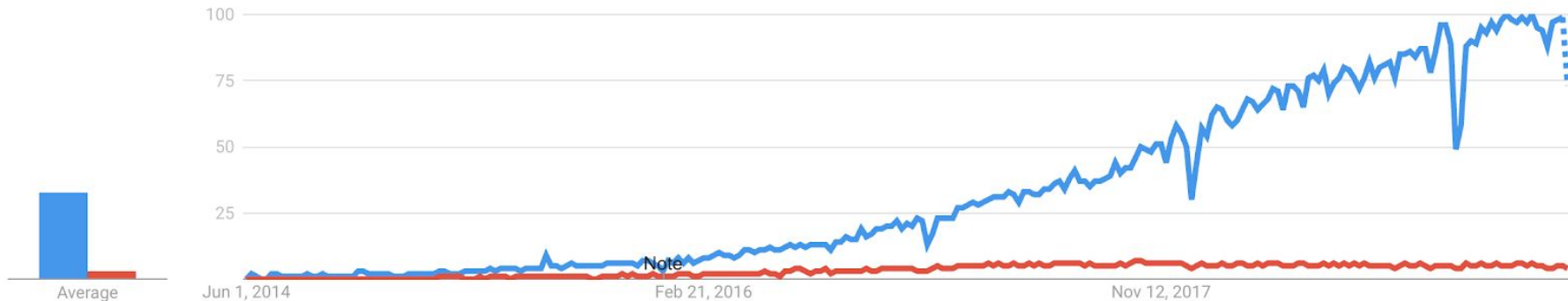
Worldwide ▼

Past 5 years ▼

All categories ▼

Web Search ▼

### Interest over time



● **kubernetes**  
Поисковый запрос

● **Swarm**  
Поисковый запрос

+ [Добавить сравнение](#)

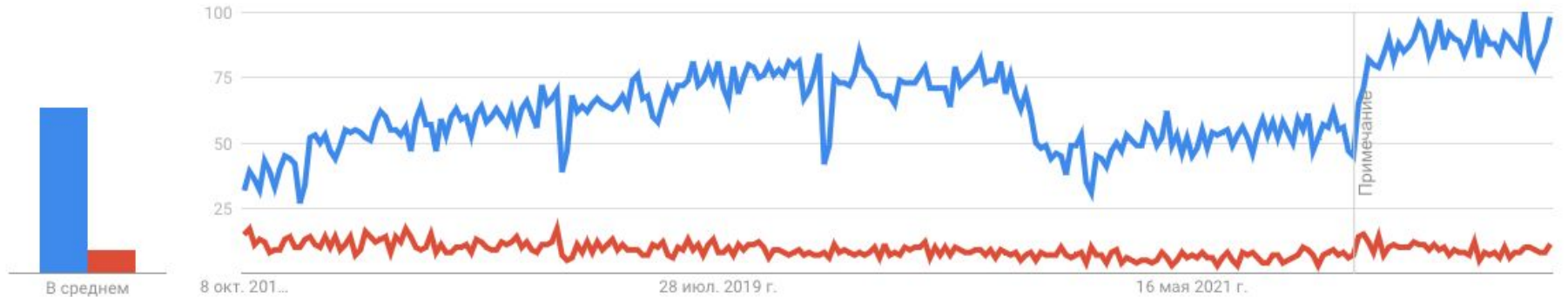
По всему миру ▾

Последние 5 лет ▾

Программное обеспечение ▾

Веб-поиск ▾

Динамика популярности ?



● **kubernetes**  
Поисковый запрос

● **Swarm**  
Поисковый запрос

+ **Добавить сравнение**

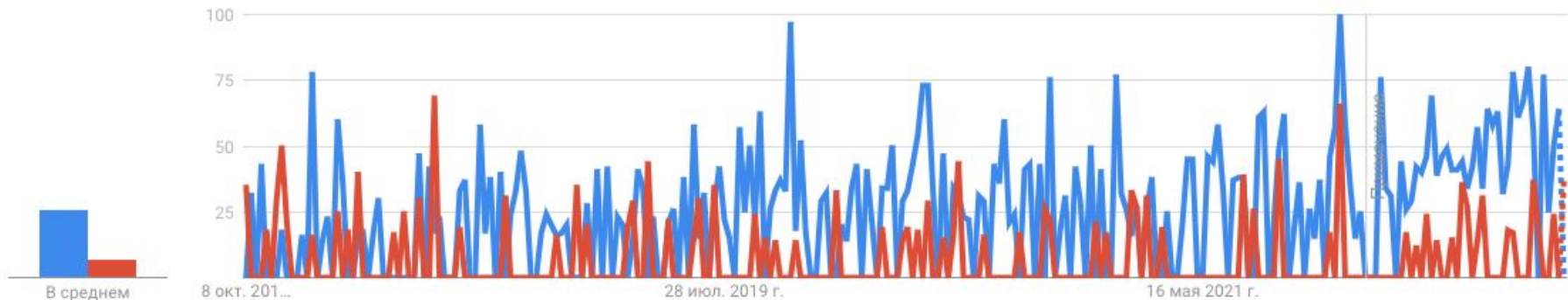
Россия ▾

Последние 5 лет ▾

Программное обеспечение ▾

Веб-поиск ▾

Динамика популярности ?





k8s



скорость

стабильность

**декларативность**

ограничения

**гибкость**

удобство разработки

удобство сопровождения

**docker  
swarm**



**k8s**



**1 год**

**3 года**



*Наш путь: не было экспертизы в кубах*

специфика



*skydeploy*

единое средство деплоя

*insight*

cmdb – единая система хранения описания инфры

единый стек

skyeng = php (не везде)

