

Инфраструктурные кочевники

Делаем переезды скучными

ПЯТЬ ПЕРЕЕЗДОВ — ОДИН ПРИНЦИП



Будущие



Сергей Фролов

DevOps

Kubernetes Community Podcast

Kubernetes · CI/CD · ML · Documentation

github.com/opsmon



Практика и сообщество

Контрибьютор

Kubernetes

Перевод Официальной документации

Deckhouse

Сообщество пользователей

Yandex Cloud

Документация

Развиваюсь и делюсь

MLOps

ML-инфраструктура и платформы

Статьи и документация

DevOps и инфраструктура

Сообщество

Доклады · митапы · pet-проекты

KUBERNETES

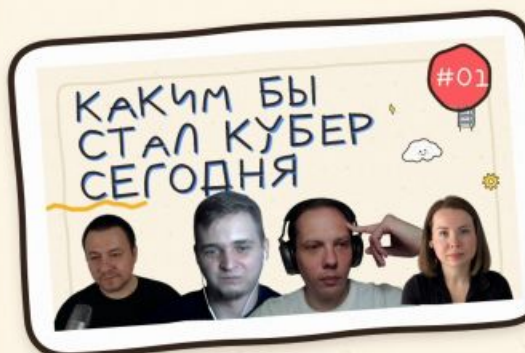
COMMUNITY PODCAST

СМОТРИТЕ НА
YOUTUBE

- ★ НЕЗАВИСИМЫЙ ПОДКАСТ СООБЩЕСТВА
- 📶 КРУТЫЕ ГОСТИ
- ⚡ УВЛЕКАТЕЛЬНЫЕ ТЕМЫ
- 🔌 БЕЗ КОРПОРАТИВНОЙ ЦЕНЗУРЫ



youtube.com/@K8sCommunityPodcast



KUBERNETES

COMMUNITY PODCAST

ВЕДУЩИЕ



СМОТРИТЕ НА
YOUTUBE



youtube.com/@K8sCommunityPodcast

ТЧМУР



СЕРГЕЙ



ЮЛЧЯ



АРТЕМ





Это не готовый рецепт Это мой опыт

—

Ошибки

—

Контекст

—

Практики

Именно такой экран хочется увидеть в финале

prod-migration | kubernetes | us-east-1 | context: production | Tue, Apr 23 14:27:11

```
ops@infra:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
ip-10-0-1-12        Ready    control-plane   42d   v1.28.4
ip-10-0-2-18        Ready    worker         42d   v1.28.4
ip-10-0-3-27        Ready    worker         41d   v1.28.4

ops@infra:~$ helm upgrade --install platform ./charts/platform \
> --namespace prod --values values-prod.yaml --wait --timeout 10m
Release "platform" has been upgraded. Happy Helming!
NAME: platform
LAST DEPLOYED: Tue Apr 23 14:18:04 2024
NAMESPACE: prod
STATUS: deployed
REVISION: 12
TEST SUITE: None

ops@infra:~$ kubectl rollout status deployment/platform-api -n prod
deployment/platform-api successfully rolled out

ops@infra:~$ kubectl get pods -n prod
NAME                                READY   STATUS    RESTARTS   AGE
platform-api-7d9f8c6d9b-2k4sm       1/1     Running   0           2m14s
platform-api-7d9f8c6d9b-f9q2p       1/1     Running   0           2m16s
platform-worker-6b7c9f8d6-ptz7h     1/1     Running   0           2m16s
platform-scheduler-5d6d4b7f9-q18nz  1/1     Running   0           2m12s

ops@infra:~$ kubectl get cronjobs -n prod
NAME                SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
report-generation   0 2 * * *     False    0        2h ago          12d
user-sync            */15 * * * *  False    0        14m ago         12d
cleanup-temp        0 4 * * *     False    0        4h ago          12d

ops@infra:~$ kubectl get svc -n prod | grep platform
platform-api        ClusterIP   10.96.12.34   <none>    8080/TCP    12d
platform-worker     ClusterIP   10.96.56.78   <none>    9090/TCP    12d

ops@infra:~$ echo "Post-migration checks completed successfully."
Post-migration checks completed successfully.
ops@infra:~$
```

