

# Ок, бумер! Вот тебе Cloud-Native

Федор Чемашкин

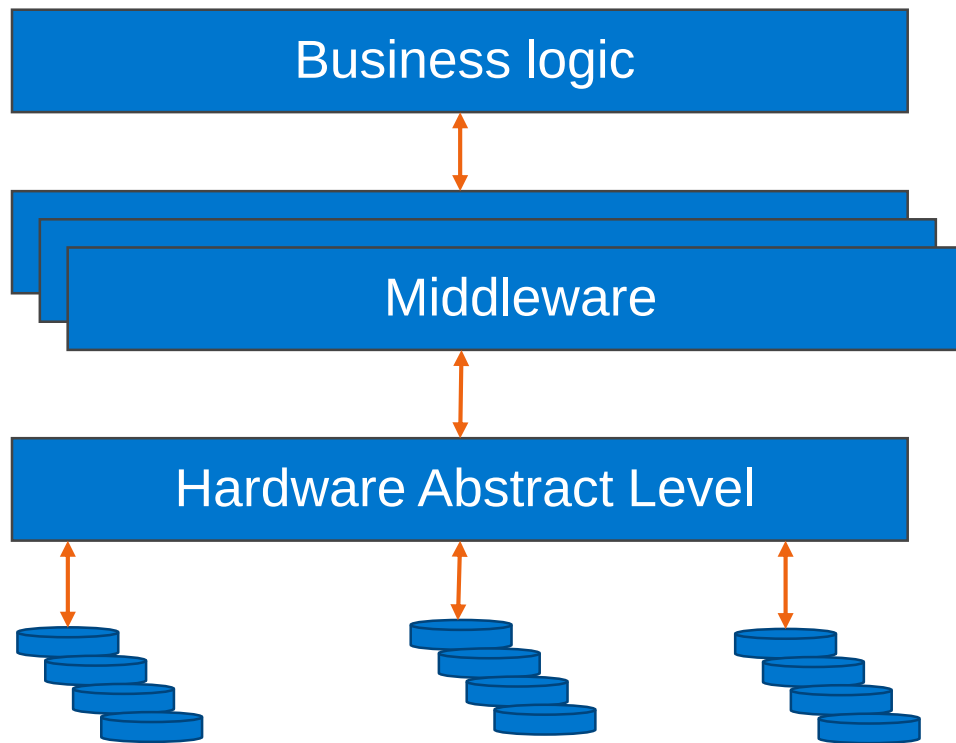
Fedor.Chemashkin@Dell.com  
@fed\_ch

**DELL**Technologies

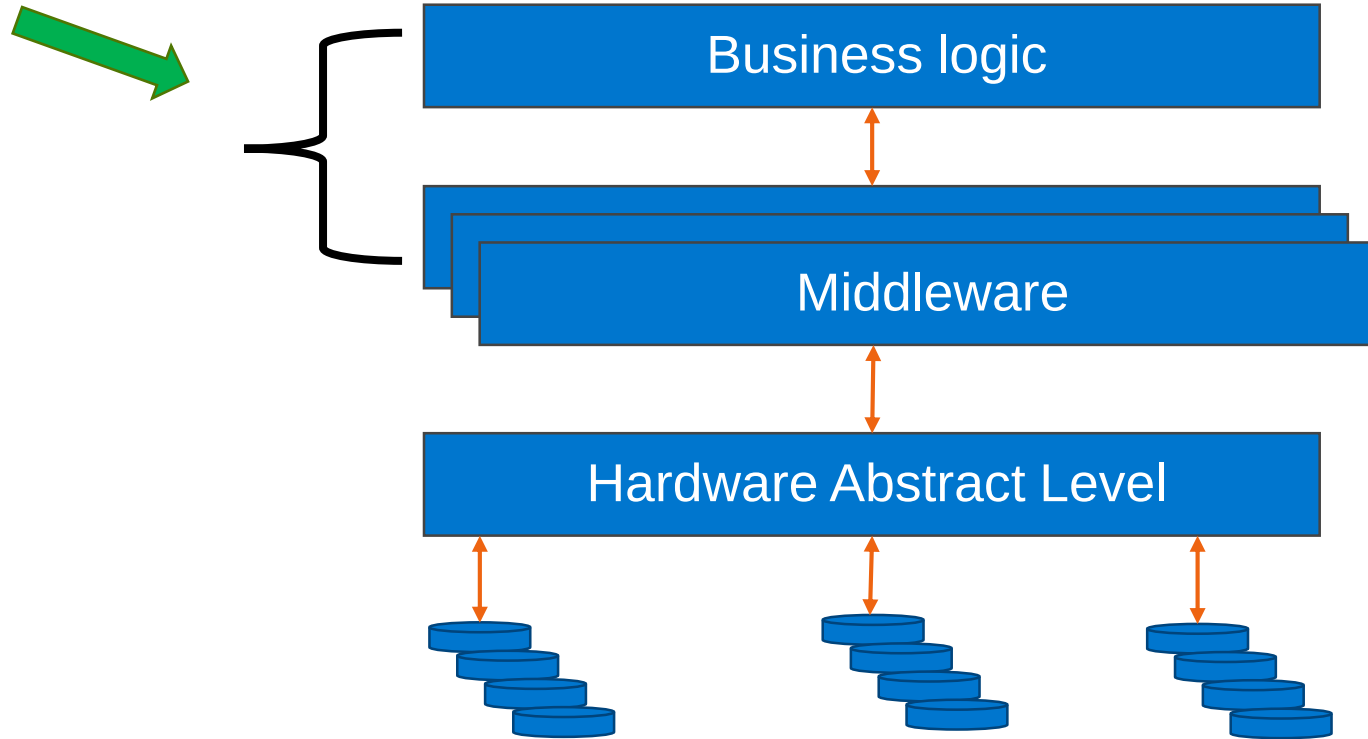
# О чем поговорим

- Что такое Cloud-Native
- Без чего это невозможно
- Как мы становились Cloud-Native
- Почему теория облегчает жизнь
- Внедрение DevOps практик снизу

# Типичный Storage



# Типичный Storage



# Обо мне

- Senior Software Engineer
- Dell EMC
- ECS
- Верю в DevOps != Engineer





# 7R

# 7R

## Replace

- Replace application with SaaS service
- Build cloud native application with similar/improved characteristics and features

Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in *IEEE Cloud Computing*, vol. 4, no. 5.



# 7R

## Replace

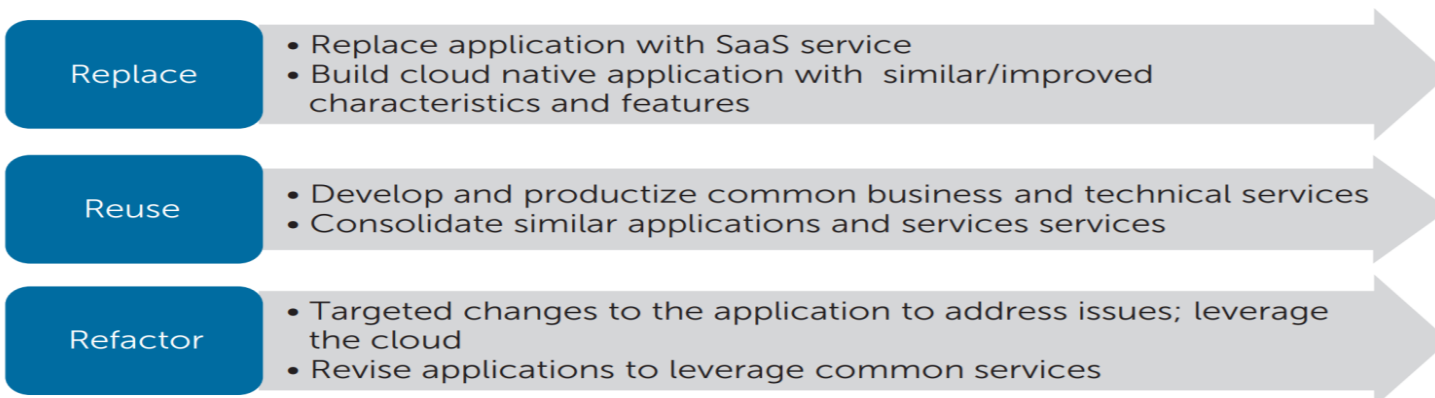
- Replace application with SaaS service
- Build cloud native application with similar/improved characteristics and features

## Reuse

- Develop and productize common business and technical services
- Consolidate similar applications and services services

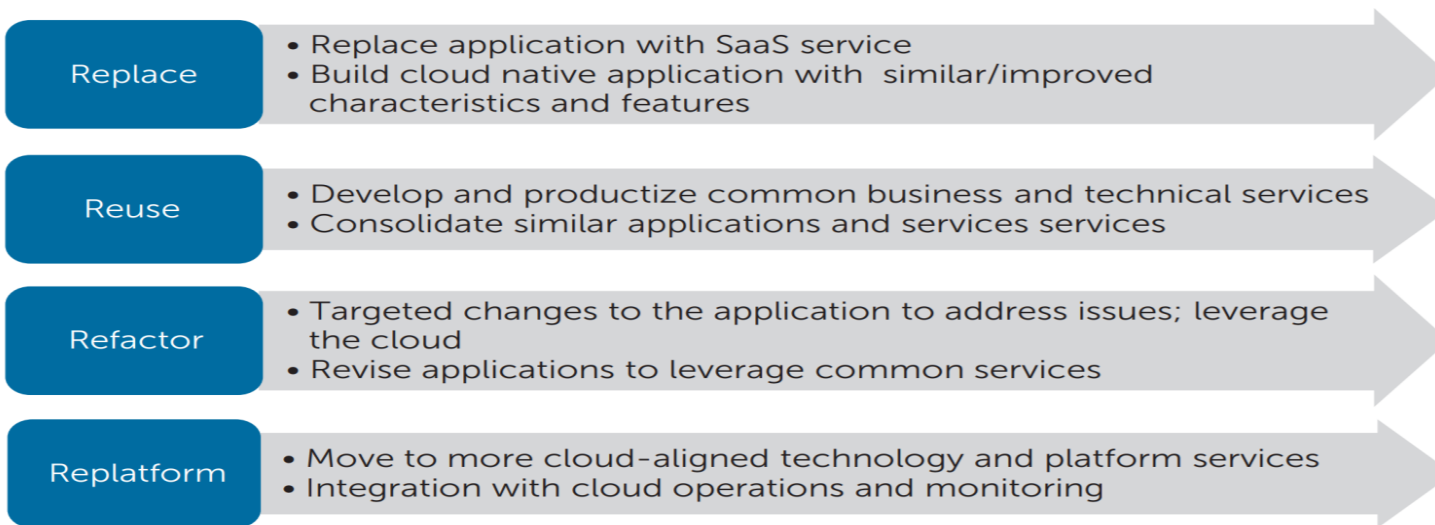
Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in *IEEE Cloud Computing*, vol. 4, no. 5.