*ansible*

централизованный

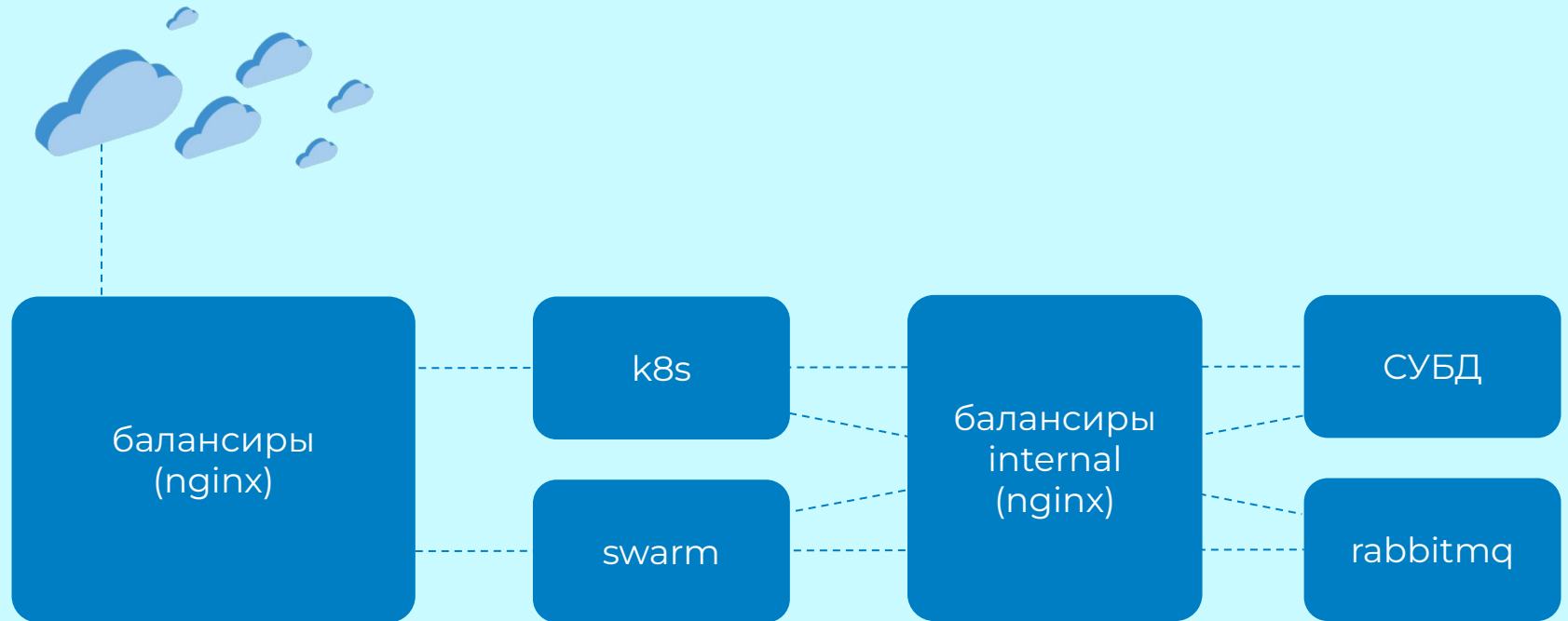
*jenkins*

централизованный

очереди и субд

живут отдельно  
от кластеров обработки

специфика  skyeng





# ПОДГОТОВКА К ПРОЕКТУ





dev & qa



dev ops



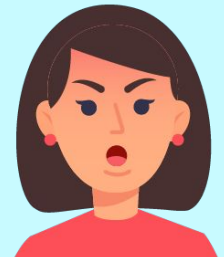
ops

# куберизация



нет ресурсов

бизнес

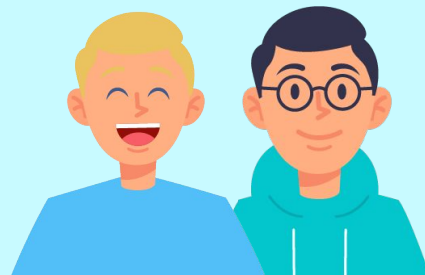


руководство



качество

dev & qa



dev ops



ops

**куберизация**

**качественно**



нет ресурсов

бизнес

качество



руководство



качество

скорость

dev & qa



dev ops



ops

**куберизация**

**качественно**

**быстро**



скорость

нет ресурсов

бизнес

качество

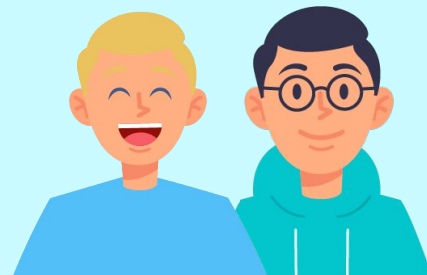


руководство

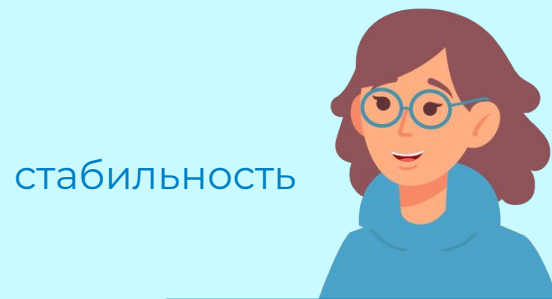


качество  
скорость

dev & qa



dev ops



стабильность

ops

**куберизация**

**качественно**

**быстро**

**стабильно**



скорость  
нет ресурсов

бизнес

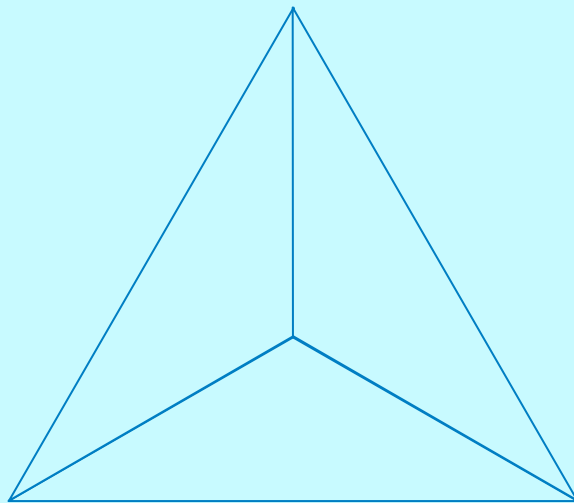


качество

руководство

# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

**качественно**

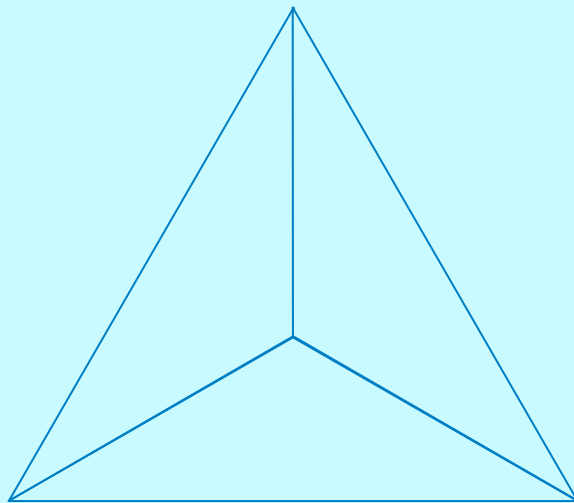


**стабильно**

**быстро**

# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

**качественно**



**стабильно**

техдолг

**быстро**

# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

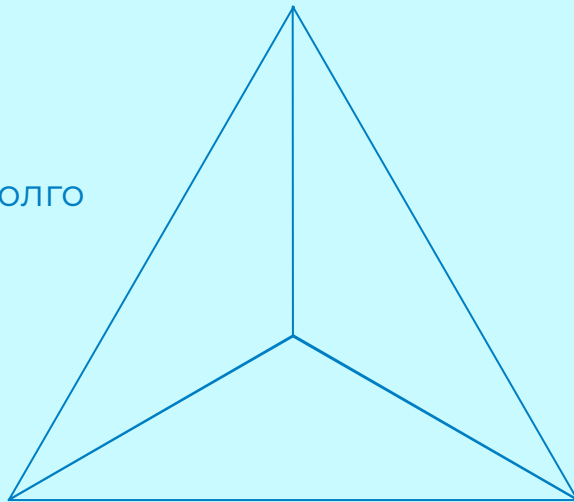
**качественно**

долго

**стабильно**

техдолг

**быстро**





# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

**качественно**

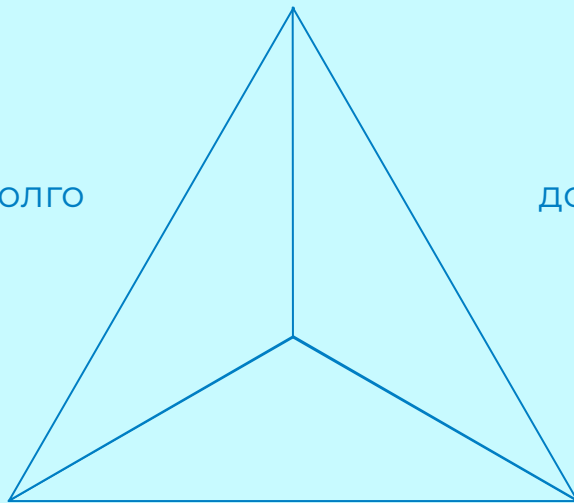
долго

дорого

**стабильно**

техдолг

**быстро**



# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

**качественно**

долго

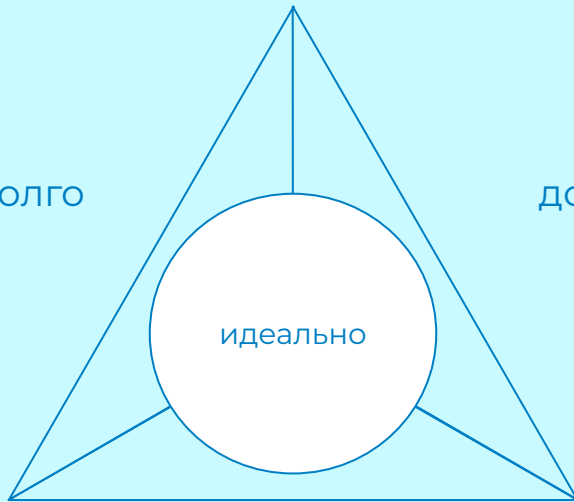
дорого

идеально

**стабильно**

техдолг

**быстро**



# ТРЕУГОЛЬНИК ТРЕБОВАНИЙ

**качественно**

долго

дорого

идеально

**стабильно**

техдолг

**быстро**



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

миграция сократит техдолг

**стабильность**

никаких ошибок из-за миграции

**количество**

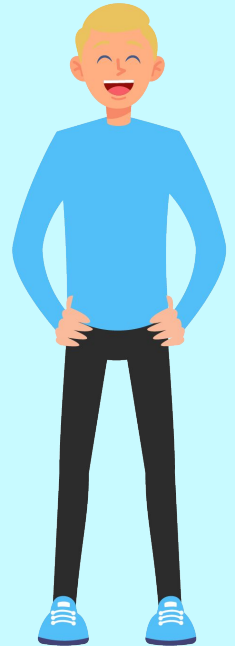
справиться с ворохом проектов

**ноль изменений**

минимум действий для разработки

**скорость**

срок – год



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

миграция сократит техдолг

**стабильность**

никаких ошибок из-за миграции

**количество**

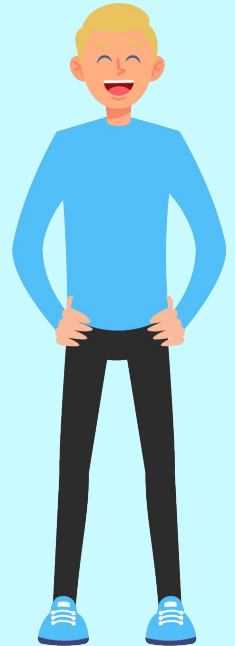
справиться с ворохом проектов

**ноль изменений**

минимум действий для разработки

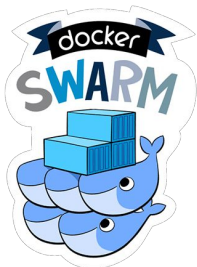
**скорость**

срок – год



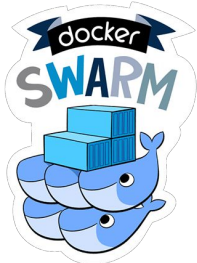
качество

миграция сократит техдолг



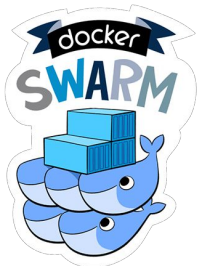
качество

миграция сократит техдолг



качество

миграция сократит техдолг



Skydeploy



качество

миграция сократит техдолг



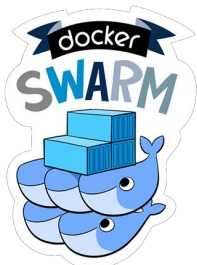
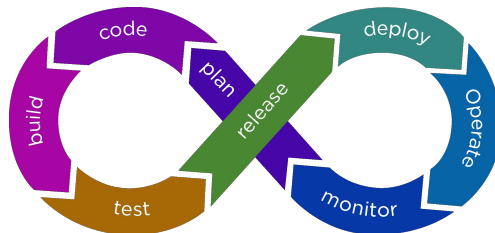
DOMAIN  
SPECIFIC  
LANGUAGES