Migration checklist6/6

✓

Apply infrastructure (Terraform)2m ago

✓

Deploy application (Helm)2m ago

✓

Verify deployments1m ago

✓

Check cron jobs1m ago

✓

Run smoke tests30s ago

✓

Confirm traffic & monitoringjust now

Cluster resourcesLast 15 minutes

CPU28%

Memory62%

Pods48 / 60

Recent eventsAll systems nominal

14:26:52 ✓ platform-apiScaled up replica set to 2

14:24:18 ✓ platform-workerSuccessfully rolled out

14:21:03 ✓ ConfigMapUpdated

14:18:04 ✓ HelmRelease upgraded (rev. 12)

14:17:39 ✓ cronjob/user-syncSchedule verified

14:16:11 ✓ ServiceEndpoints healthy



Пять переездов позади

5

завершённых
миграций



Пять переездов позади

5

завершённых
миграций

6-я

TO BE CONTINUED



Маршрут доклада

01

Контекст
и переезды

02

Инженерная
часть

03

Командная
зрелость

**Один ручной сервис
может задержать
весь переезд**

ВЗРОСЛЕНИЕ ПРОДУКТА



Переезд — нормальный этап взросления

01



02



03

Сложность

Технический долг и ограничения архитектуры.

Рост

Больше нагрузки, команды и бизнес-логики.

Миграция

Переход к решениям, подходящим продукту.