# 7R



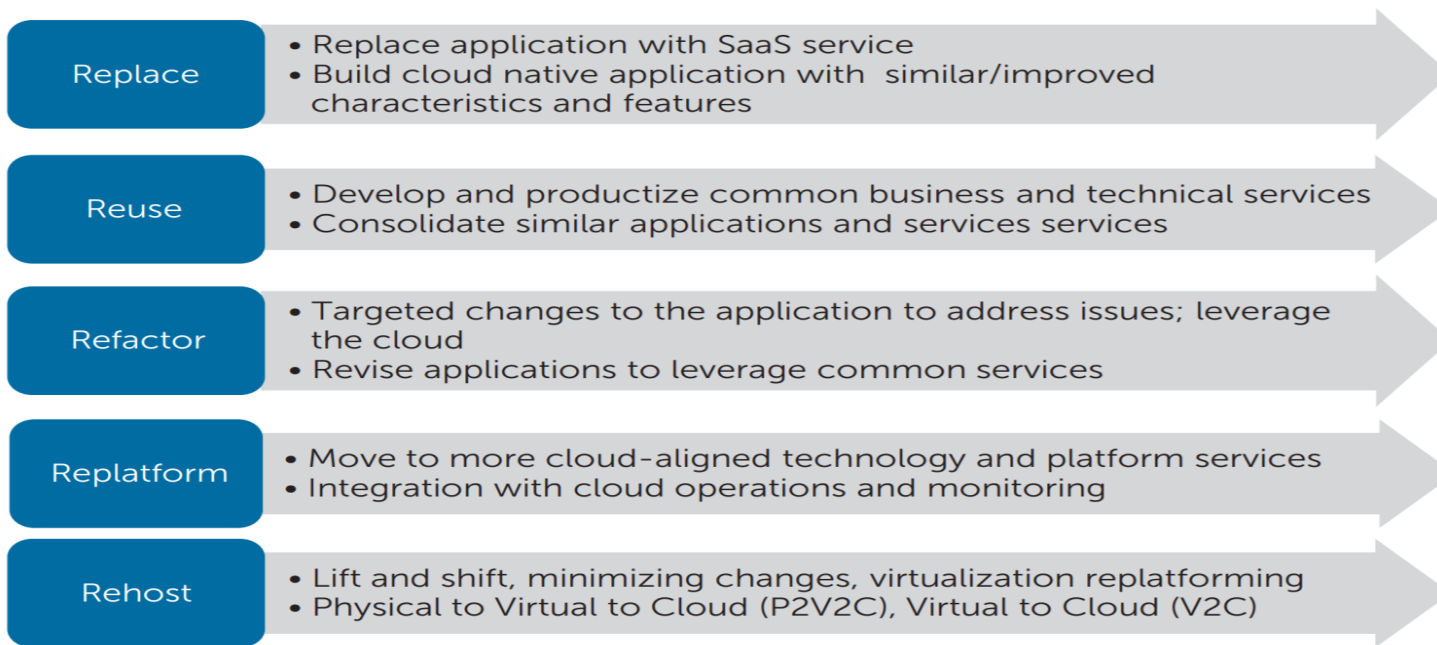
Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in *IEEE Cloud Computing*, vol. 4, no. 5.

# 7R



Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in *IEEE Cloud Computing*, vol. 4, no. 5.

# 7R



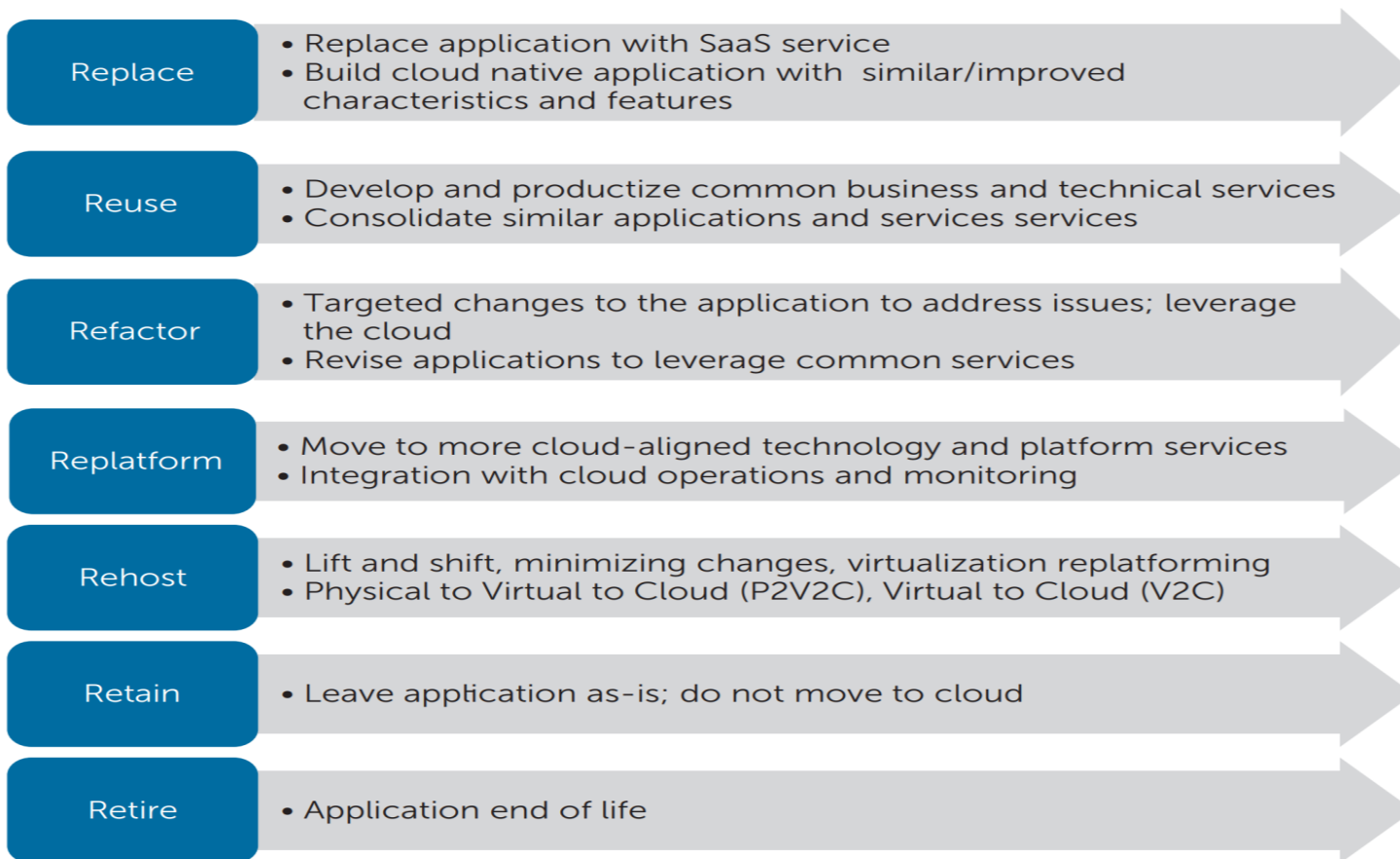
Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in *IEEE Cloud Computing*, vol. 4, no. 5.

# 7R



Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in IEEE Cloud Computing, vol. 4, no. 5.

# 7R



Источник: D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in IEEE Cloud Computing, vol. 4, no. 5.

**Когда перенес в Cloud**



# Кесарю кесарево – Cloud'у cloud'ное

## Индустрия



**CLOUD NATIVE**  
COMPUTING FOUNDATION

## Научное сообщество





# CNCF – Technical Oversight Committee

- Определение технического видения CNCF
- Принятие новых проектов в CNCF
- Определение общих практик для проектов CNFC
- [github.com/cncf/toc](https://github.com/cncf/toc)

# CNCF - определение Cloud-Native

<https://github.com/cncf/toc/blob/master/DEFINITION.md>

- Динамические среды
- Контейнеры, сервисные сита, микросервисы, неизменяемая инфраструктура, и т.д.
- Надежная автоматизация
- Значительные изменения с минимальными усилиями

# Интересная теория



# Научное сообщество

## Критерии для отбора статьи

- Цитируемость статьи
- Индекс Хирша авторов
- Место публикации

# Научное сообщество

cloud native

Статьи

Результатов: примерно 1 210 000 (0,04 сек.)

Мой профиль

За все время

С 2020

С 2019

С 2016

Выбрать даты

По релевантности

По дате

☒ включая патенты

☒ показать цитаты

☐ Создать оповещение

**Microservices architecture enables devops: Migration to a **cloud-native** architecture**

[A Balalaie, A Heydarnoori, P Jamshidi](#) - [Ieee Software](#), 2016 - [ieeexplore.ieee.org](#)

This article reports on experiences and lessons learned during incremental migration and architectural refactoring of a commercial mobile back end as a service to microservices architecture. It explains how the researchers adopted DevOps and how this facilitated a ...

☆ Цитируется: 353 [Похожие статьи](#) [Все версии статьи \(9\)](#)

[\[PDF\] imperial.ac.uk](#)

**Migrating to **cloud-native** architectures using microservices: an experience report**

[A Balalaie, A Heydarnoori, P Jamshidi](#) - ... on [Service-Oriented and \*\*Cloud\*\*](#) ..., 2015 - [Springer](#)

Migration to the **cloud** has been a popular topic in industry and academia in recent years. Despite many benefits that the **cloud** presents, such as high availability and scalability, most of the on-premise application architectures are not ready to fully exploit the benefits of this ...

☆ Цитируется: 133 [Похожие статьи](#) [Все версии статьи \(9\)](#)

[\[PDF\] arxiv.org](#)

**Kubernetes and the path to **cloud native****

[EA Brewer](#) - [Proceedings of the Sixth ACM Symposium on \*\*Cloud\*\*](#) ..., 2015 - [dl.acm.org](#)

We are in the midst of an important shift to higher levels of abstraction than virtual machines. Kubernetes aims to simplify the deployment and management of services, including the construction of applications as sets of interacting but independent services. We explain ...

☆ Цитируется: 85 [Похожие статьи](#) [Все версии статьи \(2\)](#)

**Understanding **cloud-native** applications after 10 years of **cloud** computing-a systematic mapping study**

[N Kratzke, PC Quint](#) - [Journal of Systems and Software](#), 2017 - [Elsevier](#)

It is common sense that **cloud-native** applications (CNA) are intentionally designed for the **cloud**. Although this understanding can be broadly used it does not guide and explain what a **cloud-native** application exactly is. The term "**cloud-native**" was used quite frequently in ...

☆ Цитируется: 89 [Похожие статьи](#) [Все версии статьи \(3\)](#)

[\[PDF\] researchgate.net](#)