Skydeploy

качество

миграция сократит техдолг



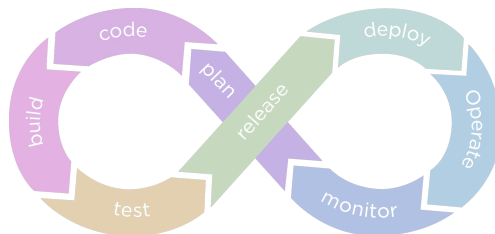
DOMAIN  
SPECIFIC  
LANGUAGES



Skydeploy

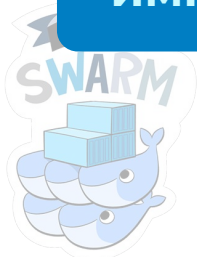
**качество**

миграция сократит техдолг



**императивность**

**декларативность**



Skydeploy



**DOMAIN  
SPECIFIC  
LANGUAGES**

## качество

миграция сократит техдолг

ратуем за качество

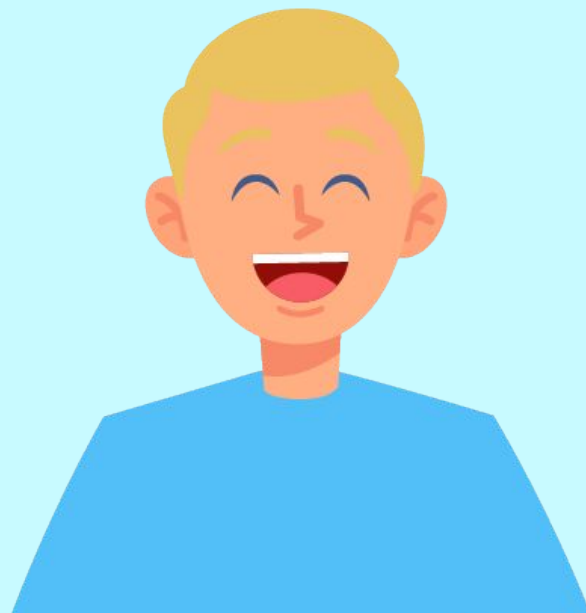
принимаем коллективные решения

проверяем друг друга

пишем тесты

собираем архитектурные советы

общаемся, делимся решениями



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

**приоритет качества, коллективные решения, тесты и экспертиза снаружи**

**стабильность**

никаких ошибок из-за миграции

**количество**

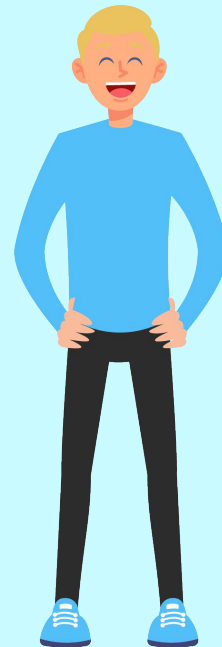
справиться с ворохом проектов

**ноль изменений**

минимум действий

**скорость**

срок – год



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

**приоритет качества, коллективные решения, тесты и экспертиза снаружи**

**стабильность**

никаких ошибок из-за миграции

**количество**

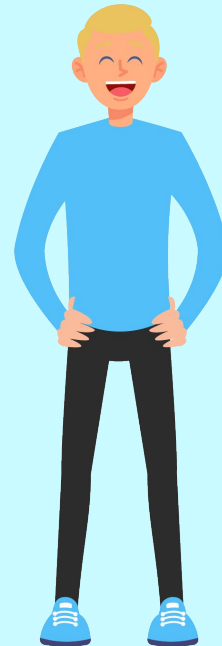
справиться с ворохом проектов

**ноль изменений**

минимум действий

**скорость**

срок – год



## стабильность

никаких ошибок из-за миграции

~95-99(100)% **покрытия тестами**, кода немного

предусмотреть **моментальный откат**

показать максимально наглядно, что **миграция пройдет хорошо**

предусмотреть **ошибки будущего**

убрать человеческий фактор – **максимально автоматизировать** процесс

**пайплайны не содержат логики**

все описывать максимально **понятно и кратко**

# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

**приоритет качества, коллективные решения, тесты и экспертиза снаружи**

**стабильность**

**тесты, план Б, куча проверок, уведомления**

**количество**

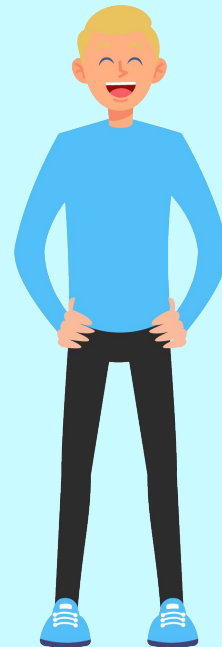
справиться с ворохом проектов

**ноль изменений**

минимум действий

**скорость**

срок – год





# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

**приоритет качества, коллективные решения, тесты и экспертиза снаружи**

**стабильность**

**тесты, план Б, куча проверок, уведомления**

**количество**

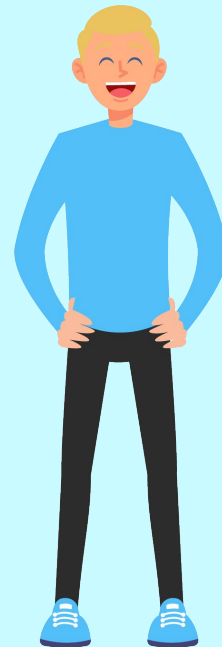
справиться с ворохом проектов

**ноль изменений**

минимум действий

**скорость**

срок – год



## КОЛИЧЕСТВО

справиться с ворохом проектов



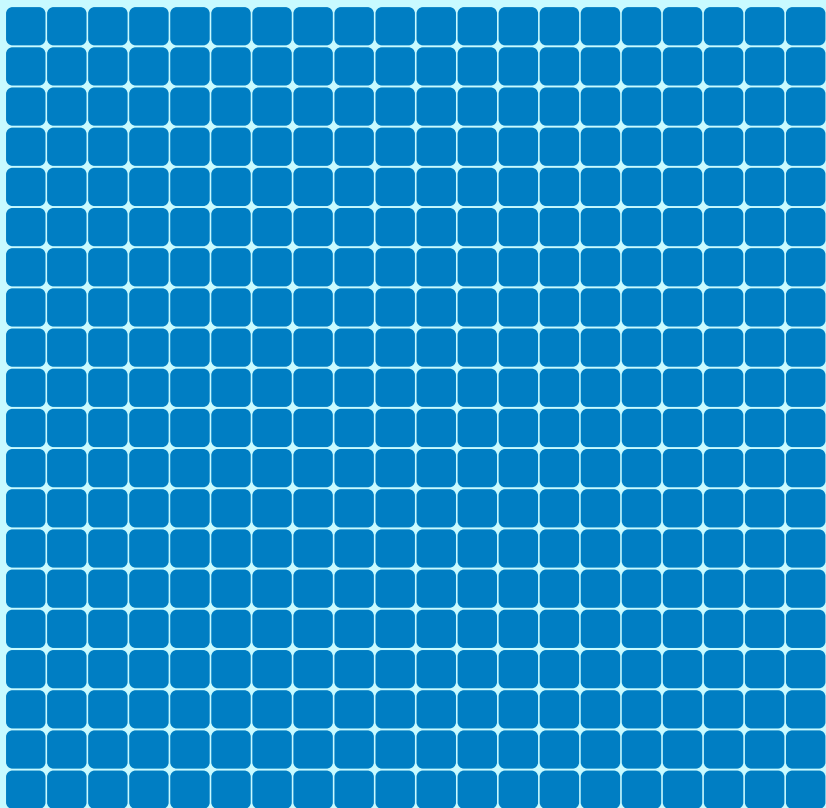
это один микросервис



а это пара инженеров

## КОЛИЧЕСТВО

справиться с ворохом проектов



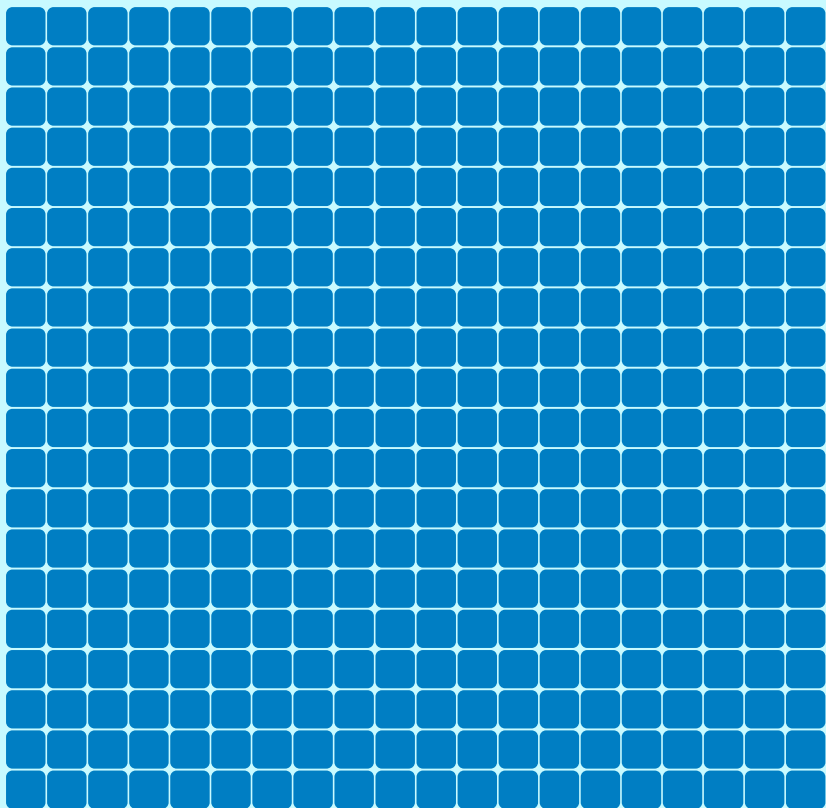
← это 400 микросервисов



а это все еще пара инженеров

## КОЛИЧЕСТВО

справиться с ворохом проектов



это 400 микросервисов



а это штат dev & qa

# КОЛИЧЕСТВО

# справиться с ворохом проектов

- Создать дочернюю страницу по приложению, скопировав эту страницу, и прикрепить ссылку в тикет
- Заполнить **предварительно скопированный чеклист по приложению** и попросить команду проекта дозволнить / проверить этот ч
- Натравить skuehex wta-проект на докеризируемые проект
- Адаптировать проект
  - Приготовить docker-compose.\* конфиги (О docker-compose.\*)
    - Заархивировать порт приложения в таблице
  - Приготовить сборочную обвязку
    - Dockerfile
    - pipeline.yml
    - gitignore и .dockerignore
    - перенести и адаптировать erb-шаблоны конфигов приложения
      - nginx
        - задать set\_real\_ip\_from заголовки: для приложений, которые уже за балансировщиком, заменить snippet https://github.com/skuehex/ansible/blob/master/roles/load-balancer/templates/nginx.conf.j2
      - cron (переносим as-is: один контейнер со всеми задачами, шаблон конфига для крона берётся из конфигов календ
      - supervisor (переносим as-is: один контейнер со всеми задачами, шаблон конфига для супервизора берётся из config-project-files-checksum.sh
  - Адаптировать код приложения
    - указать уникальное имя для файла с логами
    - оставить одну точку входа для веб-приложения (для старых версий Symfony)
  - Приготовить skuehex-обвязку
    - skuehex.json
    - env
    - proxy2docker.conf
    - deploy/docker/dev
    - добавить мета-данные приложения в skuehex
    - протестировать развёртывание на локалосте
    - протестировать развёртывание на удалённом окружении
  - Перенести env переменные в pipeline.yml из Jenkins-pipeline
  - Добавить конфиг: файл для пайплайна <https://github.com/skuehex/jenkins-pipeline/tree/master/resources/skuehex/jenkins/pipeline>
  - Изменить Jenkinsfile и JenkinsfileRollback
  - Обеспечить функционал require\_project (см. Приложение с require\_project)
    - Приложение-сервер?
      - Да, адаптировано
        - для тестингов
        - для окружений разработчи
        - для прода
        - протестировано
      - Нет
    - Приложение-клиент?
      - Да, адаптировано
        - обновлена версия пакета
        - обновлён конфиг приложения
        - протестировано
      - Нет
- Функциональность LEGO (см. Пакет LEGO)
  - Имеется, адаптировано
    - пакет обновлён
    - обновлён конфиг приложения и переменные окружения в пайплайне
    - добавить NFS волюм для ассетов, см. ниже
    - LEGO не используется
  - Подключить NFS, если требуется (спросить у разработчиков, нужно ли? как добавить: Подключение NFS в докеризированном приложении)
    - Требуется (например, для пакета LEGO)
      - nfs volume добавлен в docker-composer.stack.yml
      - инициализирующие контейнер скрипты написаны и добавлены в docker-composer.stack.yml
      - Dockerfile обновлен
    - Нет
- Приложение использует БД?
  - Да - Попросить у Ops **добавить разрешения** на обращение к базе с адреса сворма. (Доступ к базам разрешен с определенных адресов, поэтому нужно добавить адреса подсети нашего кластера (10.254.0.0/20) PRIVILEGES ON 'skuehex\_production' TO 'cms' @ '10.254.0.%' )
  - Нет

- Создать дочернюю страницу по приложению, скопировав эту страницу, и прикрепить ссылку в тикет
- Обеспечить достаточность ресурсов в кластере
  - Ресурсы оценены (см. Оценка достаточности ресурсов в кластере)
    - Внести новые ноды в эксплуатацию
- Переключить проект на swarm-версию в проде
  - если приложение использует ID API - то суть в том, что ID проект имеет IP whitelist - см <https://github.com/skuehex/jf/blob/master/ansible/prod-swarm> же ходит к ним через шлюзы - 5.8.77.245,5.8.77.246. Это и надо добавить к вашей переменной в проекте id в файле env-PR, потом сами #auth-dev заапрувают, смедкат и выкатят
  - подготовить балансировщик (skuehex/ansible)
    - создать отдельный .conf.j2 в roles/load-balancer/templates/nginx/
    - добавить запись в inventories/group\_vars/load-balancer-russia.yml
    - добавить запись в roles/load-balancer/tasks/generate\_configs.yml
    - убедиться, что настроен require\_project, если нужно
    - Приложение живёт на домне 4-го уровня?
      - Да - необходимо создать индивидуальные сертификаты и ключ по инструкции Заведение SSL сертификата для нового домена
      - Нет
  - синхронизировать код в приложении: в ветке с swarm адаптация должны быть все коммиты из мастера
  - разделить приложение без бэкграундных задач в swarm. Например, комментировать сервисы крона и супервизора в docker-compose манифесте. Если фоновая задача занимается подготовкой ассетов или генерацией sitemap - ее можно не отключать
  - QA команды тестирует приложение сквозь балансировщика, изменяя адрес хоста приложения в /etc/hosts на IP балансировщика, QA даёт добро
  - согласовать время переключения трафика на прод с людьми и QA.
  - Подготовить мониторинг: трафика приложения в кластере
    - уменьшить TTL доменного имени до 10 сек. Делается это в консоли Амазона <https://console.aws.amazon.com/route53>, ищем поддомен → кликаем на него → меняем TTL → Save Record Set
    - сообщаем о последующих активностях в #infra-news
    - изменить в DNS адрес хоста:
      - шлём уведомление в канал #infra-news о переключении трафика на swarm (нужно указать наименование проекта и домен)
      - если проект критичный и нужно частичное переключение трафика - выполняем переключение по инструкции Динамическое переключение с помощью Route53
        - если частичное переключение не требуется - переключаем домен на IP балансировщиков (188.246.235.67, 188.246.235.69). Все в той же консоли Амазона, ищем поддомен → кликаем на него → меняем Value → Save Record Set
  - мониторим ошибки в логах на протяжении часа (newrelic + zapu + grafana в том числе график по общим ошибкам)
  - отключаем бэкграундные задачи на старом сервере и запускаем на новом. Чтобы отключить бэкграундные задачи необходимо переименовать папку проекта и остановить бегущие процессы. Другие способы ненадежны. Делем это непосредственно перед тем, как подтверждаем деплой бэкграундных задач в swarm
  - запрещаем деплой master ветку на кластере, например, и сообщаем об этом команде; это нужно бэкграундные задачи в двух местах
  - Сделать \_\_ROLLBACK джобу в Jenkins рядом с обычной (ссылаться должна на JenkinsfileRollback в прилож
  - Убедиться, что проект стабилен
    - нужно договориться с тимлидом, когда ветка пойдёт в мастер
    - если возможно, отложить 2 дня (за это время остановить новые релизы)
    - смержить ветку с докером в мастер
    - увеличить TTL доменного имени до 300 сек
    - убрать supervisor и cron конфиги приложения на старом сервере
    - сообщить в канал команды разработки об успешной докеризации проекта, скинуть ссылки на running
    - исправить запуск джобы в [https://skuehex.tech/job/School/job/DeployAllSchoolProjects\\_Parallel](https://skuehex.tech/job/School/job/DeployAllSchoolProjects_Parallel)
  - Оптимизация расходов
    - на старом Selectel-сервере не осталось приложений? если да, то добьются от менеджера Selectel



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

приоритет качества, коллективные решения, тесты и экспертиза снаружи

**стабильность**

тесты, план Б, куча проверок, уведомления

**количество**

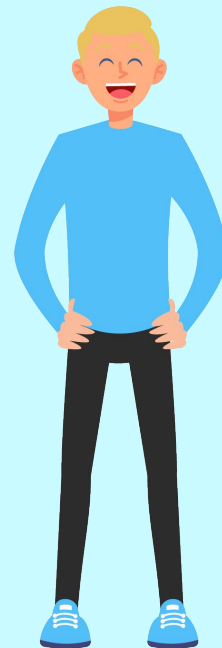
привлекаем к миграции Dev & QA

**ноль изменений**

минимум действий

**скорость**

срок – год



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

приоритет качества, коллективные решения, тесты и экспертиза снаружи

**стабильность**

тесты, план Б, куча проверок, уведомления

**количество**

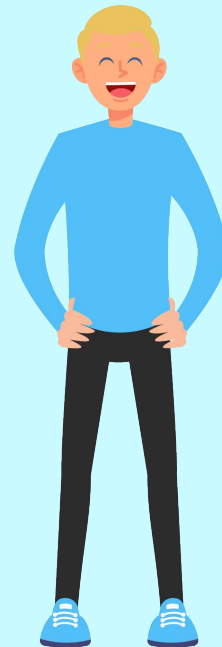
привлекаем к миграции Dev & QA

**ноль изменений**

минимум действий

**скорость**

срок – год



**НОЛЬ ИЗМЕНЕНИЙ**

минимум действий

проектировать **свои решения**

создать **понятный DSL**

искать **удобные решения**, а не брать первые





# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

приоритет качества, коллективные решения, тесты и экспертиза снаружи

**стабильность**

тесты, план Б, куча проверок, уведомления

**количество**

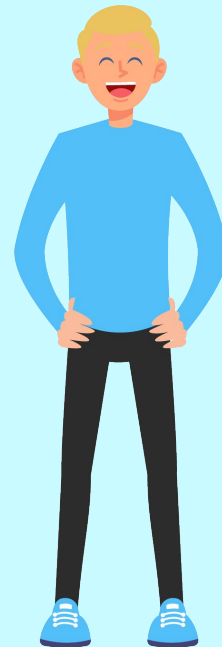
привлекаем к миграции Dev & QA

**ноль изменений**

проектирование, DSL, не спешить

**скорость**

срок – год



# КАКИЕ ПРОБЛЕМЫ РЕШАЕМ

**качество**

приоритет качества, коллективные решения, тесты и экспертиза снаружи

**стабильность**

тесты, план Б, куча проверок, уведомления

**количество**

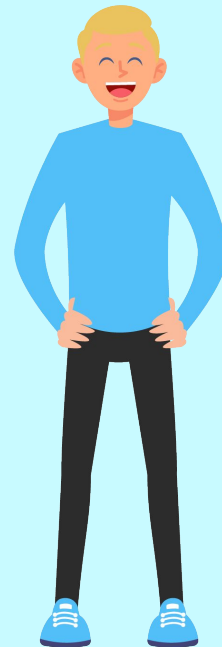
привлекаем к миграции Dev & QA

**ноль изменений**

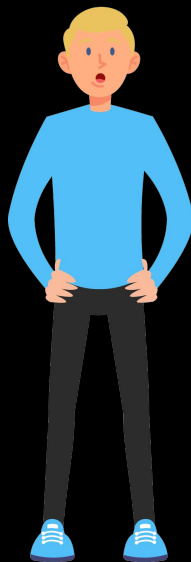
проектирование, DSL, не спешить

**скорость**

срок – год  
(достигается за счет пунктов выше)



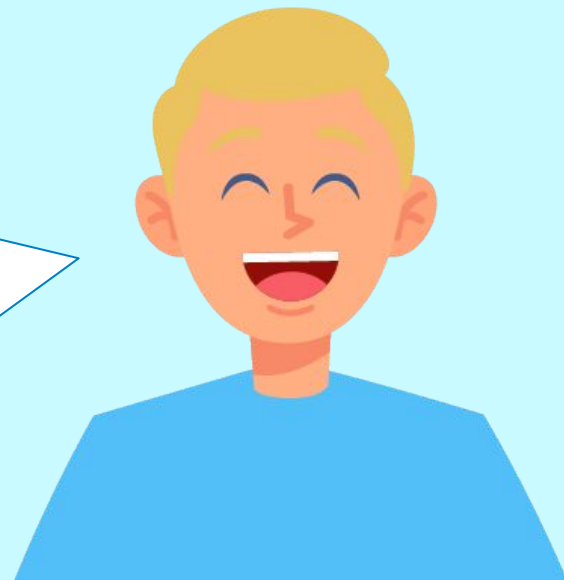
# ПОДГОТОВКА СРЕД



prod – kubespray.io

**ansible** имплементация инсталляции кубов  
работаем с железом и виртуалками

проблем  
замечено  
не было



## test – minikube

**cilium with minikube** не сработал

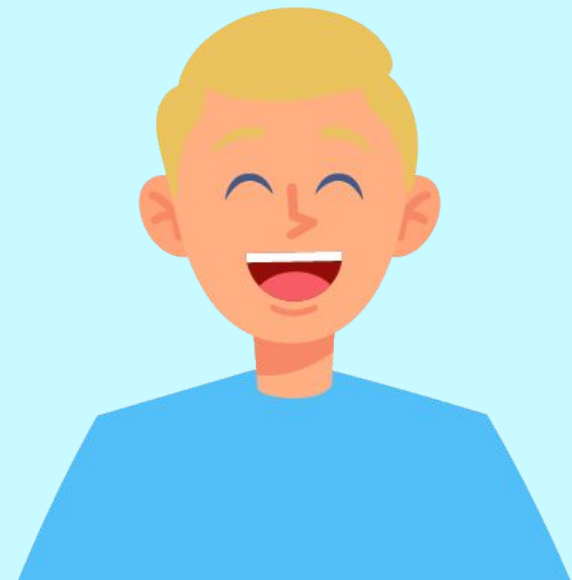
примерно месяц готовили тестовую среду и продовую среду