Один продукт — разные стадии жизни

01



02



03

**Внутренний
стартап**

Небольшой проект
внутри крупной
компании

Рост продукта

Команда и нагрузка
увеличились

**Платёжный
сервис**

Больше внешних
и внутренних
интеграций



Масштаб — больше, чем число инженеров

ПРОДУКТ

**Десятки
специалистов**

Продуктовая команда

Backend · Frontend · Mobile · QA
Аналитика · поддержка · архитектура

ИНФРАСТРУКТУРА

**Небольшая
команда**

DevOps + архитектор



Стек: приложение и данные

Python

Разработка приложения

PostgreSQL

Основная база данных

Redis

Кэш и очереди

RabbitMQ · Kafka

Сообщения и потоковые данные



Стек: инфраструктура и доставка

Docker · Kubernetes

Контейнеризация и оркестрация

Helm · Argo CD

Шаблоны и доставка через GitOps

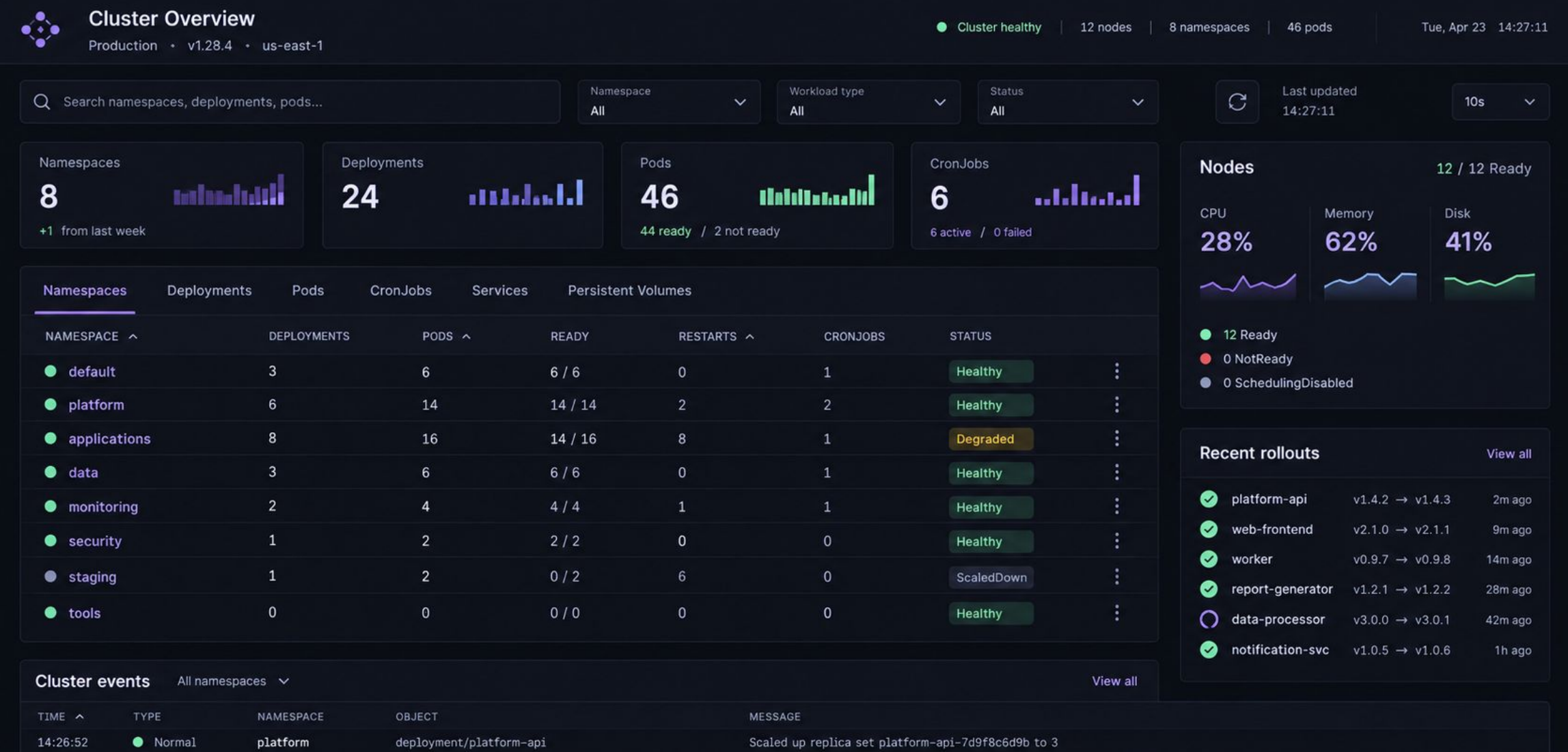
Terraform

Описание инфраструктуры

Vault

Управление секретами

Стек превращается в работающий контур





Состав и состояние контура

Состав

Сервисы

Реплики

Расписания

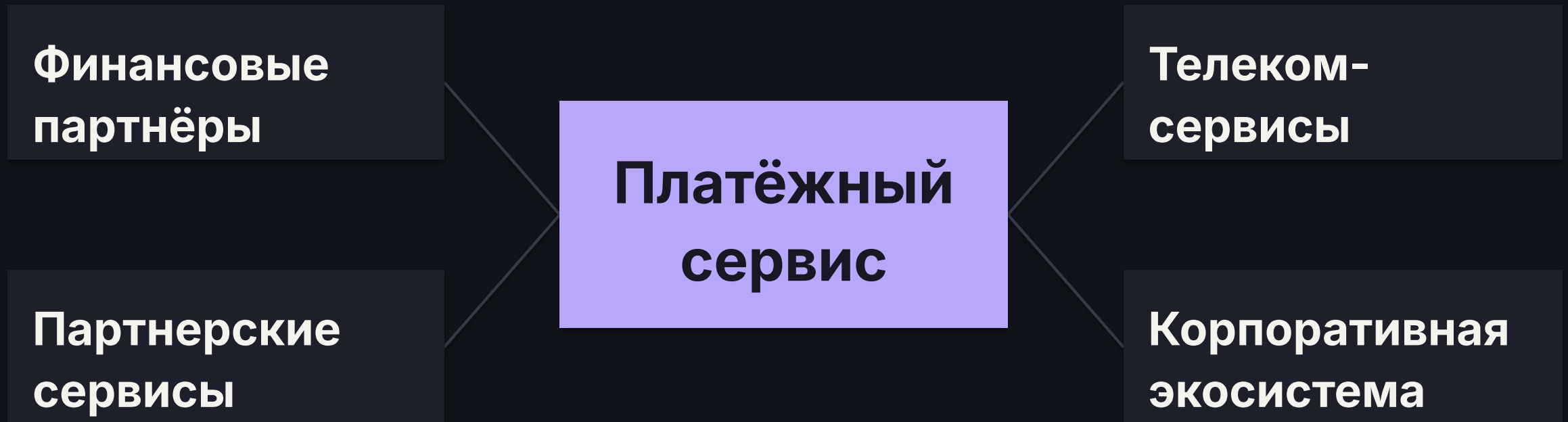
Состояние

Что запущено

Что отличается
от ожидаемого



Интеграции тоже переезжают



Внутренние системы и внешние API остаются частью картины.

**Вчерашняя удачная платформа
может стать сегодняшним
ограничением**

ПРИЧИНЫ ПЕРЕЕЗДОВ



Приоритеты меняются вместе с продуктом

На старте

- Скорость запуска
- Гибкость архитектуры
- Минимум согласований

Это смена приоритетов, а не спор «правильно или неправильно».