# Научное сообщество – Cloud-Native Application

- Распределенное
- Масштабируемое
- Микросервисное
- Динамически управляемое

# Научное сообщество – Cloud-Native Architecture

## Принципы

- Сервис-ориентированность
- Виртуализация
- Неопределенность
- Адаптивность



## Паттерны

- Микросервисы
- Динамические модели
- Контроллеры с обратной связью

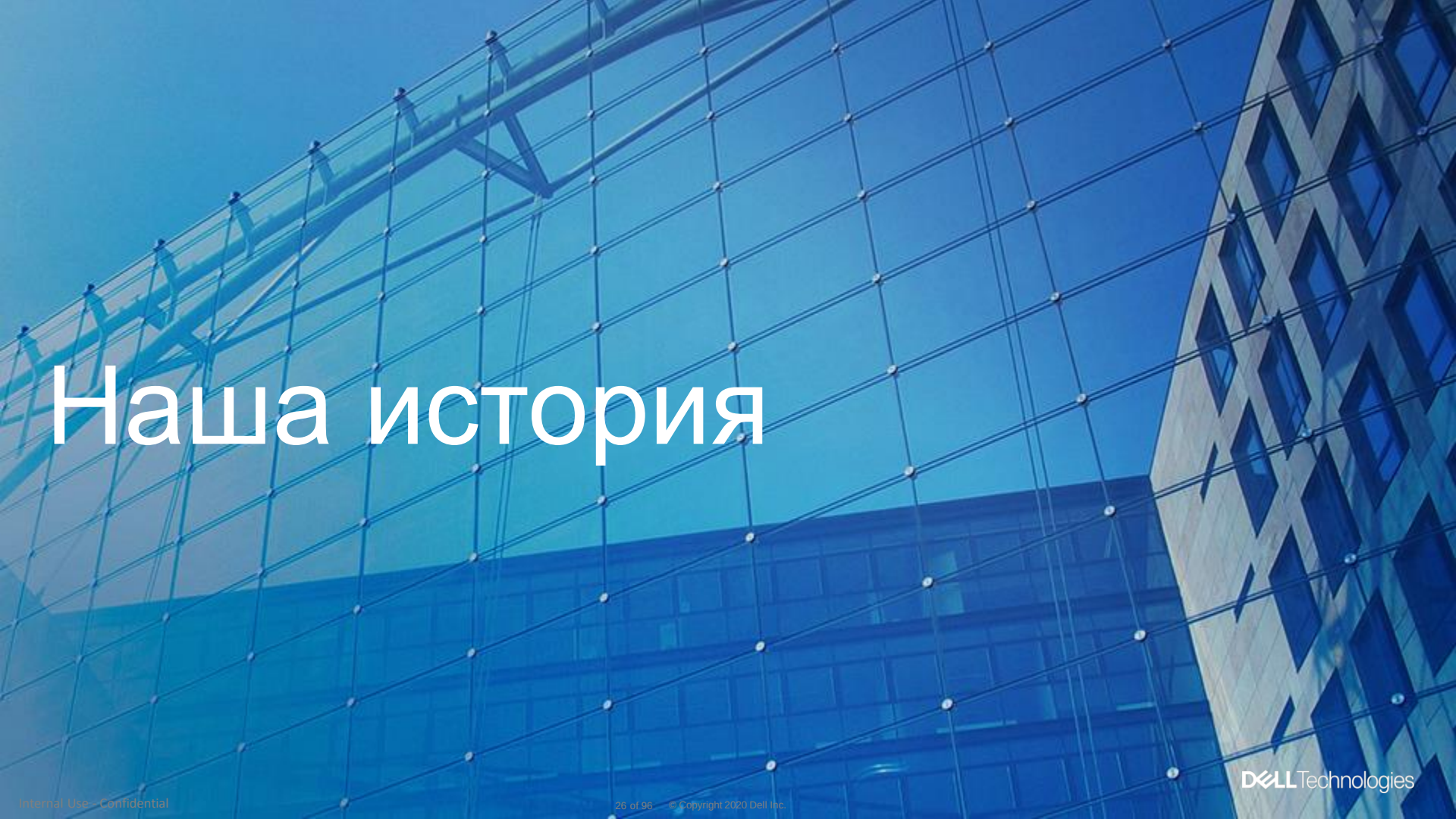
# Общая мысль, а для нас главная



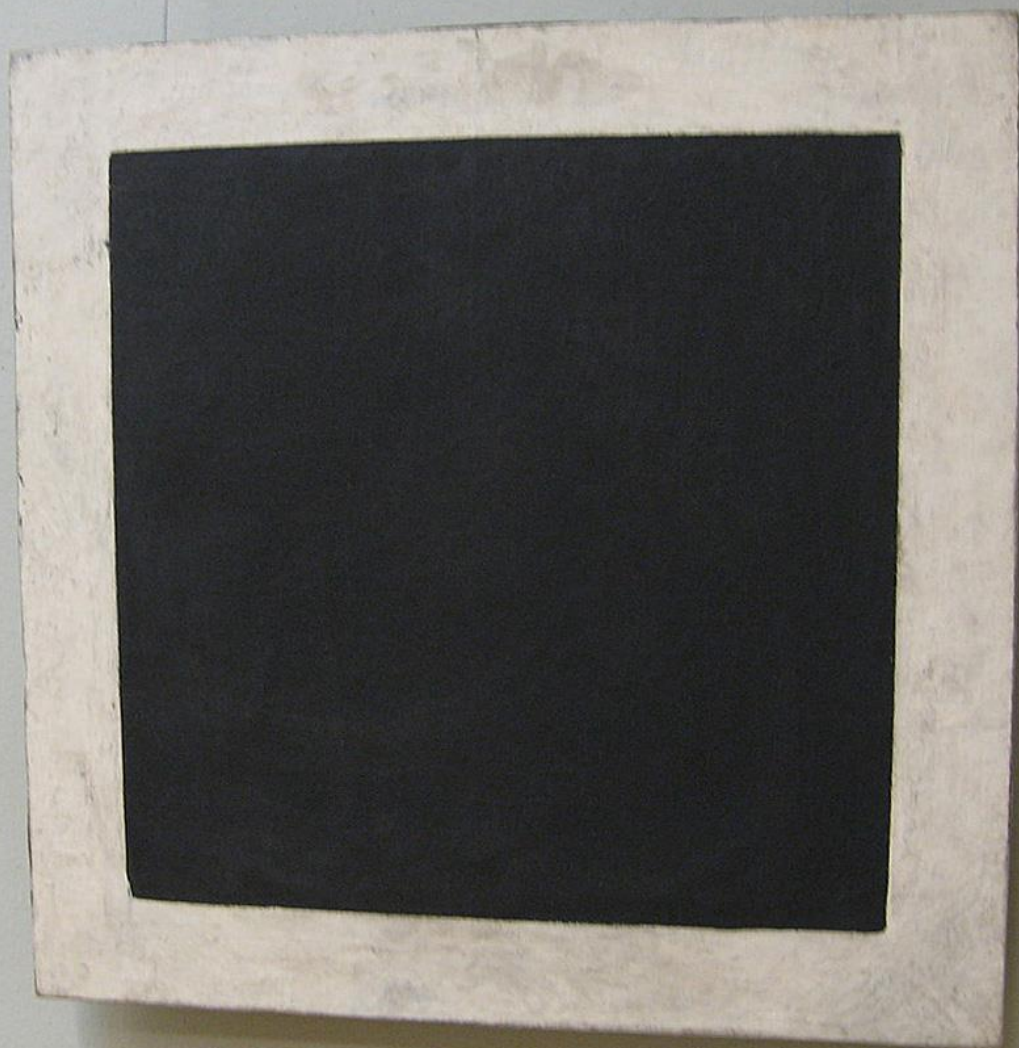
Общая мысль, а для нас главная



**Без DevOps –  
Нет Cloud-Native**

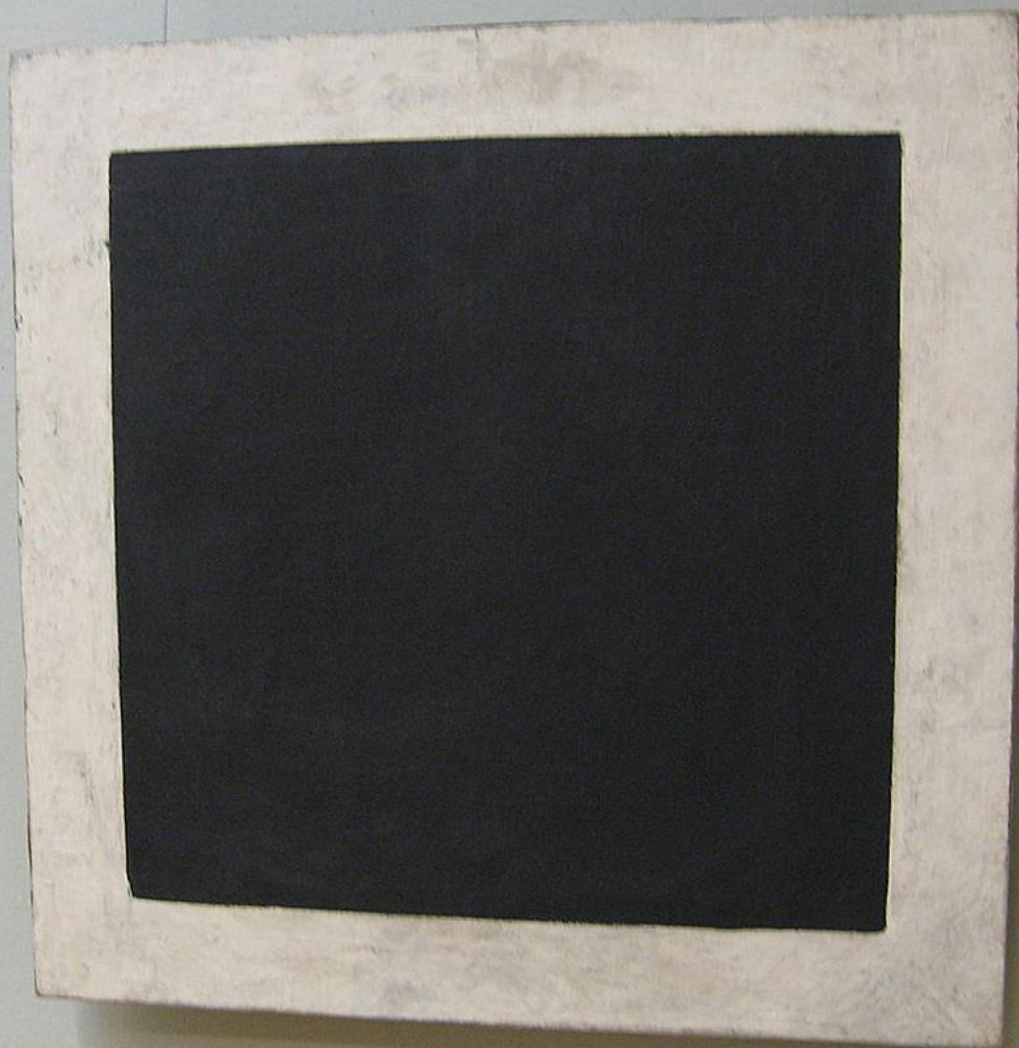


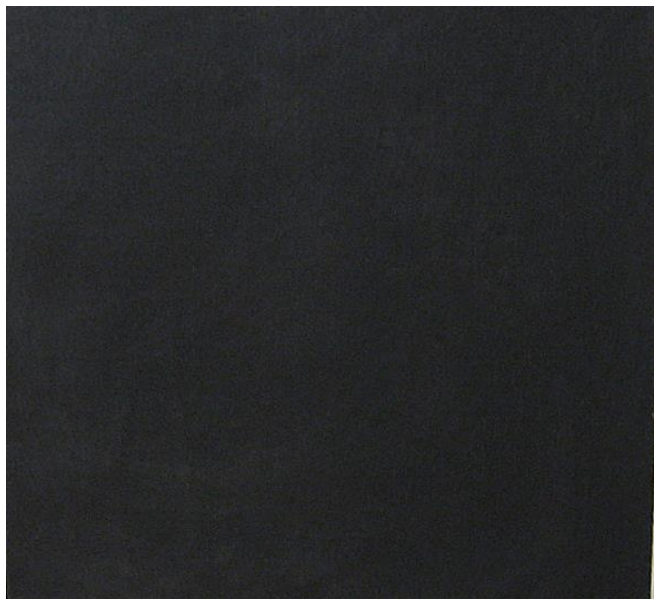
# Наша история





**Монолит**





**Рефакторинг**



**Распределенная  
команда**



# Маленький шаг от монолита...

- Отдельные Dockerfile
- Отдельные Makefile с одинаковыми таргетами
- Отдельные Jenkinsfile
- У каждого свой entrypoint



«Шаблон»

# entypoint.sh

```
#!/bin/sh
```

```
PROCESS_NAME='process'
```

```
...
```

```
trap '_on_sigterm ${PROCESS_NAME} ${SERVICE_PID}' SIGTERM
```

```
# Start services
```

```
/bin/... file:/.../conf/...xml &
```

```
SERVICE_PID=$!
```

```
if [ $LOG_DIRECTION == 'stdout' ]; then
```

```
    ln -sf /dev/stdout /.../logs/${PROCESS_NAME}.log
```

```
fi
```

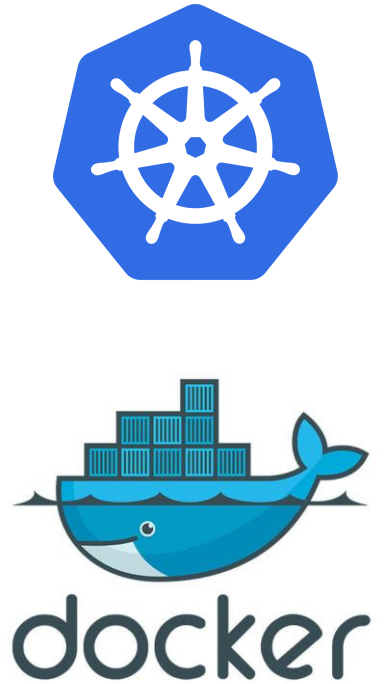
```
wait "$SERVICE_PID"
```

# Поддержка со стороны Automation команды

```
'component-master-build': [  
  // script  
  scriptPath: '.../build.Jenkinsfile',  
  // parameters  
  branchName      : 'master'  
  ...  
  // flags & arguments  
  repo            : 'git'  
  refspec         : '',  
  // generator  
  quietPeriod     : 0,  
  concurrentBuild: false,  
  logRotator      : ['90', '30', '14', '30'],  
  disabled        : false,  
  trigger         : [],  
]
```