доработки заняли полгода  
для приведения  
в промышленное использование



local – devspace.sh

тренд уходить от локальной разработки  
но оставляем compose для плавной миграции  
devspace 5 → 6



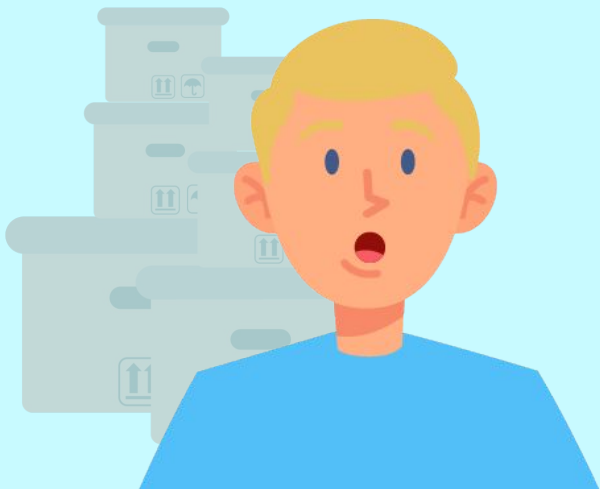


portainer



dashboard

для разработки  
**НИЧЕГО**  
не поменялось



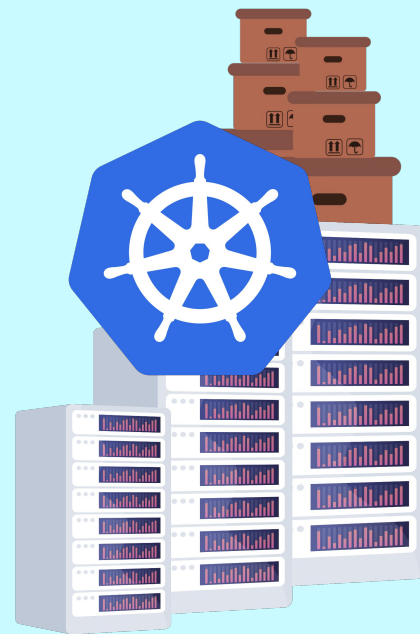
# ДА ЧТО ТАМ ДЕЛАТЬ?





# ДА ЧТО ТАМ ДЕЛАТЬ?

меняется же только оркестратор!



# ИЗМЕНЕНИЯ



# ИЗМЕНЕНИЯ

**изменения базовых образов**

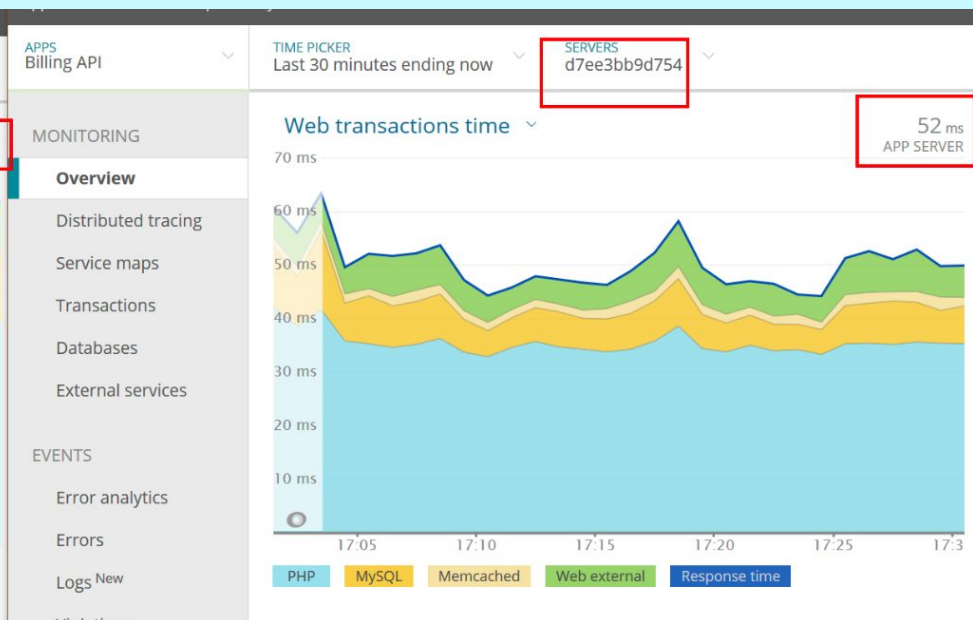
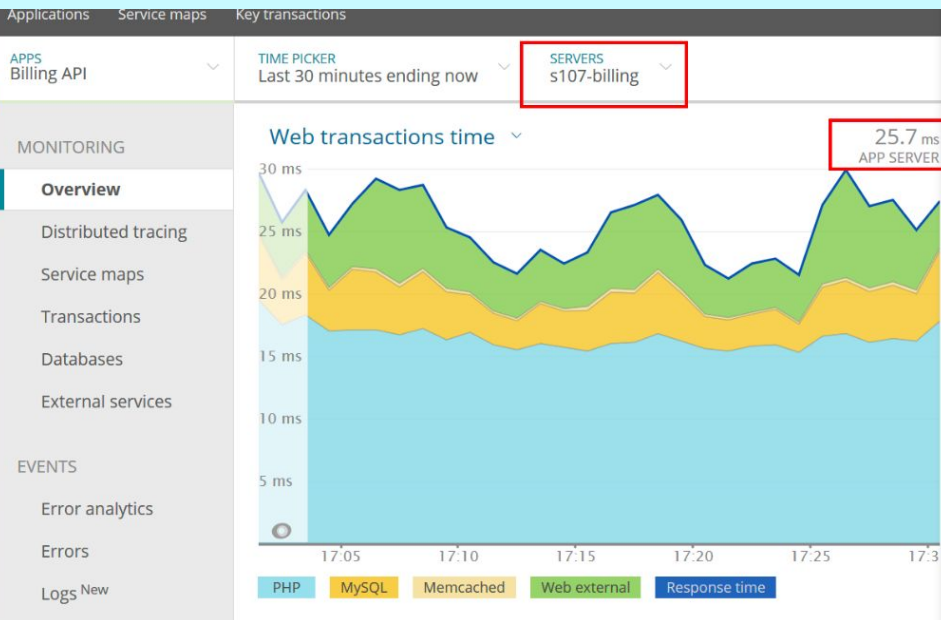
# ИЗМЕНЕНИЯ

## изменения базовых образов

- Alpine -> Debian (libc)
- Метрик собираются по другому
- Поддержка новых xdebug + обновления

# ИЗМЕНЕНИЯ

## изменения базовых образов



# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

- Стандартизировать, убрать кастом
- Выстроить единую систему деплоя и сборки
- Обновить изменившиеся параметры

# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

**изменения внутреннего взаимодействия сети**



# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

**изменения внутреннего взаимодействия сети**

- nginx имеет свой dns
- swarm: stack\_name-container\_name
- k8s: pod\_name

# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

**изменения внутреннего взаимодействия сети**

**изменения в контроле доступа**

# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

**изменения внутреннего взаимодействия сети**

**изменения в контроле доступа**

- Portainer с своей БД и моделью доступов
- k8s dashboard интегрирован с LDAP

# ИЗМЕНЕНИЯ

**изменения базовых образов**

**изменения конфигураций**

**изменения внутреннего взаимодействия сети**

**изменения в контроле доступа**

# РЕАЛИЗАЦИЯ



# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансирьы** (для автоматического переключения трафика)

**мониторинг** (для наблюдения)

**пайплайны** (для упрощения работы – только кликать)

**конвертеры** (DSL и внесение изменений)

**автоматические проверки** (sanity check)

# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансиры** (для автоматического переключения трафика)

**мониторинг** (для наблюдения)

**пайплайны** (для упрощения работы – только кликать)

**конвертеры** (DSL и внесение изменений)

**автоматические проверки** (sanity check)

# РЕАЛИЗАЦИЯ

## DSL

балансиры

мониторинг

пайплайны

конвертеры

автоматические проверки

## Базовый проект

Попробуем создать DSL для эфемерного проекта.

Минимальный DSL:

```
services:  
  app:  
    containers:  
      app:  
        type: custom  
        image: myproject/application:1.2.3
```

Для примеров специально используется тип контейнеров custom, так

Это создаст базовый [список kubernetes ресурсов](#), задеплойв одну реплику



# РЕАЛИЗАЦИЯ

## DSL

балансиры

МОНИТОРИНГ

пайплайны

конвертеры

автоматические проверки

## Добавление memcached

Memcached будет доступен по адресу `tcp://memcached:11211` :

```
----
services:
  app:
    containers:
      app:
        type: custom
        image: myproject/application:1.2.3
+ memcached:
+ containers:
+ memcached:
+ type: memcached
```

# РЕАЛИЗАЦИЯ

## DSL

балансиры

мониторинг

пайплайны

конвертеры

автоматические проверки

## Горизонтальное масштабирование реплик

Включаем горизонтальное масштабирование реплик:

```
---
+environment: production # HPA работает только в production окружении
+
  services:
    app:
+   hpa:
+     enabled: true
+     min: 2 # Минимальное количество реплик
+     max: 20 # Максимальное количество реплик
    containers:
      app:
        type: custom
        image: myproject/application:1.2.3
```

# РЕАЛИЗАЦИЯ

## DSL

балансиры

мониторинг

пайплайны

конвертеры

автоматические проверки

## Добавление nginx контейнера

Добавляем нашему приложению nginx контейнер, в тот же сервис где

```
---
services:
  app:
    containers:
+   nginx:
+     type: nginx
+     image: myproject/nginx:1.2.3
+     environment:
+       PROXY_PASS: http://127.0.0.1:80
    app:
      type: custom
      image: myproject/application:1.2.3
```

# РЕАЛИЗАЦИЯ

## DSL

балансиры

мониторинг

пайплайны

конвертеры

автоматические проверки

## RBAC Разрешить команде работать с проектом

Добавляем доступ к проекту команде DevelopmentTeam :

```
----  
+RBAC:  
+ teams:  
+   - DevelopmentTeam  
+  
services:  
  app:  
    containers:  
      app:  
        type: custom  
        image: myproject/application:1.2.3
```

# РЕАЛИЗАЦИЯ

## DSL

балансиры

МОНИТОРИНГ

пайплайны

конвертеры

автоматические проверки

## Аргументы запуска контейнера

Аргументы запуска контейнера можно передавать как строкой, так и массивом:

```
---
services:
  app:
    containers:
      app:
        type: custom
        image: myproject/application:1.2.3
+   command: php bin/console # строка
+   args: # массив
+     - status:update
+     - -i
+     - 60
+     - -l
+     - 50
+     - -s
+     - "*" * * * *
```

# РЕАЛИЗАЦИЯ

## DSL

балансиры

мониторинг

пайплайны

конвертеры

автоматические проверки

## Регулирование скорости деплоя

По умолчанию, kubernetes может выключить до 25% реплик во

Это поведение контролируется отдельной опцией:

```
---
services:
  app:
+ max_unavailable: 1 # апдейтим по 1 контейнеру за
  containers:
    app:
      type: custom
      image: myproject/application:1.2.3
```

# DSL

```
51 services:
52   cron:
53     containers:
54       cron:
55         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
56         type: cron
57       replicas: 0
58   memcached:
59     containers:
60       memcached:
61         type: memcached
62   supervisor:
63     containers:
64       supervisor:
65         environment:
66           CONTAINER_CONFIG: |
67             [
68               {
69                 "source": "deploy/templates/supervisor_demo.conf.erb",
70                 "target": "/etc/supervisor.d/app.ini"
71               }
72             ]
73         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
74         type: supervisor
75       replicas: 0
76   web:
77     containers:
78       nginx:
79         environment:
80           FASTCGI_PASS: 127.0.0.1:9000
81           NGINX_SERVER_NAME: test-app.skyeng.link
82         image: registry.skyeng.tv/infradev/test-app-nginx:1adff2b18744f324d487a5041cbbff35
83         ports:
84         - 80
85         - 443
86         type: nginx
87       php-fpm:
88         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
89         type: php-fpm
90     ports:
91     - 8293:80
92     replicas: 2
```