Приоритеты меняются вместе с продуктом

На старте

- Скорость запуска
- Гибкость архитектуры
- Минимум согласований

На зрелом этапе

- Отказоустойчивость и откат
- Безопасность и стандарты
- Наблюдаемость

Это смена приоритетов, а не спор «правильно или неправильно».



Что дороже: терпеть или переехать?

Оставить как есть

- Ограничения роста
- Ручные операции
- Риски отказов
- Стоимость поддержки



Что дороже: терпеть или переехать?

Оставить как есть

- Ограничения роста
- Ручные операции
- Риски отказов
- Стоимость поддержки

Вложиться в миграцию

- Повторяемая автоматизация
- Управляемый откат
- Требования безопасности
- Предсказуемый рост

**Меняются требования —
меняются приоритеты**

ПЕРЕЕЗДЫ

ПЕРЕЕЗД 1



Мы просто выросли

Продукт вырос

Больше ресурсов
Больше изменений
Выше требования

Потребовалось

Больше гибкости
в управлении
инфраструктурой

ПЕРЕЕЗД 1 ЗАВЕРШЁН

ПЕРЕЕЗД 2



Технически хорошо, но есть ИБ :)

Получили

Ресурсы

Гибкость

Масштабирование

Ограничение

Требования ИБ

Из внешнего облака
пришлось уйти

ПЕРЕЕЗД 2 ЗАВЕРШЁН

ПЕРЕЕЗД 3



Платформа отстала от продукта

VMware → OpenStack

Мощности · гибкость · инфраструктурная база

ПЕРЕЕЗД 3 ЗАВЕРШЁН

ПЕРЕЕЗД 4



Организация открыла возможности

Изменилась

Организационная
структура

Стала доступна

Корпоративная
инфраструктурная
платформа

ПЕРЕЕЗД 4 ЗАВЕРШЁН

ПЕРЕЕЗД 5



Собрать всё в одном месте

Было

Распределённые
системы

Стало

Консолидация
в целевой среде

ПЕРЕЕЗД 5 ЗАВЕРШЁН

ПЕРЕЕЗД 6



Следующий переезд впереди

Новый ЦОД

Дата не определена

6-я

TO BE CONTINUED

ПЛАН · ПОДДЕРЖИВАЕМ ПЕРЕНОСИМОСТЬ



У каждого переезда была своя причина

01 ЗАВЕРШЁН

Рост

Ресурсы и гибкость

02 ЗАВЕРШЁН

Требования ИБ

Смена внешней площадки

03 ЗАВЕРШЁН

Новая база

VMware → OpenStack

04 ЗАВЕРШЁН

Структура компании

Доступ к платформе

05 ЗАВЕРШЁН

Консолидация

Системы в одном контуре

06 ПЛАН

Следующий ЦОД

Сохраняем переносимость



Скорость стала побочным эффектом зрелости

- 01 Автоматизация и стандарт развёртывания
- 02 Описание инфраструктуры и чек-листы
- 03 План отката и раннее участие разработчиков

**Мы хотели не быстрее
Мы хотели предсказуемее**

ЧТО ВСКРЫЛ ПЕРЕЕЗД



Всплывает всё, что откладывали

Связи

Неизвестные зависимости

Неизвестные владельцы

Переезд — это ещё и аудит зрелости продукта.



Всплывает всё, что откладывали

Связи

Неизвестные зависимости
Неизвестные владельцы

Автоматика

Забытые CronJob
Старые скрипты

Переезд — это ещё и аудит зрелости продукта.



Всплывает всё, что откладывали

Связи

- Неизвестные зависимости
- Неизвестные владельцы

Автоматика

- Забытые CronJob
- Старые скрипты

Конфигурация

- Ручные настройки
- Мёртвые переменные

Переезд — это ещё и аудит зрелости продукта.



Всплывает всё, что откладывали

Связи

- Неизвестные зависимости
- Неизвестные владельцы

Конфигурация

- Ручные настройки
- Мёртвые переменные

Автоматика

- Забытые CronJob
- Старые скрипты

Долг

- Завышенные requests / limits
- Старые обходные решения

Переезд — это ещё и аудит зрелости продукта.