[http://bit.ly/job\\_dsl](http://bit.ly/job_dsl)





# 12 факторов

[12factor.net/ru/](https://12factor.net/ru/)

1. Кодовая база
2. Зависимости
3. Конфигурация
4. Сторонние службы
5. Сборка, релиз, выполнение
6. Процессы
7. Привязка портов
8. Параллелизм
9. Утилизируемость
10. Паритет разработки/работы приложения
11. Журналирование
12. Задачи администрирования

# 12 факторов

[12factor.net/ru/](https://12factor.net/ru/)

1. Кодовая база

2. Зависимости

3. Конфигурация

4. Сторонние службы

5. Сборка, релиз,

6. Процессы



портов

ЗМ

МОСТЬ

работки/работы

ание

министрирования

# 12 факторов

[12factor.net/ru/](https://12factor.net/ru/)

1. Кодовая база

2. Зависимости

3. Конфигурация

4. Сторонние службы

5. Сборка, релиз, выполнение

6. Процессы

■ - хорошо

■ - так себе

■ - есть куда расти

7. Привязка портов

8. Параллелизм

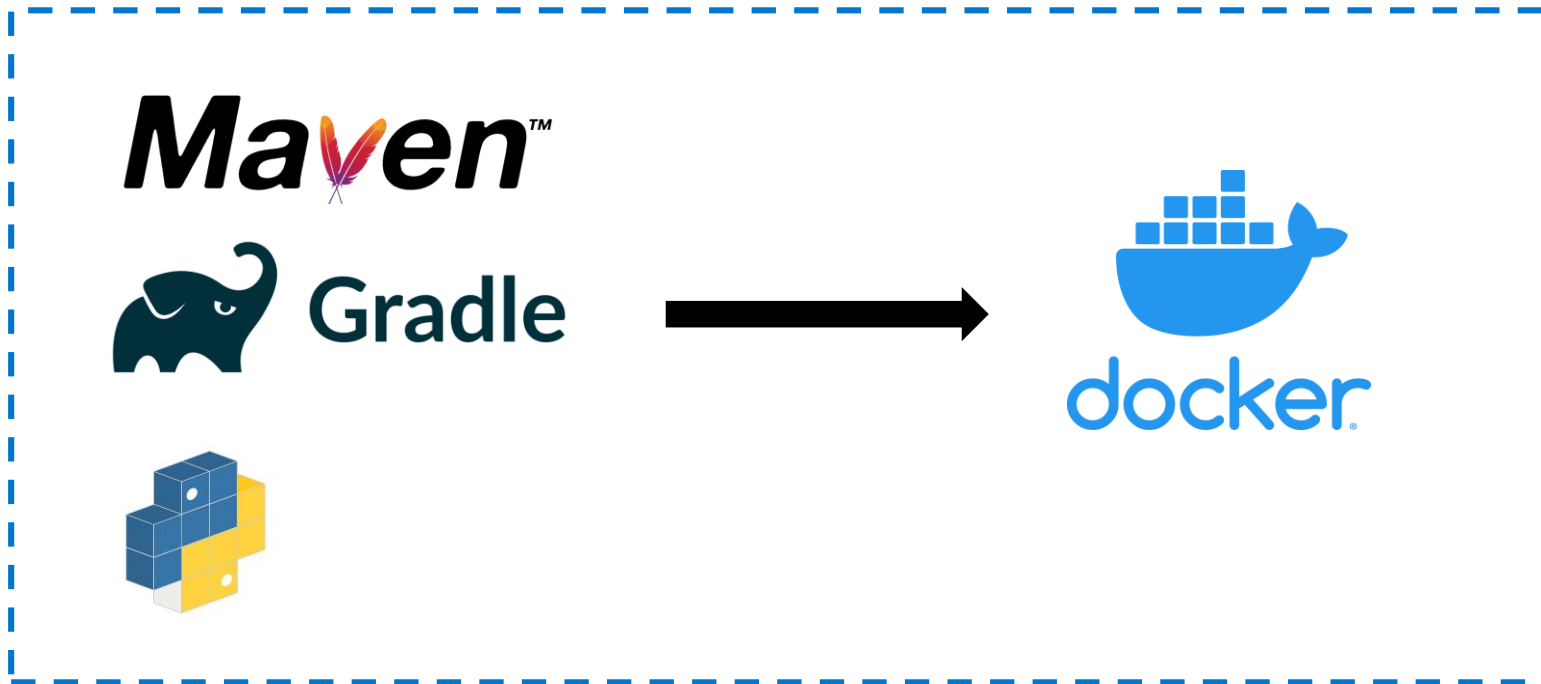
9. Утилизируемость

10. Паритет разработки/работы приложения

11. Журналирование

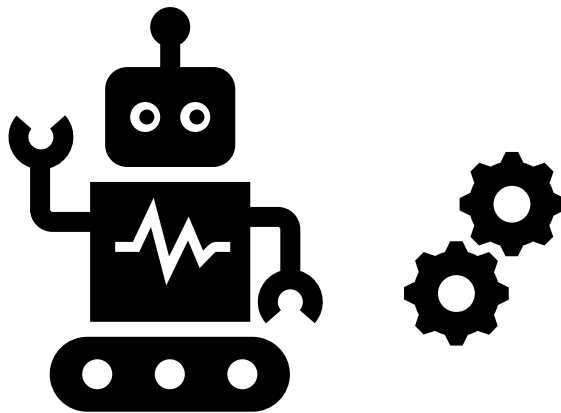
12. Задачи администрирования

## 2. Зависимости

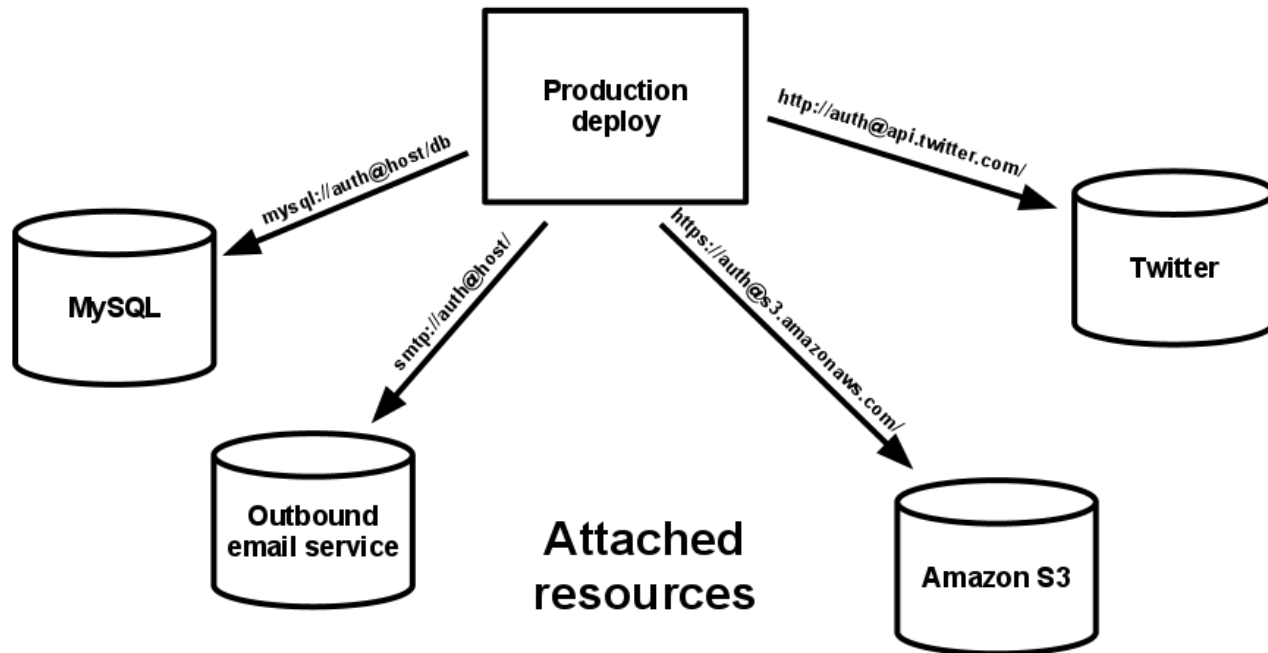


### 3. Конфигурации

Среда выполнения

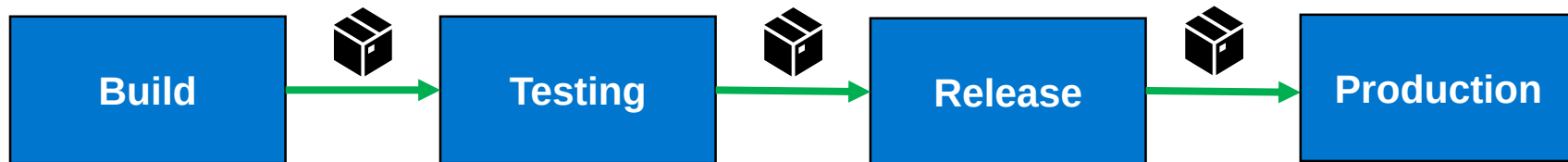


## 4. Сторонние службы



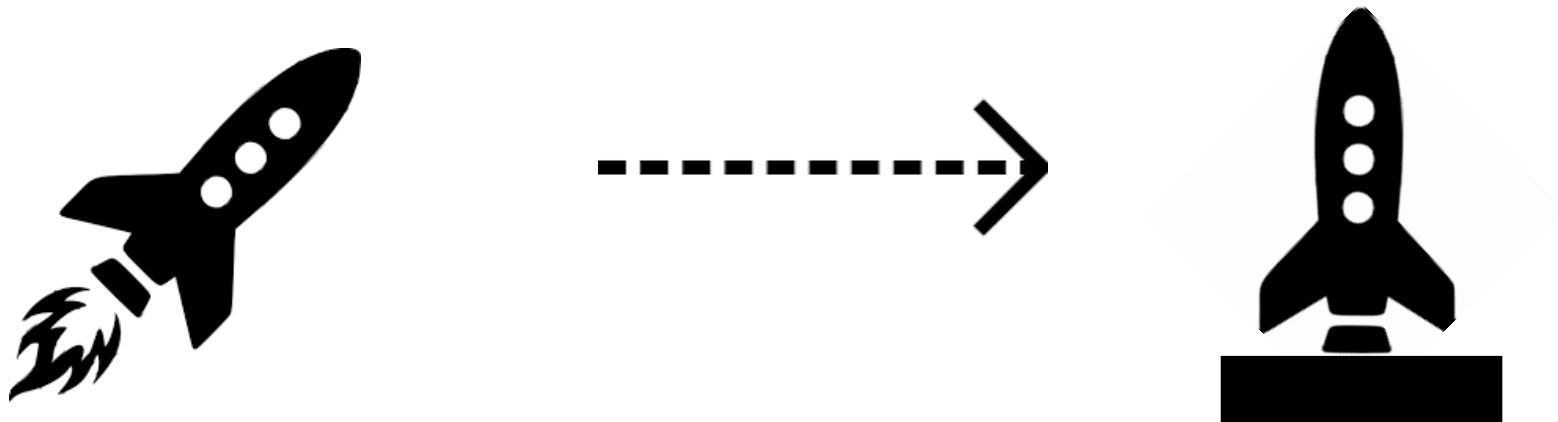
Google Cloud Platform

## 5. Сборка, релиз, выполнение

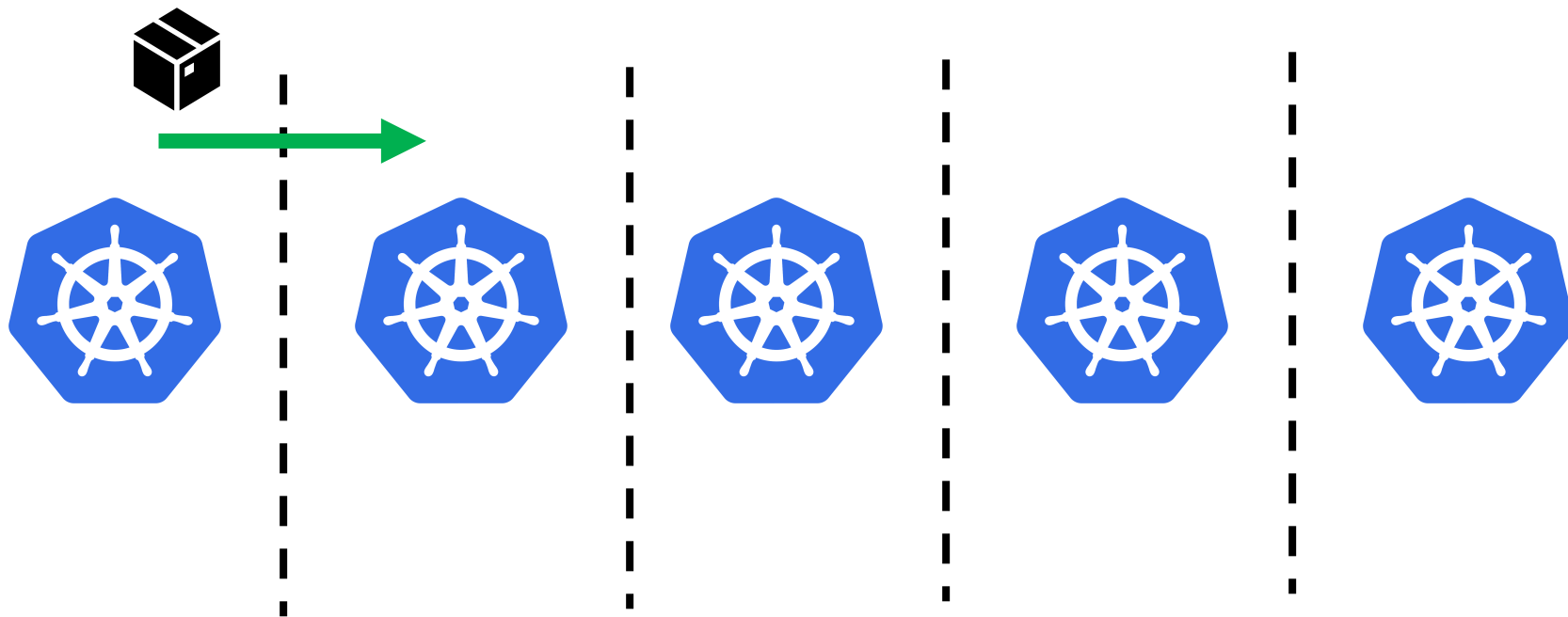




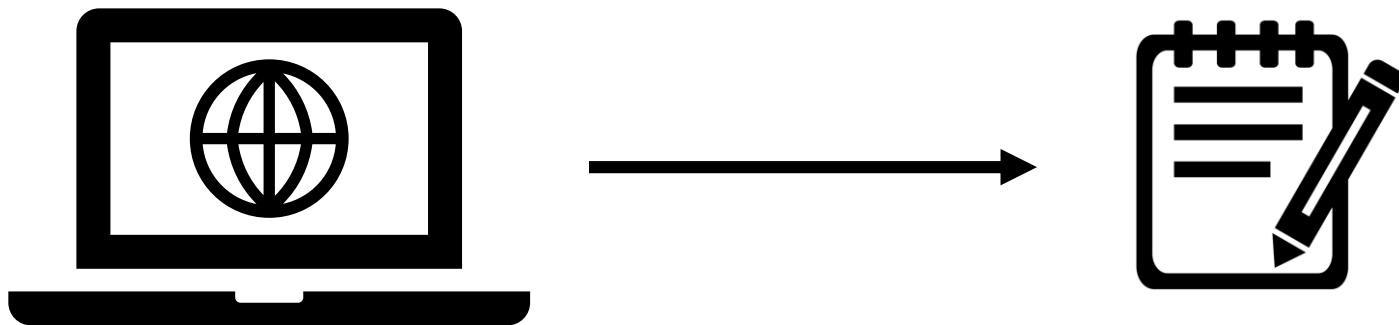
## 9. Утилизируемость



# 10. Паритет разработки/работы приложения



# 11. Журналирование



# 11. Журналирование

```
#!/bin/sh
```

```
PROCESS_NAME='process'
```

```
...
```

```
trap '_on_sigterm ${PROCESS_NAME} ${SERVICE_PID}' SIGTERM
```

```
# Start services
```

```
/bin/... file:/.../conf/...xml &
```

```
SERVICE_PID=$!
```

```
if [ $LOG_DIRECTION == 'stdout' ]; then
```

```
    ln -sf /dev/stdout /.../logs/${PROCESS_NAME}.log
```

```
fi
```

```
wait "$SERVICE_PID"
```

**ТОЛЬКО НЕ В ФАЙЛ**

**ТОЛЬКО НЕ В ФАЙЛ**



# Но и это не все

13. Наблюдаемость

14. Планируемость/прогнозируемость

15. Обновляемость

16. Минимум привилегий

17. Проверяемость

18. Безопасность

19. Измеримость

# Но и это не все

13. Наблюдаемость

14. Планируемость/прогнозируемость

15. Обновляемость

16. Минимум привилегий

17. Проверяемость

18. Безопасность

19. Измеримость

■ - хорошо

■ - так себе

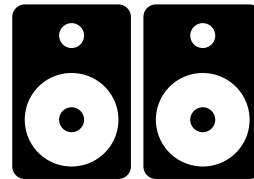
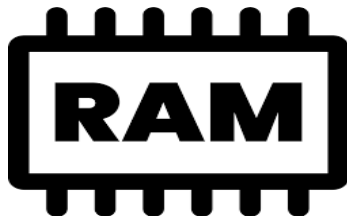
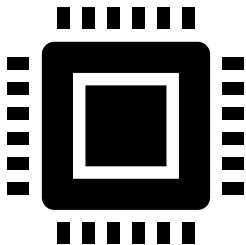
■ - есть куда расти