# DSL

```
51 services:
52   cron:
53     containers:
54       cron:
55         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
56         type: cron
57       replicas: 0
58   memcached:
59     containers:
60       memcached:
61         type: memcached
62   supervisor:
63     containers:
64       supervisor:
65         environment:
66           CONTAINER_CONFIG: |
67             [
68               {
69                 "source": "deploy/templates/supervisor_demo.conf.erb",
70                 "target": "/etc/supervisor.d/app.ini"
71               }
72             ]
73         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
74         type: supervisor
75       replicas: 0
76   web:
77     containers:
78       nginx:
79         environment:
80           FASTCGI_PASS: 127.0.0.1:9000
81           NGINX_SERVER_NAME: test-app.skyeng.link
82         image: registry.skyeng.tv/infradev/test-app-nginx:1adff2b18744f324d487a5041cbbff35
83         ports:
84           - 80
85           - 443
86         type: nginx
87       php-fpm:
88         image: registry.skyeng.tv/infradev/test-app-php:1adff2b18744f324d487a5041cbbff35
89         type: php-fpm
90     ports:
91       - 8293:80
92     replicas: 2
```

```
1 ---
2 RBAC:
3   teams: ${RBAC_TEAMS}
4   users: ${RBAC_USERS}
5   dsl_version: 0.1
6   files:
7     require_cert.crt: ${SERVICE_CRM_CRT_FILE}
8     require_cert_key: ${SERVICE_CRM_KEY_FILE}
9     dot_env: ./env
10  services:
11    web:
12      replicas: ${SWARM_PHPFPM_REPLICAS}
13      ports:
14        - 8089:80
15      containers:
16        nginx:
17          environment:
18            NGINX_SERVER_NAME: ${NGINX_SERVER_NAME}
19            FASTCGI_PASS: ${FASTCGI_PASS}
20          image: registry.skyeng.tv/sales/crm2-backend-nginx:${IMAGE_TAG_nginx}
21          type: nginx
22          php-fpm:
23            files:
24              - source: require_cert.crt
25                target: /opt/app/var/require_project/service.crt
26              - source: require_cert_key
27                target: /opt/app/var/require_project/service.key
28          image: registry.skyeng.tv/sales/crm2-backend-php:${IMAGE_TAG_php}
29          type: php-fpm
30    cron:
31      replicas: 1
32      containers:
33        cron:
34          files:
35            - source: require_cert.crt
36              target: /opt/app/var/require_project/service.crt
37            - source: require_cert_key
38              target: /opt/app/var/require_project/service.key
39          image: registry.skyeng.tv/sales/crm2-backend-php:${IMAGE_TAG_php}
40          type: cron
41    supervisor:
42      replicas: ${CONSUMER_REPLICAS_DEFAULT}
43      containers:
44        supervisor:
45          environment:
46            CONTAINER_CONFIG: |\n \{\n   "source": "${SUPERVISOR_CONFIG_TEMPLATE}",\n   "target": "/etc/supervisor.d/app.ini"\n \}
47          files:
48            - source: require_cert.crt
49              target: /opt/app/var/require_project/service.crt
50            - source: require_cert_key
51              target: /opt/app/var/require_project/service.key
52          image: registry.skyeng.tv/sales/crm2-backend-php:${IMAGE_TAG_php}
53          ports:
54            - 9001
55          type: supervisor
56    memcached:
57      containers:
58        memcached:
59          environment:
60            MEMORY_LIMIT: 1024
61          type: memcached
```



# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансир** (для автоматического переключения трафика)

**мониторинг** (для наблюдения)

**пайплайны** (для упрощения работы – только кликать)

**конвертеры** (DSL и внесение изменений)

**автоматические проверки** (sanity check)

# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансиры** (для автоматического переключения трафика)

- доработали роль
- сделали для всех проектов сразу автоматически доступ по куке
- зафиксировали контракт на актуальность мастера
- некоторые проекты все равно остались только для ручных 100% переключений (ночные работы)

# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансиры** (для автоматического переключения трафика)

**мониторинг** (для наблюдения)

**пайплайны** (для упрощения работы – только кликать)

**конвертеры** (DSL и внесение изменений)

**автоматические проверки** (sanity check)

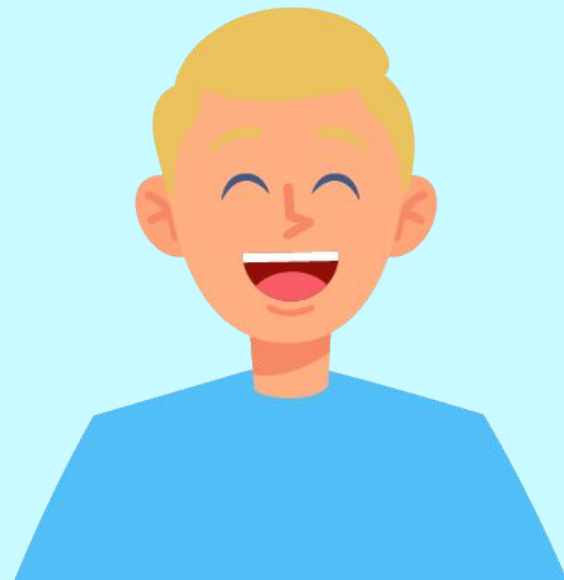
## МОНИТОРИНГ (для наблюдения)

уже было готово

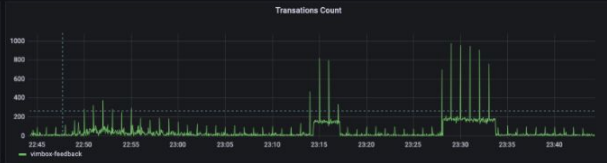
Для почти каждого проекта есть автодашборд,  
там есть все стандартное для проекта

- HTTP трафик
- длина очередей
- количество консьюмеров
- потребление сри/тет
- количество ошибок

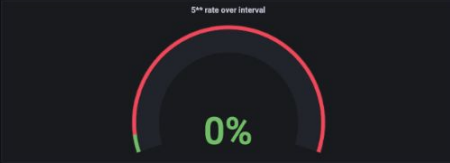
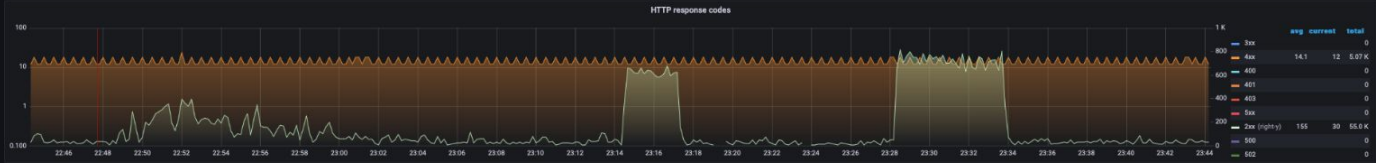
Много метрик не бывает (шум надо убирать)



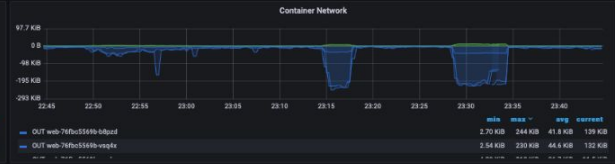
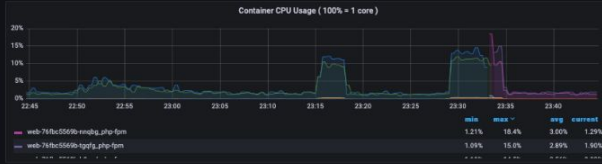
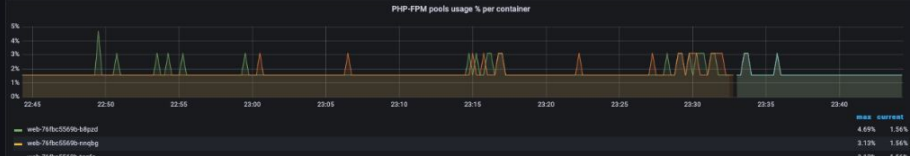
- APM



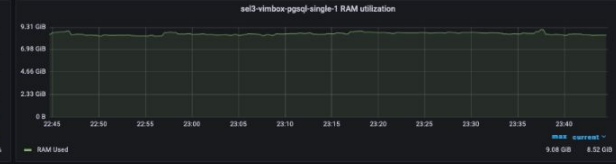
- HTTP metrics



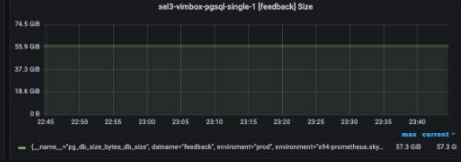
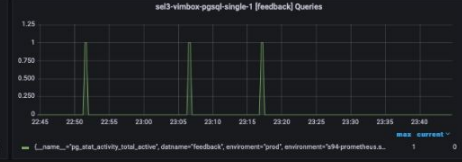
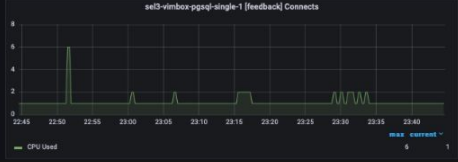
Cluster resources



Databases hosts



Databases



# РЕАЛИЗАЦИЯ

**DSL** (для простоты деплоя)

**балансиры** (для автоматического переключения трафика)

**мониторинг** (для наблюдения)

**пайплайны** (для упрощения работы – только кликать)

**конвертеры** (DSL и внесение изменений)

**автоматические проверки** (sanity check)

**пайплайны** (для упрощения работы – только кликать)

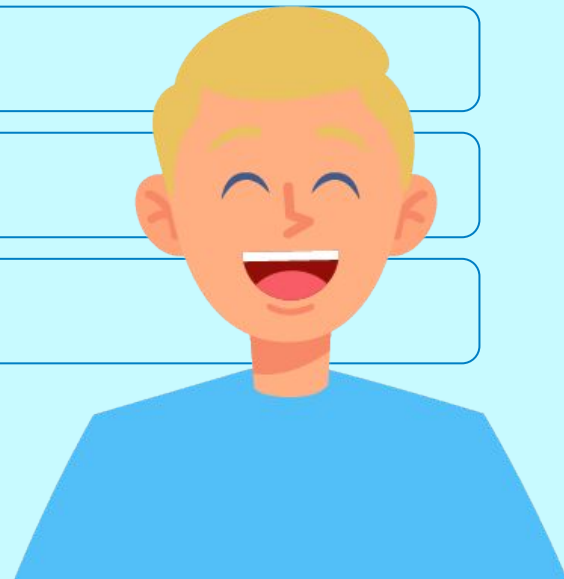
программируйте на бумажке

договаривайтесь

обсуждайте шаги

**НЕ САДИТЕСЬ СРАЗУ ЗА РЕАЛИЗАЦИЮ!**

декларативное программирование – это навык





## пайплайны (для упрощения работы – только кликать)

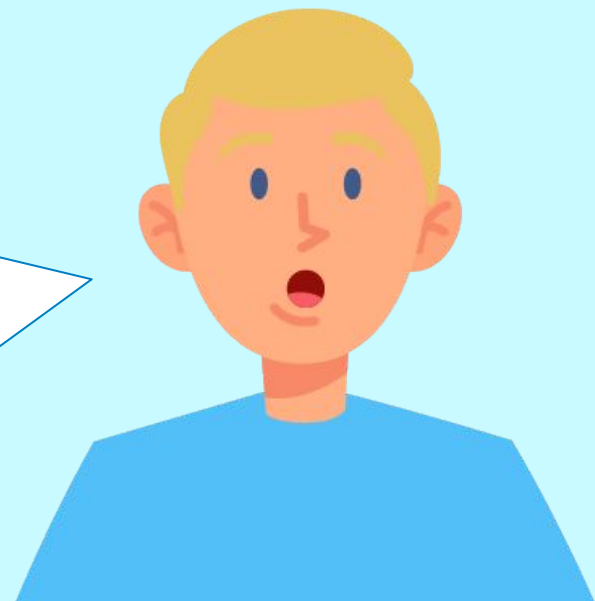
какие нам нужны шаги?

- 1 создать ветку
- 2 задеплоить ветку на тест и на прод
- 3 потестировать
- 4 проверить по кукле
- 5 переключить веб трафик
- 6 переключить асинхронные сервисы
- 7 смерджить ветку
- 8 задеплоить с мастера



**пайплайны** (для упрощения работы – только кликать)

нам нужен  
один пайплайн?



S	W	Name ↓	Last Success	Last Failure
		Infra » K8sation	8 days 6 hr - #454	8 days 0 hr - #456
		Infra » K8sation_develop	10 mo - #1	3 mo 0 days - #21
		Infra » monorepo_K8sation_develop	5 mo 3 days - #206	N/A
		Infra » pre_K8sation	10 days - #1009	7 days 2 hr - #1018
		Infra » semi_automated_K8sation	6 days 3 hr - #194	1 day 21 hr - #197
		Infra » semi_automated_K8sation_develop	8 mo 0 days - #6	3 mo 20 days - #10
		Infra » semi_prek8s	7 mo 7 days - log	N/A
		Infra » semi_prek8s » develop	4 mo 12 days - #11	4 mo 12 days - #10
		Infra » semi_prek8s » master	N/A	7 days 1 hr - #8
		Infra » semi_prek8s » MR-45	N/A	7 mo 8 days - #1