Инвентаризация — это не только pods

inventory / example-prod

read-only / пример

```
$ kubectl get deploy,sts,cronjob -n example-prod
```

NAME	READY
deployment.apps/backend-api	3/3
deployment.apps/worker	2/2
statefulset.apps/cache	1/1

NAME	SCHEDULE	SUSPEND
cronjob.batch/cleanup	0 2 * * *	False

— А у каждого найденного объекта есть владелец?



«Просто добавим CPU» — не план

CPU ↑

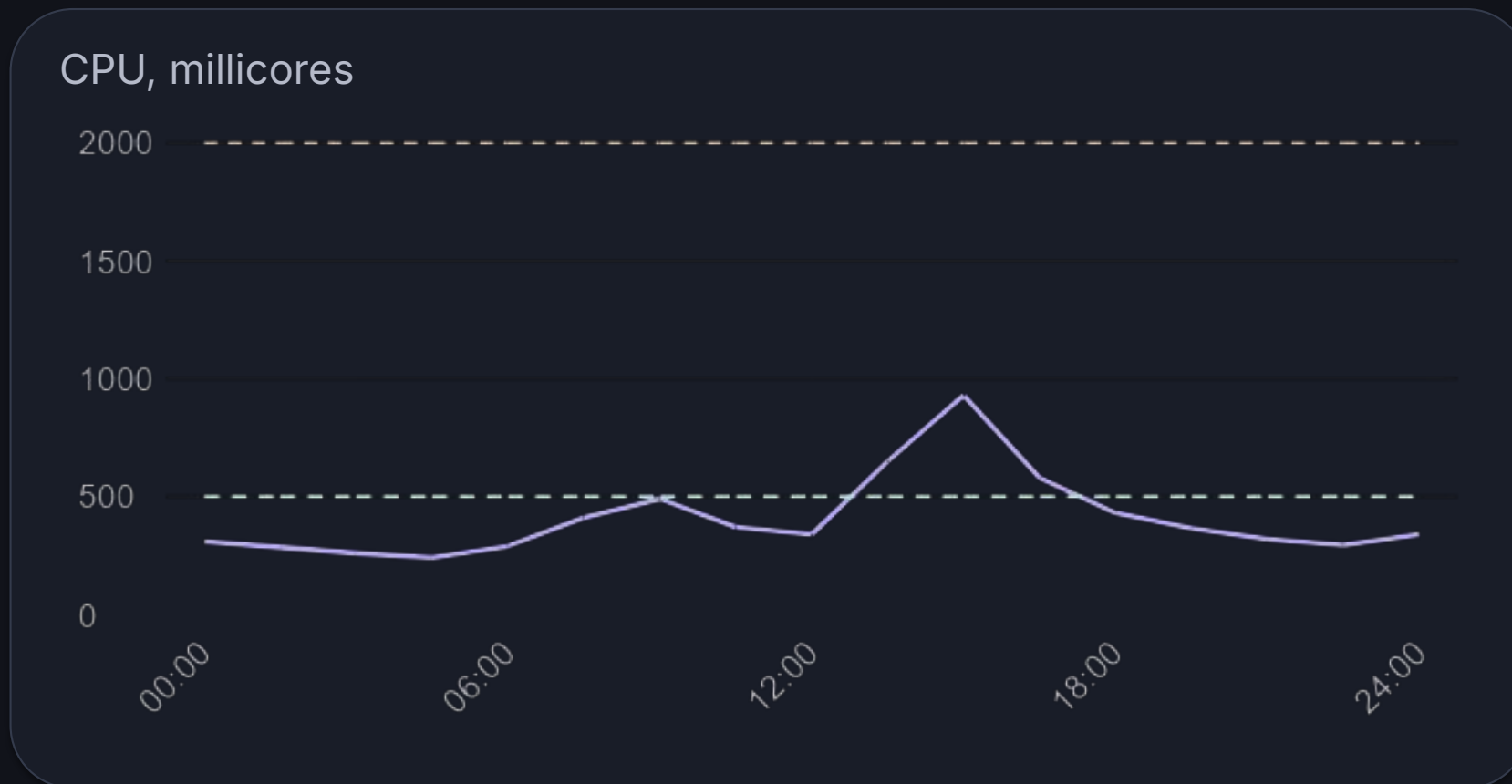
RAM ↑

«Потом разберёмся»

Requests	Заявка на ресурсы
Limits	Ограничения
Usage	Фактическая нагрузка
Replicas	Число экземпляров
Reserve	Запас на рост

— Сначала измерить. Потом менять.