# 13. Наблюдаемость

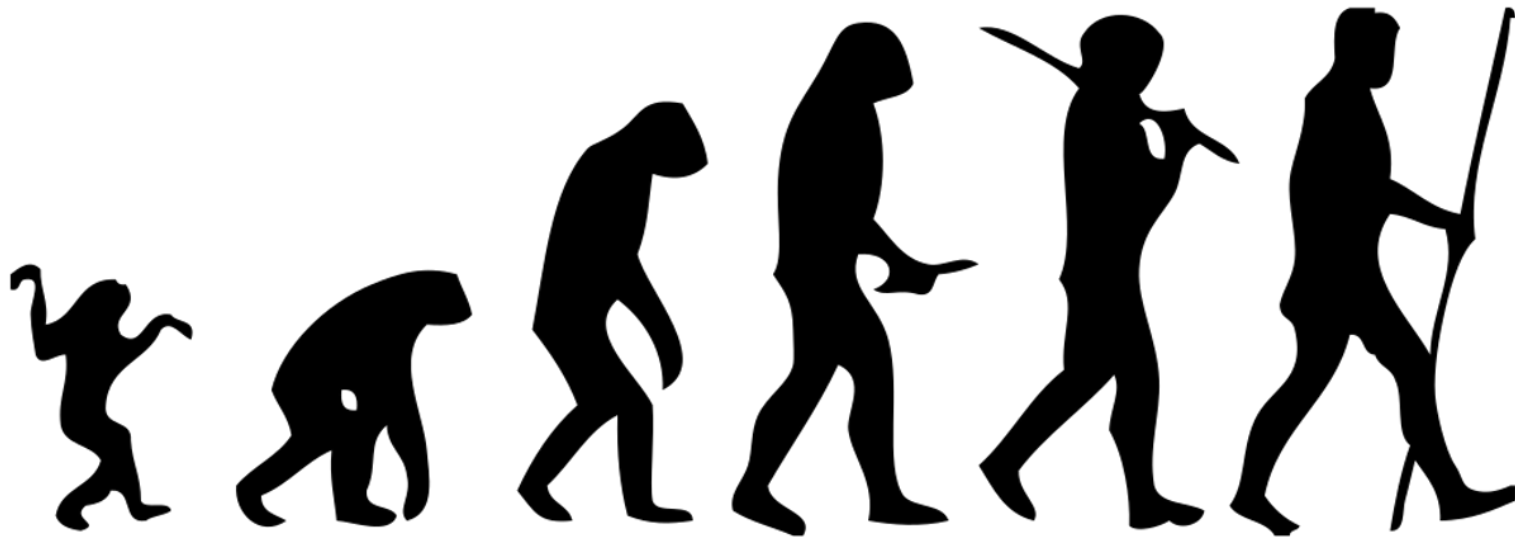




# 14. Планируемость/прогнозируемость



# 15. Обновляемость



# 17. Проверяемость

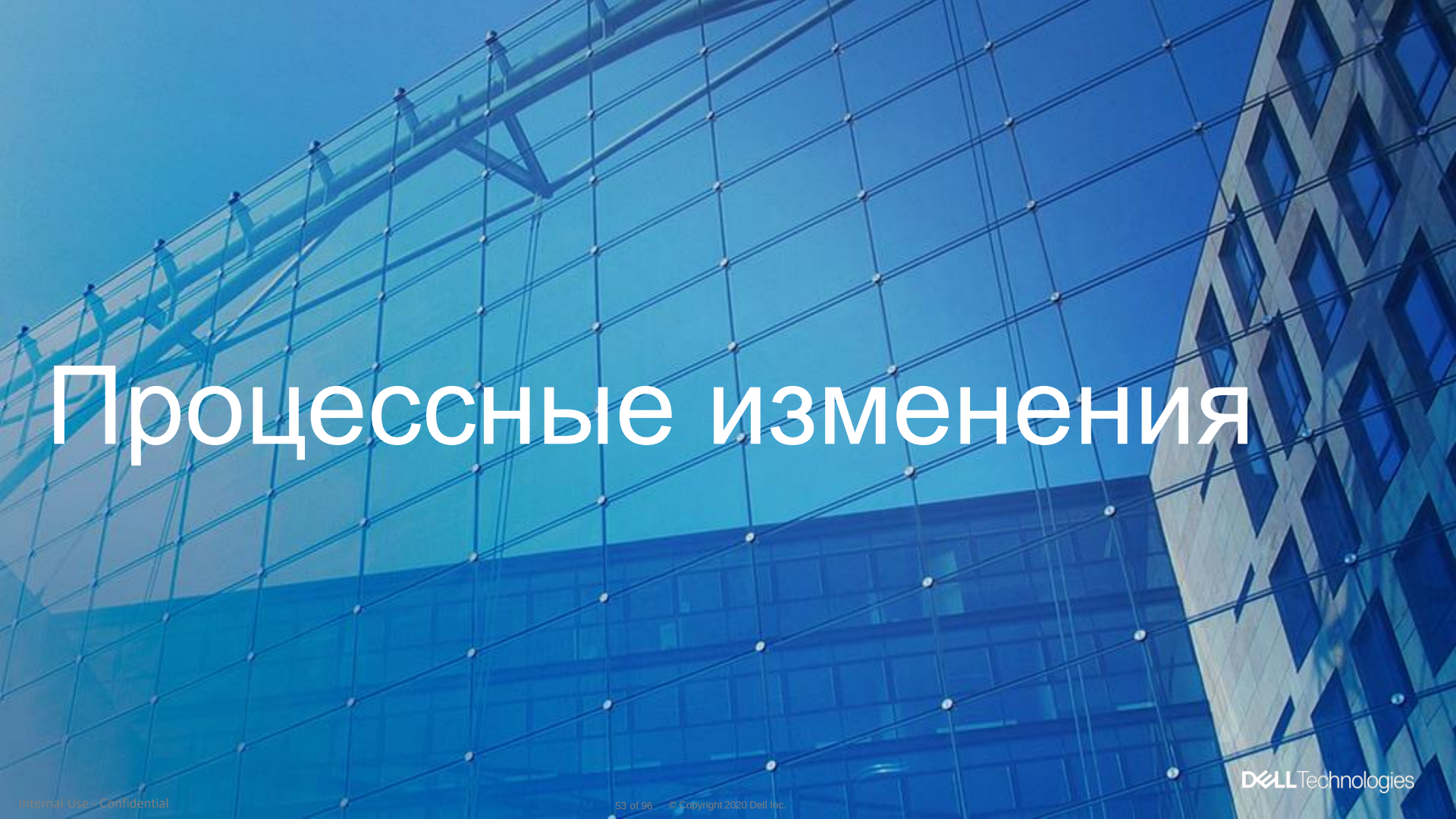


# Как мы сделали Cloud-Native Architecture

- Другие принципы и паттерны очевидны

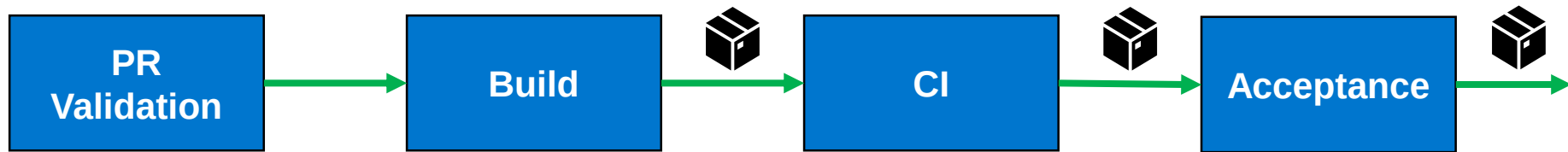


**Reconciliation loop**



# Процессные изменения

# Хотели сделать пайплайны



**НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ**

**И СДЕЛАТЬ DEVOPS**



## DevOps Boomer



## DevOps Zoomer





# Как бороться с «бумерами»?

# DevOps измеряем

- DevOps несет ценность для бизнеса
- Как ее найти?

# Где взять цифры?

## «Accelerate State of DevOps Report»

- Типы перформеров:
  - Low
  - Medium
  - High
  - Elite
- 4 ключевых метрик
  - Частота развертывания
  - Время цикла разработки
  - Время восстановления
  - Частота отказов

| Aspect of Software Delivery Performance*                                                                                                                                                                                                                                                                                     | Elite                                | High                                   | Medium                                   | Low                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|----------------------------------------|------------------------------------------|--------------------------------------------------|
| <b>Deployment frequency</b><br>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?                                                                                                                                                    | On-demand (multiple deploys per day) | Between once per day and once per week | Between once per week and once per month | Between once per month and once every six months |
| <b>Lead time for changes</b><br>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?                                                                                                   | Less than one day                    | Between one day and one week           | Between one week and one month           | Between one month and six months                 |
| <b>Time to restore service</b><br>For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?                                                                       | Less than one hour                   | Less than one day <sup>a</sup>         | Less than one day <sup>a</sup>           | Between one week and one month                   |
| <b>Change failure rate</b><br>For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0-15% <sup>b,c</sup>                 | 0-15% <sup>b,d</sup>                   | 0-15% <sup>c,d</sup>                     | 46-60%                                           |

Источник: Accelerate State of DevOps Report 2019

| Aspect of Software Delivery Performance*                                                                                                                                                                                                                                                                                     | Elite                                | High                                   | Medium                                   | Low                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|----------------------------------------|------------------------------------------|--------------------------------------------------|
| <b>Deployment frequency</b><br>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?                                                                                                                                                    | On-demand (multiple deploys per day) | Between once per day and once per week | Between once per week and once per month | Between once per month and once every six months |
| <b>Lead time for changes</b><br>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?                                                                                                   | Less than one day                    | Between one day and one week           | Between one week and one month           | Between one month and six months                 |
| <b>Time to restore service</b><br>For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?                                                                       | Less than one hour                   | Less than one day <sup>a</sup>         | Less than one day <sup>a</sup>           | Between one week and one month                   |
| <b>Change failure rate</b><br>For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0-15% <sup>b,c</sup>                 | 0-15% <sup>b,d</sup>                   | 0-15% <sup>c,d</sup>                     | 46-60%                                           |

Источник: Accelerate State of DevOps Report 2019

| Aspect of Software Delivery Performance*                                                                                                                                                                                                                                                                                     | Elite                                | High                                   | Medium                                   | Low                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|----------------------------------------|------------------------------------------|--------------------------------------------------|
| <b>Deployment frequency</b><br>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?                                                                                                                                                    | On-demand (multiple deploys per day) | Between once per day and once per week | Between once per week and once per month | Between once per month and once every six months |
| <b>Lead time for changes</b><br>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?                                                                                                   | Less than one day                    | Between one day and one week           | Between one week and one month           | Between one month and six months                 |