← основной пайплайн

← фронтальный

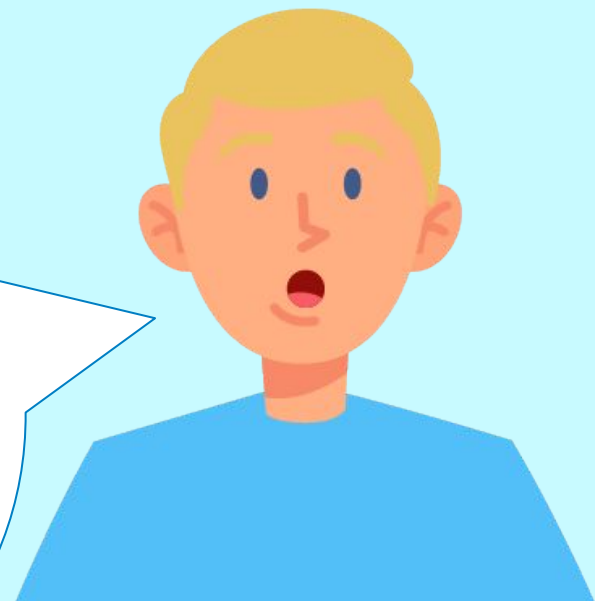
← проверочный

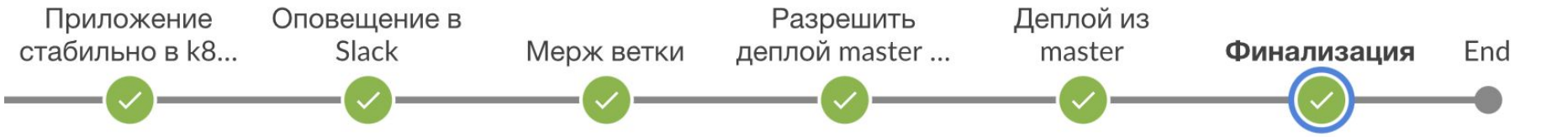
← полуавтомат

← проверочный  
полуавтомат

**пайплайны** (для упрощения работы – только кликать)

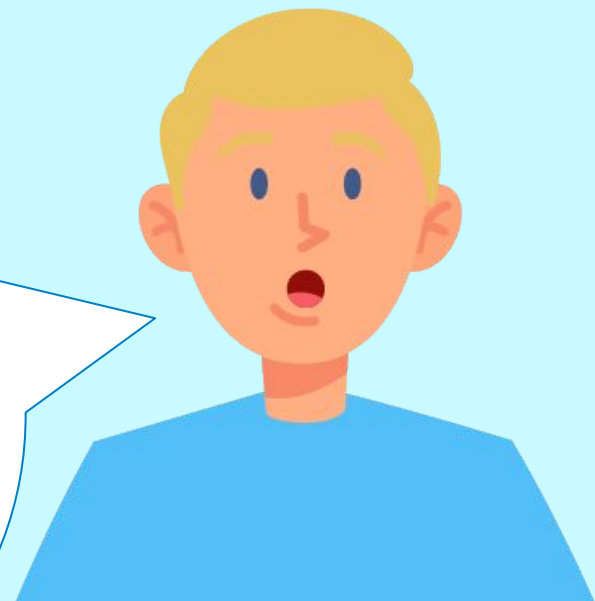
максимально  
простые действия





**пайплайны** (для упрощения работы – только кликать)

максимальная  
информативность



## # k8s-migration

New Jenkins



**New Jenkins** APP 11 minutes ago

Тред миграции в K8S проекта `test-app` (infra/test-app).

### Пайплайн кубернетизации

<https://j.skyeng.tech/job/Infra/job/K8sation/457/>

### Полезные ссылки

- `application url` <https://test-app.skyeng.link>
- `ссылка в gitlab'e` <https://gitlab.skyeng.link/skyeng/test-app>
- `задача в Jira` <https://jira.skyeng.tech/browse/INFRADEV-7777>
- `k8s-dashboard` <https://sel3-common-k8s-dashboard.skyeng.link/#/deployment?namespace=test-app>

### Дашборды

- `автодашборд` <https://grafana.skyeng.link/d/wCtyLp9Mk/test-app-generated-dashboard>
- `newrelic` <https://rpm.newrelic.com/accounts/1471826/applications/1062238805>
- `sentry` null
- `kibana` null

[@a.vazhenin](#) [@Roman Burnashev](#) подпишитесь на этот тред.



**New Jenkins** APP 11 minutes ago

Ветка готова: <https://gitlab.skyeng.link/skyeng/test-app/-/compare/master...INFRADEV-7777>



**New Jenkins** APP 8 minutes ago

Ветка INFRADEV-7777 готова: <https://gitlab.skyeng.link/skyeng/test-app/-/compare/master...INFRADEV-7777>



**New Jenkins** APP 8 minutes ago

Ветка INFRADEV-7777 протестирована автоматикой.  
Возвращайся в пайплайн: <https://j.skyeng.tech/job/Infra/job/K8sation/457/>  
Следующий шаг: деплой на тестинг для ручного тестирования. Перед выбором хоста для деплоя, убедись, что он включен!



**New Jenkins** APP 6 minutes ago

Приложение разделено на test-y220



**New Jenkins** APP 6 minutes ago

Тестирование на тестинге завершено



**New Jenkins** APP 6 minutes ago

Приложение разделено в K8S без крона и консьюмеров.  
К нему можно получить доступ по куке, см.  
<https://confluence.skyeng.tech/pages/viewpage.action?pagelD=148842742>



**New Jenkins** APP 7 minutes ago

Проверьте вручную, что ветка INFRADEV-7777 up-to-date с master (TODO проверять автоматически)



**New Jenkins** APP 7 minutes ago

Деплой test-app (infra/test-app) из мастера запрещён на время тестирования в K8S в проде (однако, деплой не заблокирован технически!).  
fyi [@Roman Burnashev](#) [@Roman Burnashev](#) [@Mikhail Tarakanov](#)  
[@Airat Tuktamyshev](#) [@maxim.pavlov](#) [@Aleksandr Prusov](#) [@Artem Yaryigin](#)  
[@a.vazhenin](#) [@vitaliy.krasnoperov](#) [@k.netesin](#)



**New Jenkins** APP 7 minutes ago

Готовы работать с PROD трафиком. При запуске PROD трафика следует мониторить дашборды приложения, ссылки на которые приведены в шапке треда



**New Jenkins** APP 7 minutes ago

Пересоздаём ветку кубернетизации с включёнными бэгрраундными задачами



**New Jenkins** APP 6 minutes ago

Отключаем бэгрраундные задачи в Swarm'e (те, которые не 'test-app\_(nginx|php-fpm|redis|memcached)')



**New Jenkins** APP 6 minutes ago

Деплоим бэгрраундные задачи в k8s



**New Jenkins** APP 5 minutes ago

Перенос бэгрраундных задач в K8S завершен



**New Jenkins** APP 5 minutes ago

Замержи ветку в Gitlab'e: [https://gitlab.skyeng.link/skyeng/test-app/-/merge\\_requests?scope=all&state=opened&search=INFRADEV-7777](https://gitlab.skyeng.link/skyeng/test-app/-/merge_requests?scope=all&state=opened&search=INFRADEV-7777)



**New Jenkins** APP 5 minutes ago

Деплой test-app (infra/test-app) из мастера разрешён  
fyi [@Roman Burnashev](#) [@Roman Burnashev](#) [@Mikhail Tarakanov](#)  
[@Airat Tuktamyshev](#) [@maxim.pavlov](#) [@Aleksandr Prusov](#) [@Artem Yaryigin](#)  
[@a.vazhenin](#) [@vitaliy.krasnoperov](#) [@k.netesin](#)



**New Jenkins** APP 5 minutes ago

Финальный шаг - деплой в K8s из мастера. Подтверди это действие в пайплайне



**New Jenkins** APP 3 minutes ago

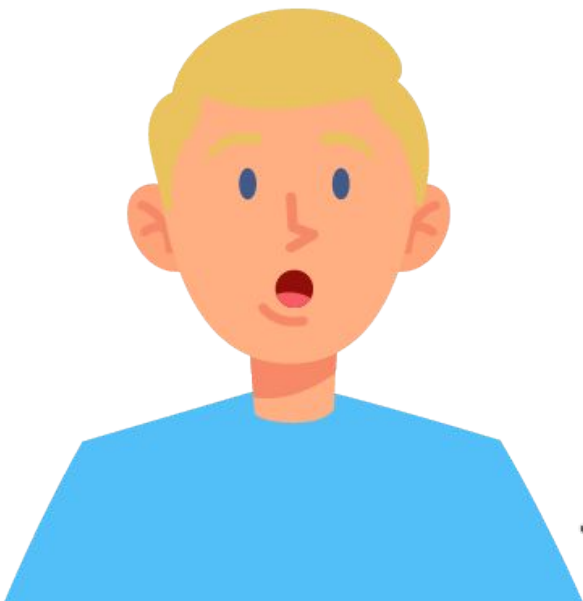
Кубернетизация завершена!  
fyi [@Aleksandr Prusov](#) [@Roman Burnashev](#)



**New Jenkins** APP 3 minutes ago

Мы подождём еще 48 часов прежде чем полностью удалим приложение из Swarm.  
Отменить это действие можно в пайплайне, который останется активным на всё это время

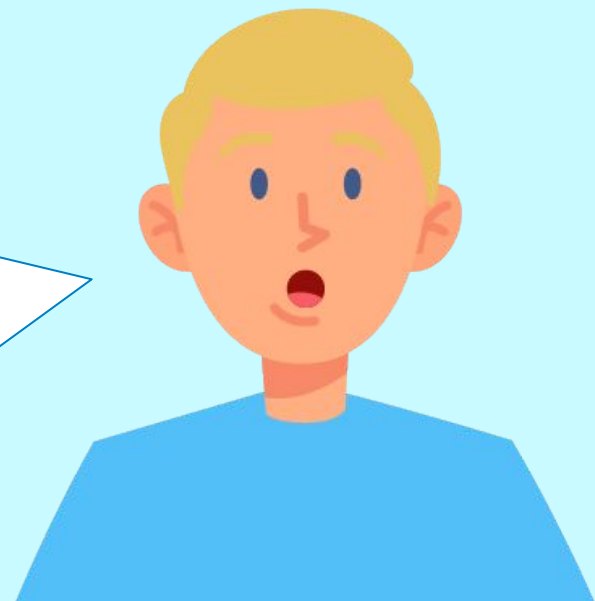
обязательно уведомлять в информационный канал об изменении ситуации в проде





**пайплайны monorepo – это фронт**

monorepo



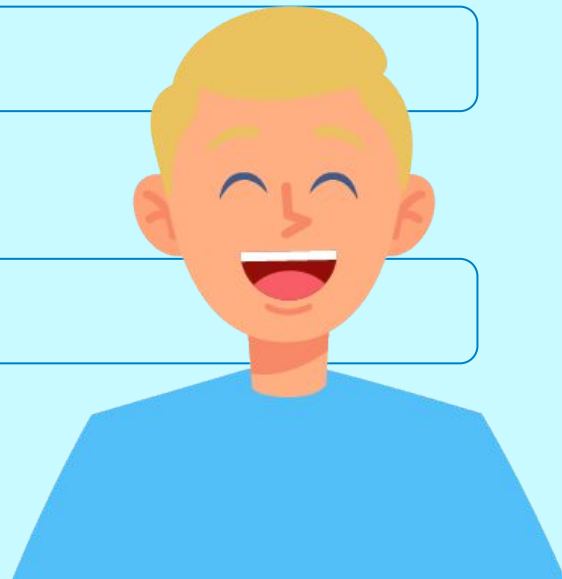
**пайплайны** monorepo – это фронт

медленно собирается

много изменений == мастер бежит очень быстро

проектов много

Применили практику **as\_is**



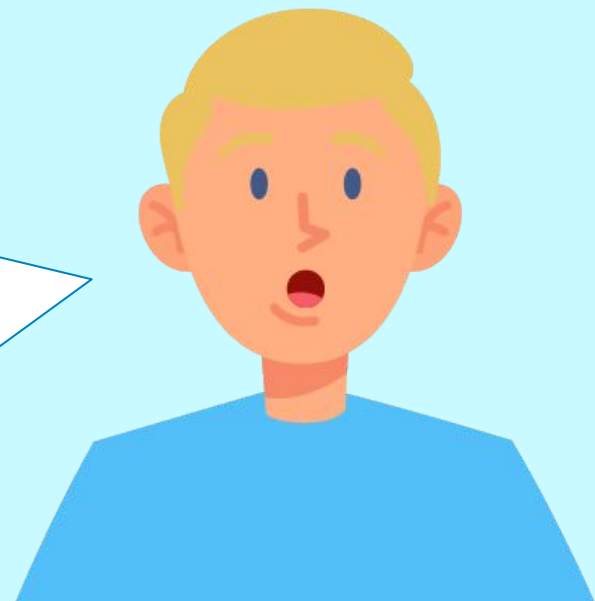
# зачем нужен пайплайн monorepo

меняется же только оркестратор!