| <b>Time to restore service</b><br>For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?                                                                       | Less than one hour                   | Less than one day <sup>a</sup>         | Less than one day <sup>a</sup>           | Between one week and one month                   |
| <b>Change failure rate</b><br>For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0-15% <sup>b,c</sup>                 | 0-15% <sup>b,d</sup>                   | 0-15% <sup>c,d</sup>                     | 46-60%                                           |

Источник: Accelerate State of DevOps Report 2019

| Aspect of Software Delivery Performance*                                                                                                                                                                                                                                                                                     | Elite                                | High                                   | Medium                                   | Low                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|----------------------------------------|------------------------------------------|--------------------------------------------------|
| <b>Deployment frequency</b><br>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?                                                                                                                                                    | On-demand (multiple deploys per day) | Between once per day and once per week | Between once per week and once per month | Between once per month and once every six months |
| <b>Lead time for changes</b><br>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?                                                                                                   | Less than one day                    | Between one day and one week           | Between one week and one month           | Between one month and six months                 |
| <b>Time to restore service</b><br>For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?                                                                       | Less than one hour                   | Less than one day <sup>a</sup>         | Less than one day <sup>a</sup>           | Between one week and one month                   |
| <b>Change failure rate</b><br>For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0-15% <sup>b,c</sup>                 | 0-15% <sup>b,d</sup>                   | 0-15% <sup>c,d</sup>                     | 46-60%                                           |

Источник: Accelerate State of DevOps Report 2019



| Aspect of Software Delivery Performance*                                                                                                                                                                                                                                                                                     | Elite                                | High                                   | Medium                                   | Low                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|----------------------------------------|------------------------------------------|--------------------------------------------------|
| <b>Deployment frequency</b><br>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?                                                                                                                                                    | On-demand (multiple deploys per day) | Between once per day and once per week | Between once per week and once per month | Between once per month and once every six months |
| <b>Lead time for changes</b><br>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?                                                                                                   | Less than one day                    | Between one day and one week           | Between one week and one month           | Between one month and six months                 |
| <b>Time to restore service</b><br>For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?                                                                       | Less than one hour                   | Less than one day <sup>a</sup>         | Less than one day <sup>a</sup>           | Between one week and one month                   |
| <b>Change failure rate</b><br>For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)? | 0-15% <sup>b,c</sup>                 | 0-15% <sup>b,d</sup>                   | 0-15% <sup>c,d</sup>                     | 46-60%                                           |

Источник: Accelerate State of DevOps Report 2019





# Диспозиция

## DevOps Zoomer

- Разбор результатов тестов
- Процесс вокруг проблем
- Хотели прозрачность
- Поседали в on-call
- **End-to-end ownership**

## DevOps Boomer

- Привычка «конвейерной» разработки
- Мало заинтересованности в быстром фидбеке
- Не пытались сделать процесс прозрачным

# Заявили о своих намерениях

- Мы хотим делать DevOps
- Будем находить проблемы
- Быстрый фидбек и релизы
- Прозрачность
- ***Обещали помогать***



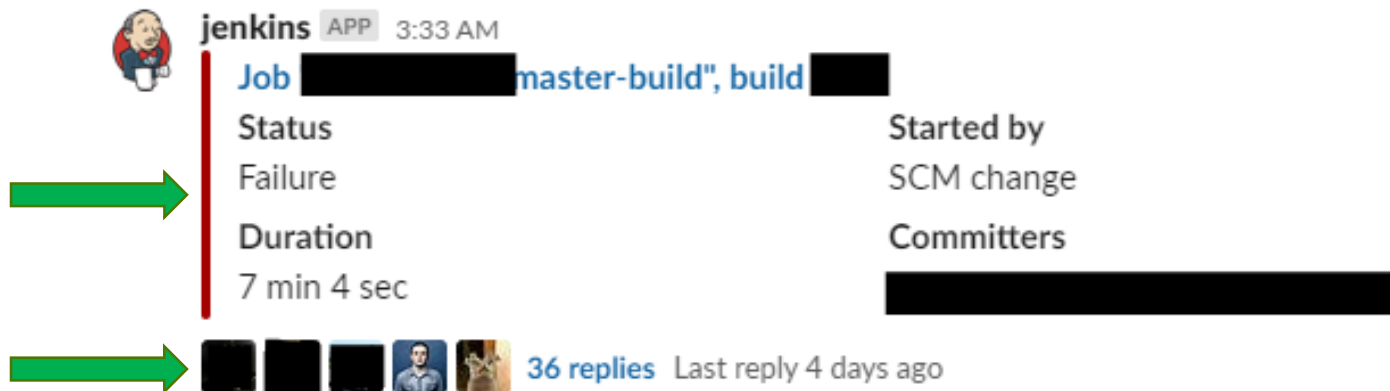
# Определили кто над чем работает

- Знать кто и что разрабатывает
- Знать кто виноват в проблемах
- GitHub owners

# Начали строить пайплайн

- Сделали для своих компонент
- Помогли с билдами для других
- Хотели делать автоматически и быстро
- Не боялись видеть красные билды

# Начали строить пайплайн



The screenshot shows a Jenkins build failure notification. A red vertical bar on the left indicates the build status. Two green arrows point to the 'Failure' status and the '36 replies' link. The notification includes the Jenkins logo, the job name, the status, the duration, the committers, and the number of replies.

 **jenkins** APP 3:33 AM

**Job** [redacted] **master-build", build** [redacted]

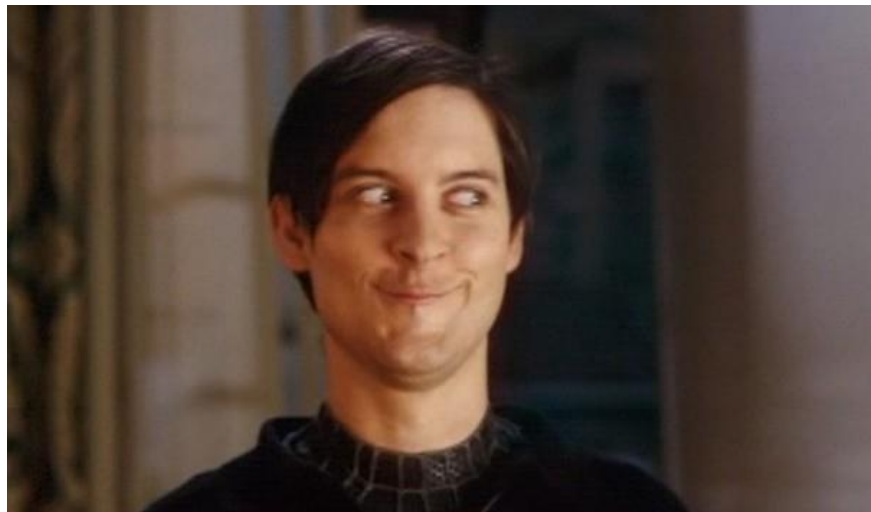
**Status** Failure **Started by** SCM change

**Duration** 7 min 4 sec **Committers** [redacted]

 **36 replies** Last reply 4 days ago

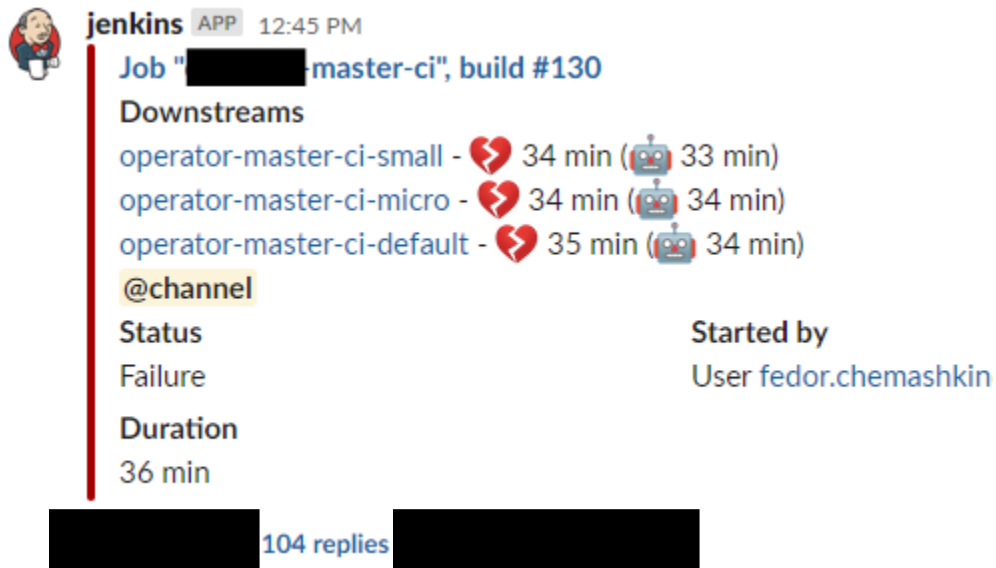
# Почему мы помогли?