**пайплайны** semi automated – полуручной (полуавтоматический)

semi automated



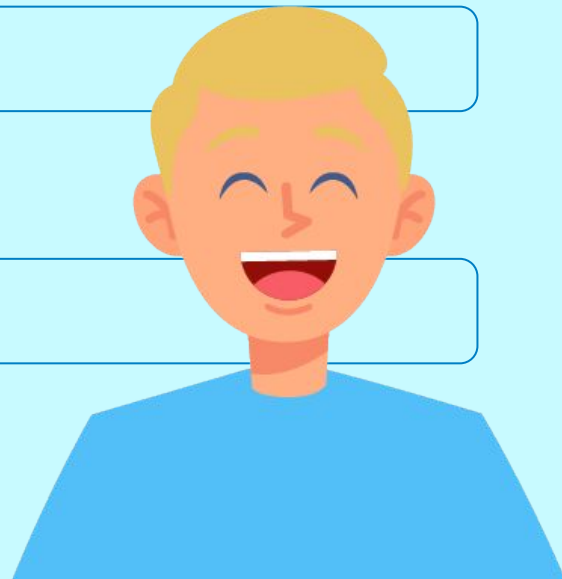
**пайплайны** semi automated – полуручной (полуавтоматический)

не все может собраться автоматически (**план Б**)

**ручные** дополнительные изменения

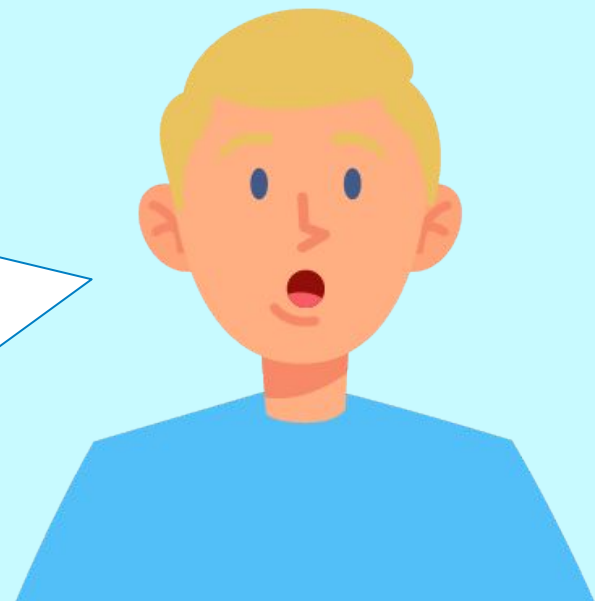
**специфика** или стандартизация в момент миграции

вносим исправления **вручную** в ветку

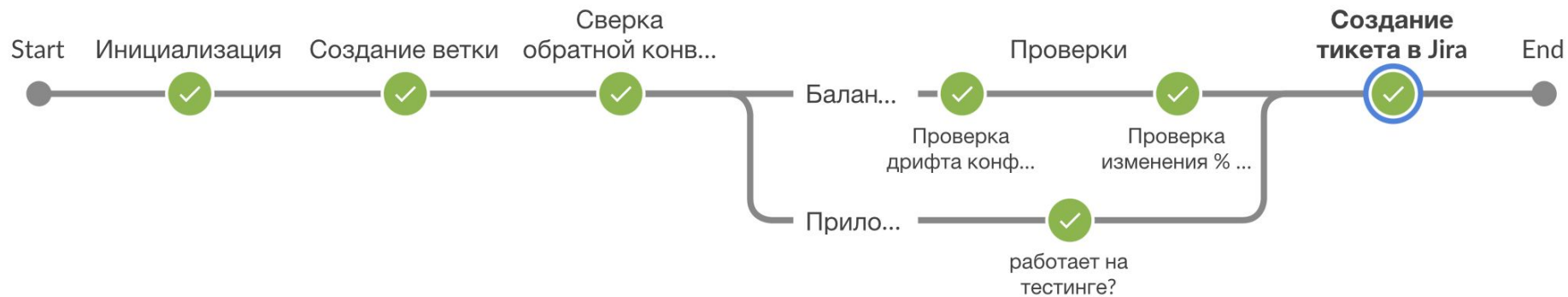


**пайплайны** pre k8sation – это проверочный

pre k8sation



# пайплайны pre k8sation – это проверочный



## Задача

Мигрировать проект api-evaluation ( hr/performance-review-back ) в K8S.

## Как решать

1. Назначить задачу на QA инженера.
2. QA инженер запускает пайплайн кубернетизации и проводит его **до** этапов работы с PROD трафиком, а далее привлекает разработчика сервиса
3. При помощи пайплайна разработчик сервиса переносит workload на K8S и mergeит ветку в мастер.

Больше подробностей можно найти ниже, в разделе **ссылки**.

Пайплайн кубернетизации: <https://j.skyeng.tech/blue/organizations/jenkins/Infra%2FK8sation/>

В форму запуска пайплайна ввести:

- ключ проекта (PROJECT): hr/performance-review-back
- номер тикета (JIRA\_TASK): номер этой задачи (HRMDEV-...)

## DoD

- Workload приложения полностью в K8S
- MR, созданный в рамках пайплайна, принят и замёржен

## Ссылки

1. Подробное описание пайплайна: <https://confluence.skyeng.tech/pages/viewpage.action?pagelId=148846585>
2. Видео-демонстрация работы с пайплайном: <https://drive.google.com/file/d/1eGQZtKN4kJI5Az-O1qRC1XI5JJshNSdQ/view?usp=sharing>
3. [Видео](#) о миграции в K8S с #dev-talks и [презентация](#)
4. Инструменты разработчика в K8s: <https://confluence.skyeng.tech/pages/viewpage.action?pagelId=148847570>



## Задача

Мигрировать проект `api-evaluation ( hr/performance-review-back )` в K8S.

## Как решать

1. Назначить задачу на `QA инженера`.
2. QA инженер запускает пайплайн кубернетизации и проводит его **до** этапов работы с PROD трафиком, а далее привлекает разработчика сервиса
3. При помощи пайплайна разработчик сервиса переносит workload на K8S и mergeит ветку в мастер.

Больше подробностей можно найти ниже, в разделе **ссылки**.

Пайплайн кубернетизации: <https://j.skyeng.tech/blue/organizations/jenkins/Infra%2FK8sation/>

В форму запуска пайплайна **ввести**:

- ключ проекта (PROJECT): `hr/performance-review-back`
- номер тикета (JIRA\_TASK): номер этой задачи (HRMDEV-...)

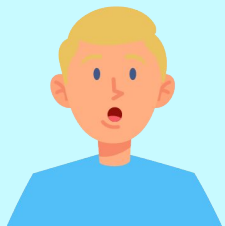
## DoD

- Workload приложения полностью в K8S
- MR, созданный в рамках пайплайна, принят и замёржен

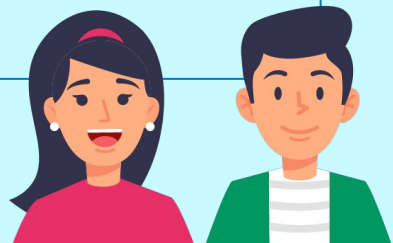
## Ссылки

1. Подробное описание пайплайна: <https://confluence.skyeng.tech/pages/viewpage.action?pagelId=148846585>
2. Видео-демонстрация работы с пайплайном: <https://drive.google.com/file/d/1eGQZtKN4kJI5Az-O1qRC1XI5JJshNSdQ/view?usp=sharing>
3. [Видео](#) о миграции в K8S с `#dev-talks` и [презентация](#)
4. Инструменты разработчика в K8s: <https://confluence.skyeng.tech/pages/viewpage.action?pagelId=148847570>

# ОЖИДАНИЯ

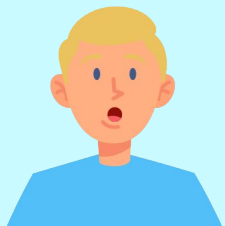


поставили задачу

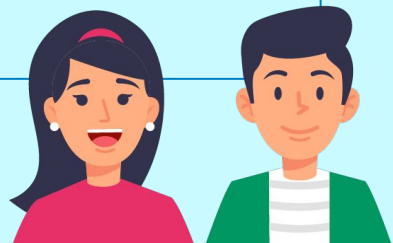


ребята на следующей неделе  
взяли в работу

# ОЖИДАНИЯ

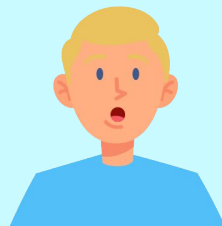


поставили задачу

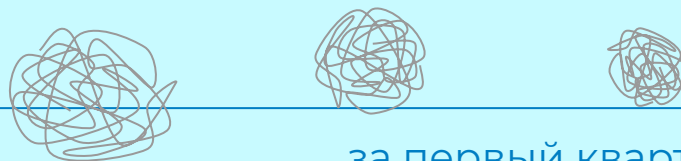


ребята на следующей неделе  
взяли в работу

# РЕАЛЬНОСТЬ



поставили задачу



за первый квартал  
почти никто не взял задачи в работу

# РЕАЛЬНОСТЬ



# РЕАЛЬНОСТЬ. ПРОЕКТ

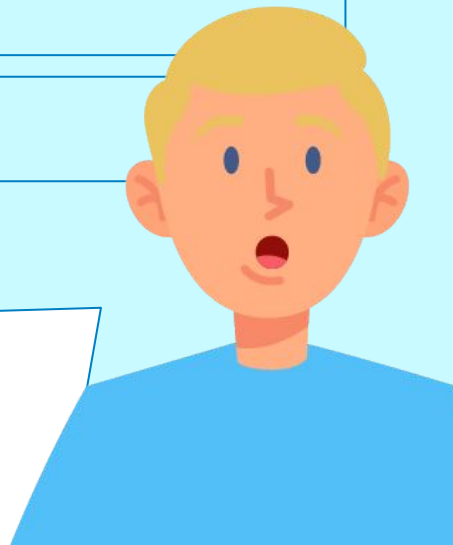
точно **спрогнозировать** невозможно

**метрики** не собирались

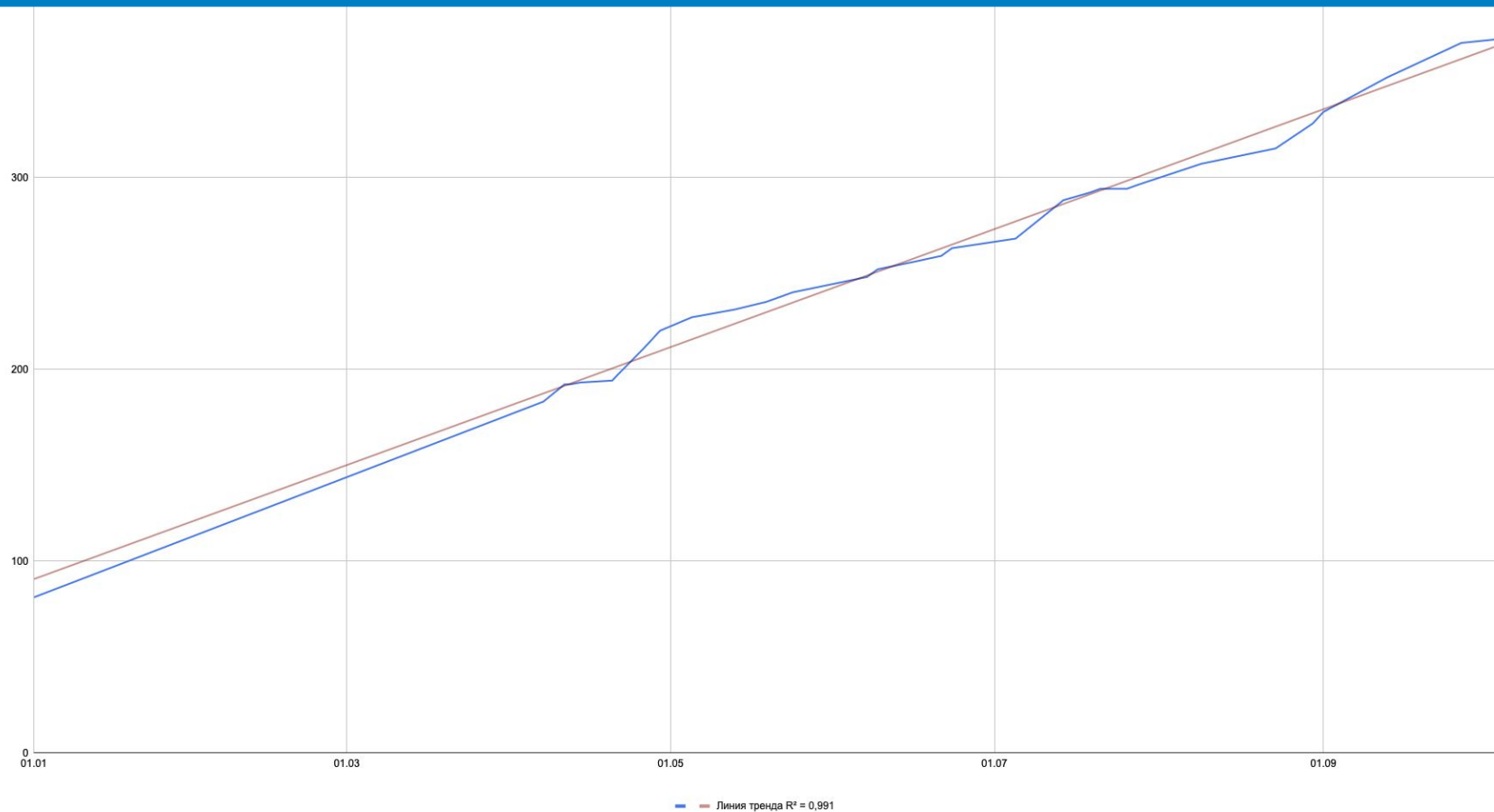
начало было на **личных** договоренностях

до **марта** создавали новые проекты в сворме

хотели мигрировать за полгода,  
и каждые полгода хотели за полгода



# скорость миграции



# РЕАЛЬНОСТЬ. ПРОЕКТ. ЧТО ДЕЛАТЬ:

планируйте **маленькими целями**, не старайтесь охватить сразу все

это хороший проект для **формирования команды**,  
большой и тяжелый

собирайте **метрики** с самого начала,  
даже если они будут потом пересмотрены

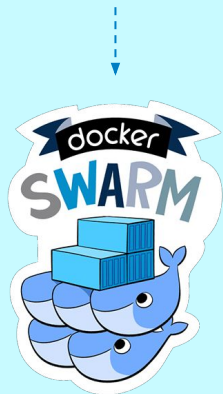
аллоцируйте **больше времени** на проект, забудьте о других активностях

не забывайте **брать перерыв**, чтобы не потонуть в рутине  
не забывайте, что **проекты могут создаваться**

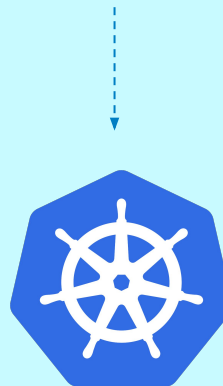
# РЕАЛЬНОСТЬ. КОНВЕРТАЦИЯ



Docker-compose  
Scheme



Skyeng-project-dsl  
Scheme

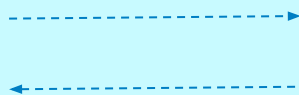
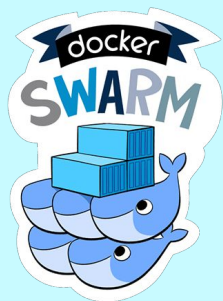