- Мы одна команда
- **Знали, что у нас появится интеграция**
- Все строилось, что идет в интеграцию
- Расставит все на свои места



# Появилась интеграция

- Прозрачность
- Дискуссия:
  - описание проблемы
  - упоминание человека
- Доступ к каналу у всех
- Вопросы про падения на daily
- ***Предложения о помощи***



The screenshot shows a chat message from a user named 'jenkins' (with a robot icon) in a Slack channel. The message is timestamped '12:45 PM' and is labeled as an 'APP' message. The content of the message is a Jenkins build status report for 'Job "[REDACTED]master-ci", build #130'. It lists three downstream jobs, all of which failed (indicated by red heart icons): 'operator-master-ci-small' (34 min), 'operator-master-ci-micro' (34 min), and 'operator-master-ci-default' (35 min). The message also includes a '@channel' mention, a 'Status' field (Failure), a 'Duration' field (36 min), and a 'Started by' field (User fedor.chemashkin). At the bottom of the message, there is a link to '104 replies'.

jenkins APP 12:45 PM

Job "[REDACTED]master-ci", build #130

Downstreams

operator-master-ci-small - 34 min (33 min)

operator-master-ci-micro - 34 min (34 min)

operator-master-ci-default - 35 min (34 min)

@channel

Status Started by

Failure User fedor.chemashkin

Duration

36 min

104 replies



**ВЫ ГОВОРИЛИ, ЧТО ОНИ  
БУДУТ РАБОТАТЬ**

**А ОНИ ПОКАЗЫВАЮТ  
НАШИ БАГИ**

**)))**







**Мы изобрели колесо**



# Почему изобрели колесо?

- Отсутствие «настоящих» релизов
- Долго получали фидбек
- Есть куда делать «CD»

# PRODUCTION

- Fake it till you make it:
  - Интеграция – pre-production
  - Longevity – production
- Проблемы из «Production» воспринимаем, как реальные

# Так почему же мы изобрели колесо?

- **Умные люди уже все сказали**

DLD★



***“You build it,  
you run it”***

***(c) Werner Vogels,  
VP&CTO Amazon***

# Кто такие Elite перформеры

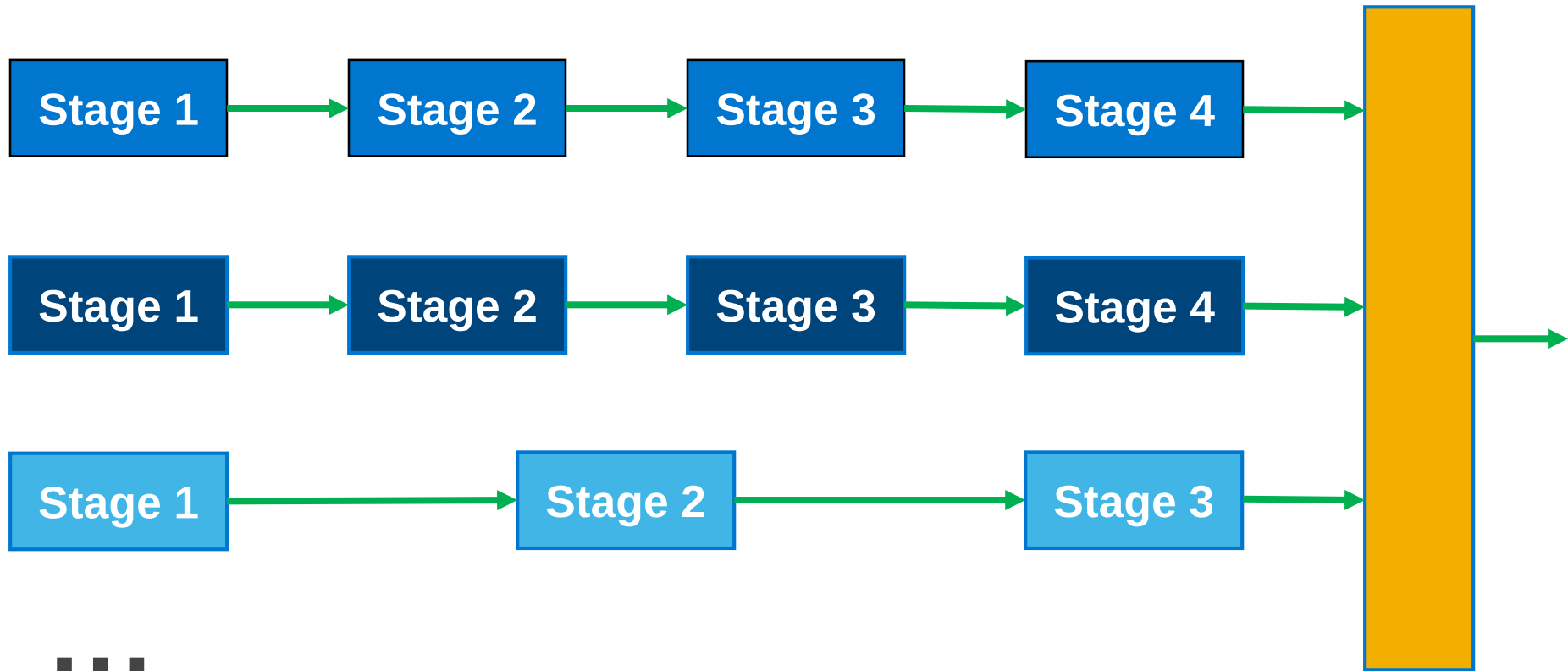
- Ownership – эмоция и обязанность
- В их интересах делать код качественнее:
  - тесты
  - линтеры
  - метрики
  - ...
- On-call/блокеры/писать тесты
- Ошибаться не страшно



# Что получили? Куда дальше?

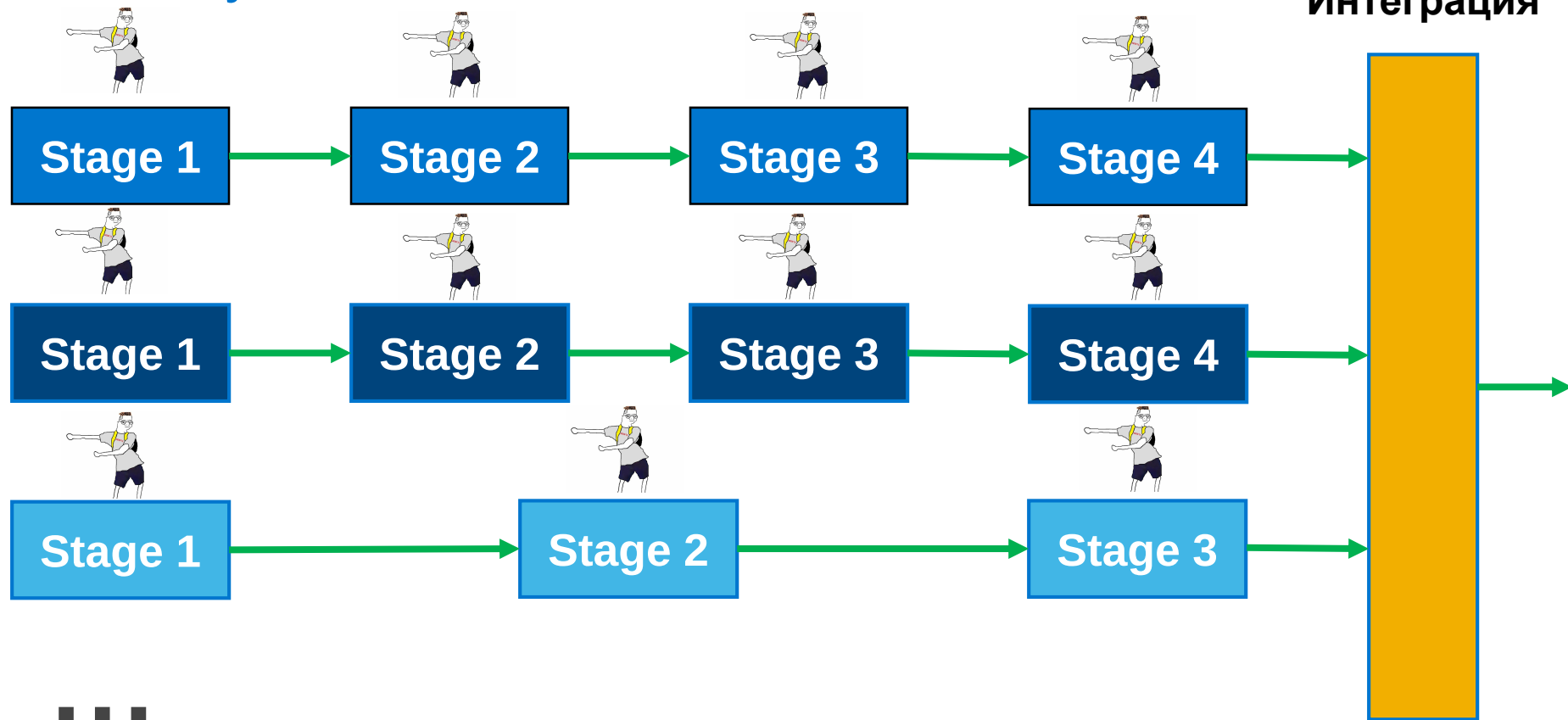
# Что получили?