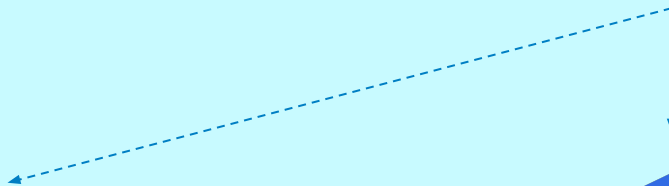
# РЕАЛЬНОСТЬ. КОНВЕРТАЦИЯ



Docker-compose  
Scheme



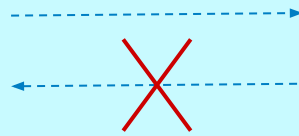
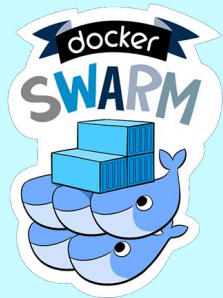
Skyeng-project-dsl  
Scheme



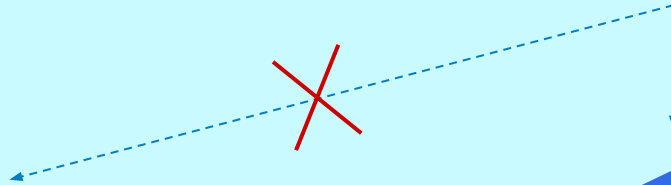
# РЕАЛЬНОСТЬ. КОНВЕРТАЦИЯ



Docker-compose  
Scheme



Skyeng-project-dsl  
Scheme



# РЕАЛЬНОСТЬ. КОНВЕРТАЦИЯ

разрабатывали конвертацию **2 недели**

доводили до ума конвертацию примерно **8 месяцев**

очень много было **непонятных решений** на ранних этапах

**обратная конвертация** для проверки прямой конвертации

поддержка деплоя из **DSL в docker-swarm**

это  
нормально

# РЕАЛЬНОСТЬ. КОНВЕРТАЦИЯ

разрабатывали конвертацию 2 недели

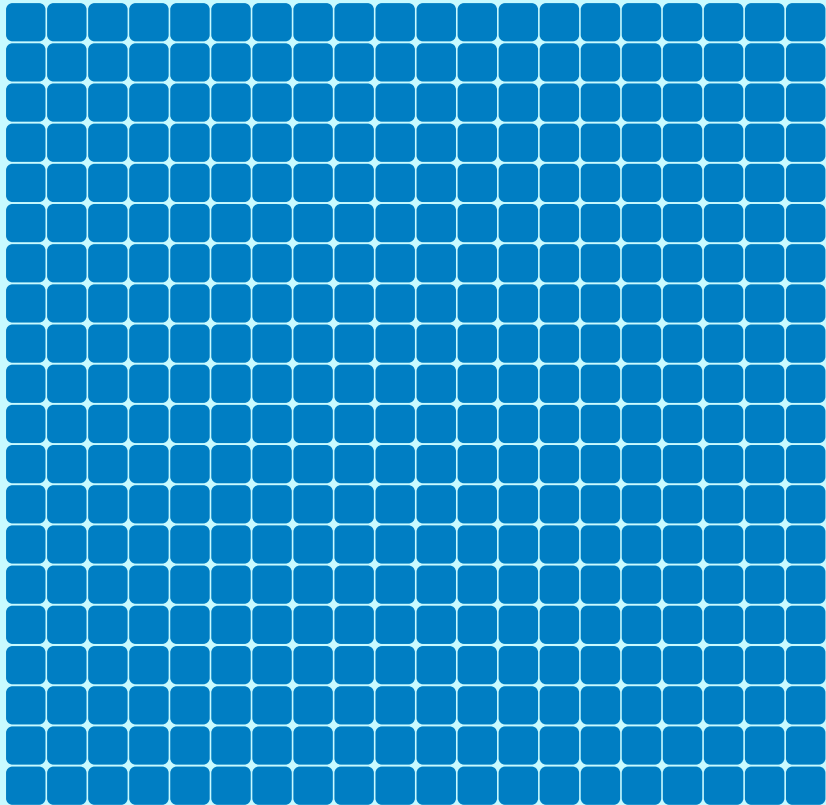
доводили до ума конвертацию примерно 8 месяцев



берите **понятные** вам DoD'ы

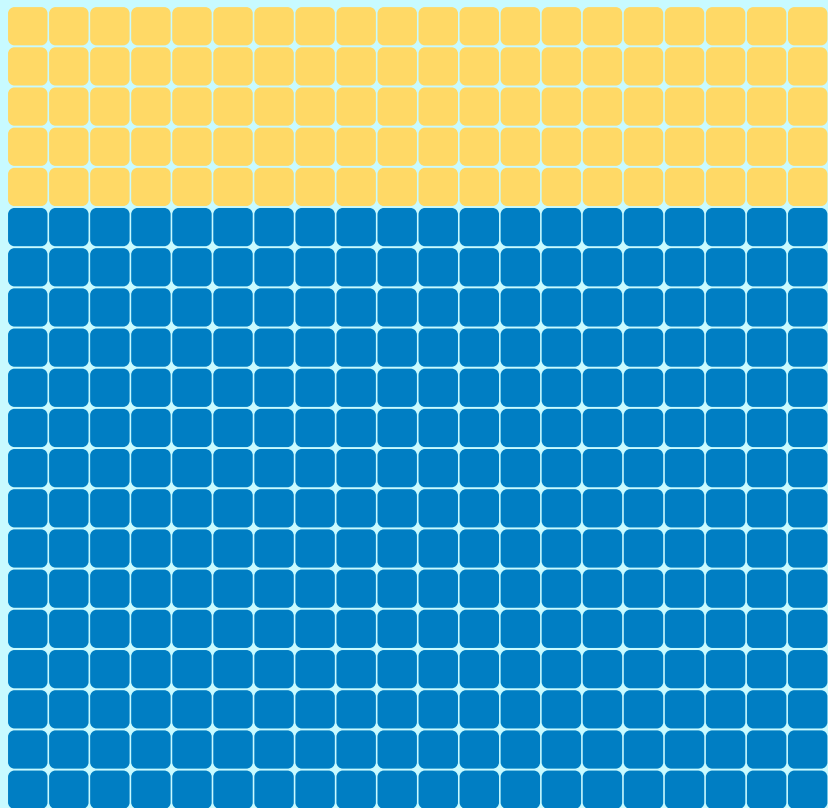
ищите решения **без создания** новых решений сначала

# РЕАЛЬНОСТЬ. КАЧЕСТВО ПРОЕКТОВ



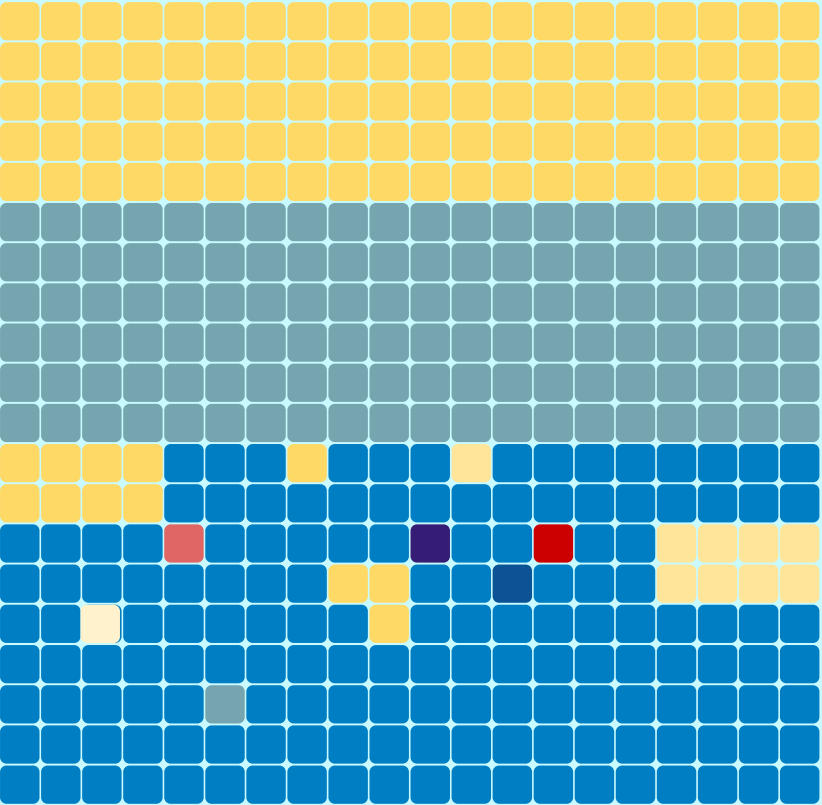
а на самом деле ... хаотичные  
сервисы, типовых на самом деле  
около половины или даже меньше

# РЕАЛЬНОСТЬ. КАЧЕСТВО ПРОЕКТОВ



а на самом деле ... хаотичные  
сервисы, типовых на самом деле  
около половины или даже меньше

# РЕАЛЬНОСТЬ. КАЧЕСТВО ПРОЕКТОВ



← - - - фронт

← - - - бэк на php

← - - - java, nodejs, монорепы,  
кастомные проекты ...

# РЕАЛЬНОСТЬ. Анализ 80 проектов

**11** проектов не подходят в диапазон портов (8000-11000)

**37** проектов имеют ошибки

**19** требуют ручного анализа

**13** успешно прошли автотесты

**16 %** проектов могут быть перевезены автоматически (64/400 проекта)



Error class	count	Error	count	Runbook	top 10 sum
none	163	replicas 0	7		32
wrong port	0	???	6		
extra volume	0	var/projects doesn't exists	5		
code error	7	408	2		
skydeploy error	19	500	2		
Dockerfile error	2	DB Connection refused	2		
k8s deploy error	0	SSLException: httpCheck follows redirect	2		
insight fault	1	composer2 incompatibility	2		
other	8	logs:/opt/app/var/logs'	2		
docker-compose	0	validation errors for Model	2		
???	7	502	1		
Composer	5	Error	1		
Env vars	1	KeyError: 'ports'	1		
MultiUpstream	35	No such library resource	1		
		Secret not found	1		
		assert len(self._loc_config.config_files) == 1	1		
		composer install failed	1		
		dep not found	1		
		env var not passed	1		
		get property 'jira_project' on null object	1		
		git clone	1		
		invalid pipeline config link	1		
		key not found: "APP_ENV"	1		
		needs composer1	1		
		npm install errors	1		
		port change	1		
		python: AttributeError: 'NoneType' object has no attribute 'copy'	1		
		python: assert len(self._loc_config.config_files) == 1	1		
		testing deleted	1		
		wrong project	1		

# РЕАЛЬНОСТЬ. Ошибки

ошибки **прошлого** (порты, подходы, рудименты)

**оригинальность** приложений

ошибка в **сторонних** сервисах

приложение морально **мертво**

# РЕАЛЬНОСТЬ. Ошибки. Что делать

стандартизируйте

ограничивайте

не запускайте **техдолг**

делай **лучше**, чем норм (правило 99% vs 101%)

can I do better?

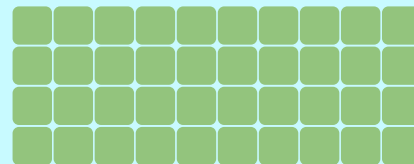
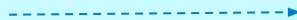
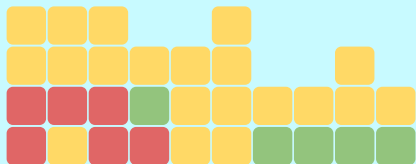
this is the best what I can do now?

# РЕАЛЬНОСТЬ. Командная работа

это работа **ВСЕЙ** компании

ваша **цель завоевать доверие** людей

ваша вторая **цель не потерять** завоеванное **доверие**



# шел 12 месяц миграции в k8s

Виталик, я всё ещё в шоке от того, как быстро и классно мы вчера переехали. Я реально ожидал какой-то ручной работы и растягивания на неделю, не меньше. Но ты всё сделал настолько быстро, я даже чувствовал, что это я торможу процесс, тем что не сразу отвечаю 😊. В общем, если ты там собираешь NPS, это 20 из 10.

Спасибо 🗨️



# РЕАЛЬНОСТЬ. Командная работа

если у команды **нет приоритетов** в миграции – **миграции не будет**

пришли в команду в **мае**, мигрировали в **октябре**

проблема **ресурсов** и конечно же **кастомщины**

**фантомные** и **реальные** проблемы

приходите **раньше**, объясняйте **раньше**, эскалируйте **раньше**

# РЕАЛЬНОСТЬ. Подходы

Q4'21

однотипные **простые** проекты (фронт)

Q1'22

любые **ненагруженные** проекты

Q2'22

самые **энергостоящие** проекты

Q3'22

самые **важные** проекты (+ самые кастомные/оставшиеся)

Q4'21

надо все потестить, подготовиться, проверить

Q3'22

давайте мы сами все сделаем, а с вас контрольные тесты

# ВЫВОДЫ





# Выводы. В чем профит?

- сэкономили множество средств
- стандартизировали большой пласт инфраструктуры
- получили гибкую инфраструктуру
- получили широкий спектр фич и улучшений на будущее
- сформировали хорошую проектную команду
- увидели воочию КАЖДЫЙ проект skyeng'a
- набрались множества опыта



# Выводы. В чем профит?

- улучшили опыт разработки devspace
- улучшили опыт разработки k8s в целом
- ускорили деплои и откаты
- добавили прозрачности в деплои  
(нормальные логи и ошибки)



# БУДУЩЕЕ



# Будущее

канарейка

namespace as a service

ingress

P.S. Сейчас сложно планировать далеко в будущее

# ЭПОПЕЯ МИГРАЦИИ ~400 ПРОЕКТОВ ИЗ SWARM В K8S: ЧЕМУ ОНА НАУЧИЛА

Андрей Важенин

 skyeng

[github.com/ag0n1k](https://github.com/ag0n1k)