Интеграция



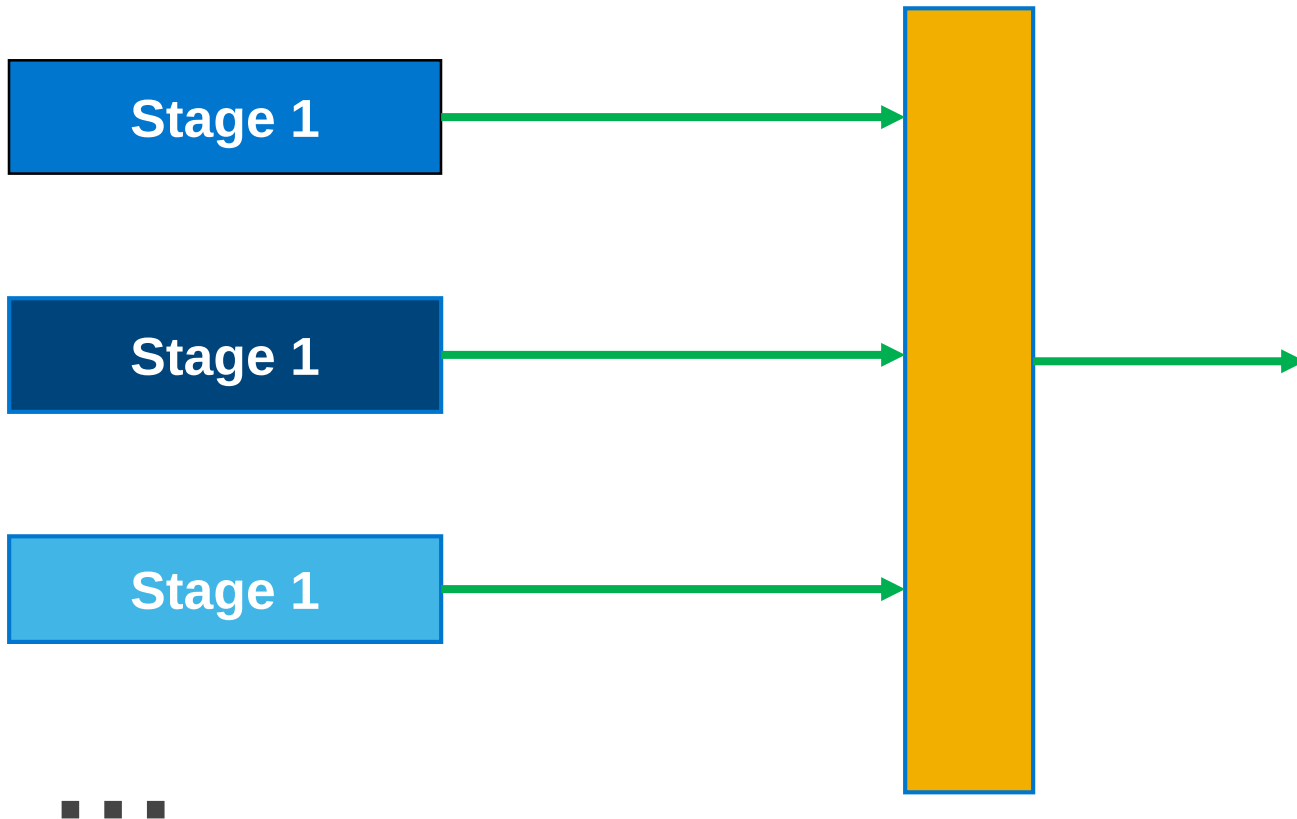
■ ■ ■

# Что получили?

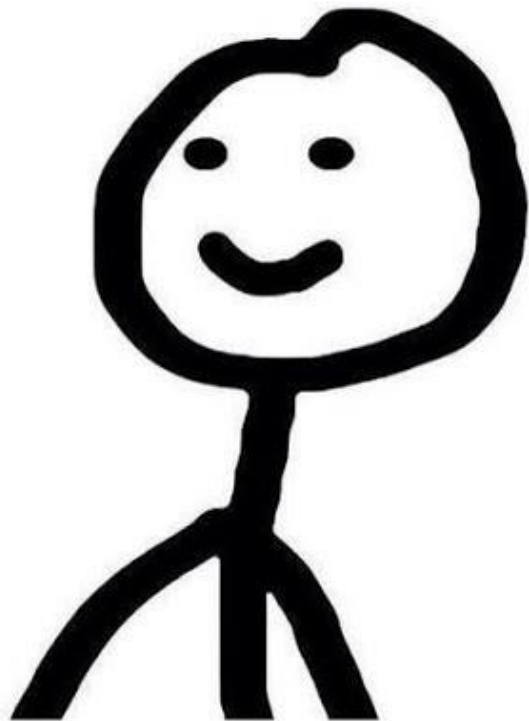


# Получили предложение

## Интеграция



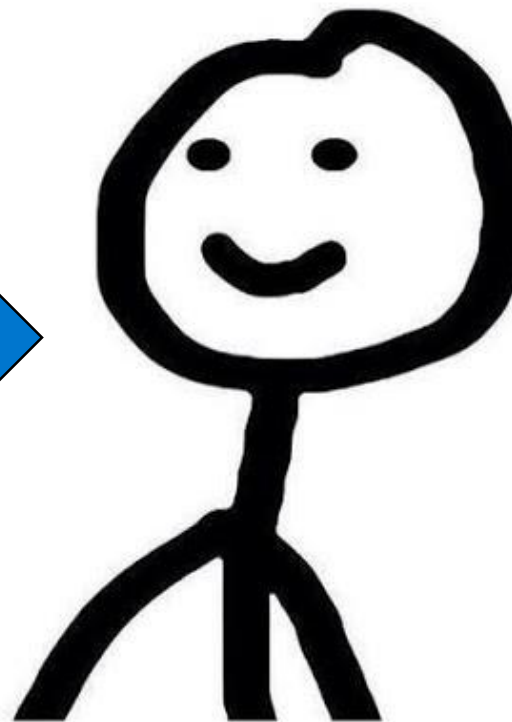
ХМ



интересно



Я



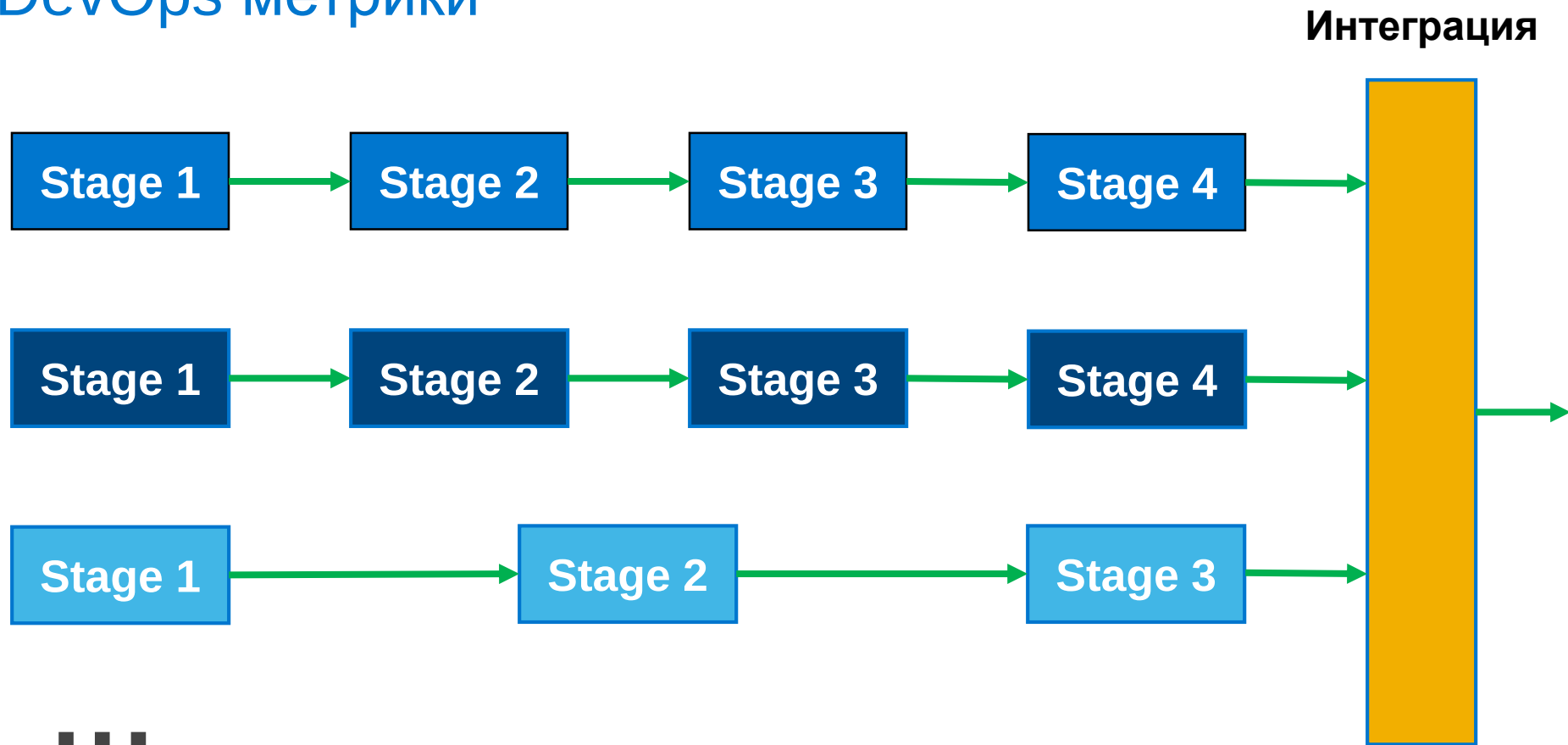
счастлив)))

# Куда дальше?

- «Добить» бумеров
- Сделать оставшиеся факторы
- “To infinity and beyond”



# DevOps метрики





# Выводы



# Выводы

- Cloud'у cloud'ное
- Прозрачность
- Четкий ownership, который нужно привить
- Ошибаться не страшно
- Измерять процессы



КТО ТАМ ПЕРВЫЙ, А?  
НЕЛЬЗЯ ЛИ ДЕЛАТЬ ПАЙПЛАЙН ПРЯМО?

ХЗ, МОЖЕТ КТО-ТО КОСОНОГИЙ  
ИЛИ КОСОГЛАЗЫЙ, ЛОЛ.

РЕАЛЬНО ЗАПАРИЛСЯ УЖЕ  
ТУДА-СЮДА ШАРАХАТЬСЯ  
ПО ПАЙПЛАЙНУ.

ОЙ, НЕ ПОХРЕНУ ЛИ?  
ДЕЛАЕМ – И НА ТОМ СПАСИБО.

Я ШЕРСТЯНОЙ ВОЛЧАРА,  
БОЖЕ, КАК Я ХОРОШ,  
КАК МОЩНЫ МОИ ПАЙПЛАЙНЫ!

КТО ТАМ ПЕРВЫЙ, А?  
НЕЛЬЗЯ ЛИ ДЕЛАТЬ ПАЙПЛАЙН ПРЯМО?

ХЗ, МОЖЕТ КТО-ТО КОСОНОГИЙ  
ИЛИ КОСОГЛАЗЫЙ, ЛОЛ.

РЕАЛЬНО ЗАПАРИЛСЯ УЖЕ  
ТУДА-СЮДА ШАРАХАТЬСЯ  
ПО ПАЙПЛАЙНУ.

ОЙ, НЕ ПОХРЕНУ ЛИ?  
ДЕЛАЕМ – И НА ТОМ СПАСИБО.

Я ШЕРСТЯНОЙ ВОЛЧАРА,  
БОЖЕ, КАК Я ХОРОШ,  
КАК МОЩНЫ МОИ ПАЙПЛАЙНЫ!



КТО ТАМ ПЕРВЫЙ, А?  
НЕЛЬЗЯ ЛИ ДЕЛАТЬ ПАЙПЛАЙН ПРЯМО?

ХЗ, МОЖЕТ КТО-ТО КОСОНОГИЙ  
ИЛИ КОСОГЛАЗЫЙ, ЛОЛ.

РЕАЛЬНО ЗАПАРИЛСЯ УЖЕ  
ТУДА-СЮДА ШАРАХАТЬСЯ  
ПО ПАЙПЛАЙНУ.

ОЙ, НЕ ПОХРЕНУ ЛИ?  
ДЕЛАЕМ – И НА ТОМ СПАСИБО.

Я ШЕРСТЯНОЙ ВОЛЧАРА,  
БОЖЕ, КАК Я ХОРОШ,  
КАК МОЩНЫ МОИ ПАЙПЛАЙНЫ!





The logo for Dell Technologies, featuring the word "DELL" in a bold, sans-serif font, followed by the word "Technologies" in a lighter, sans-serif font. The "E" in "DELL" is stylized with three diagonal lines extending from its right side.

# Полезные ссылки

## Индустрия

- [12factor.net/ru/](https://12factor.net/ru/)
- [bit.ly/job\\_dsl](https://bit.ly/job_dsl)
- <https://aws.amazon.com/blogs/enterprise-strategy/enterprise-devops-why-you-should-run-what-you-build/>

## Научные статьи

- D. S. Linthicum, «Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between»
- A. Balalaie «Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture»
- Migrating to Cloud-Native Architectures Using Microservices: An Experience Report
- Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study

# Q&A