Нужен профиль нагрузки, а не один замер



REQUEST

500m

LIMIT

2000m

ПРИМЕР ПОТРЕБЛЕНИЯ

~ 340m

Смотрим на историю нагрузки, а не на один момент.

Паспорт сервиса: кто, где, с чем

service-passport.yaml

пример / не конфигурация

```
service: backend-api
kind: Deployment
namespace: example-prod
owner: "Команда продукта"
criticality: high
dependencies:
  - Kafka
  - PostgreSQL
  - "Кэш Redis"
  - "Сервис авторизации"
```

01

Кто отвечает

Владелец
и критичность.

02

Зависимости

Связи рядом
с описанием сервиса.

Паспорт сервиса: выделено ≠ потребляется

CPU, millicores



Факт приблизительный: ~340m

RAM

Request **1Gi**

Limit **4Gi**

Факт **~ 1.2 GB**

Реплик: 3

НРА: от 2 до 10

Ресурсный профиль фиксируем до изменений.



Паспорт обновляется вместе с деплоем

Было: две отдельные работы

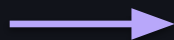
Деплой

--

Ручное обновление паспорта

Стало: один процесс

Деплой



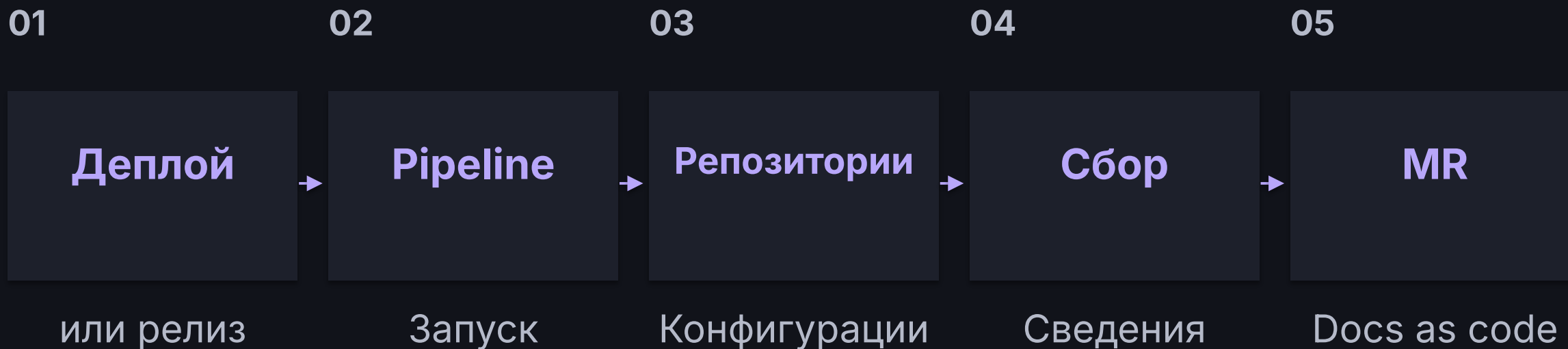
Pipeline



MR с паспортом



После деплоя — MR с паспортом



Результат pipeline — MR в docs-as-code репозиторий.

Изменение паспорта видно в MR

docs / passports / demo.yaml

```
# учебный фрагмент паспорта
service: backend-api
environment: demo
- release: demo-v1
+ release: demo-v2
# обновлено после деплоя
```

Не напоминание — часть деплоя.

ИЗМЕНЕНИЕ

В Git

История
документации.

ПРОЦЕСС

Через MR

Изменение видно
всей команде.



Паспорт стенда: состав и зависимости

Контур / кластер / namespace / владелец

Состав

Deployments

StatefulSets

Реплики

CronJob

Зависимости

PostgreSQL

Redis / Memcached

Kafka / RabbitMQ

Внешние API



Паспорт стенда: готовность к эксплуатации

Ресурсы

CPU / RAM

Requests / limits / потребление

Доступ

Секреты

Сертификаты

Контроль

Мониторинг

Оповещения

Переезд

Проверки контура

Стратегия отката

Если стенд не описан, он удивит в неудобный момент.



Внутренний стартап — не безлимитный бюджет

Контекст

Продукт ещё доказывает ценность внутри крупной компании.

Ограничение

Каждый CPU, гигабайт памяти и род приходится обосновывать.

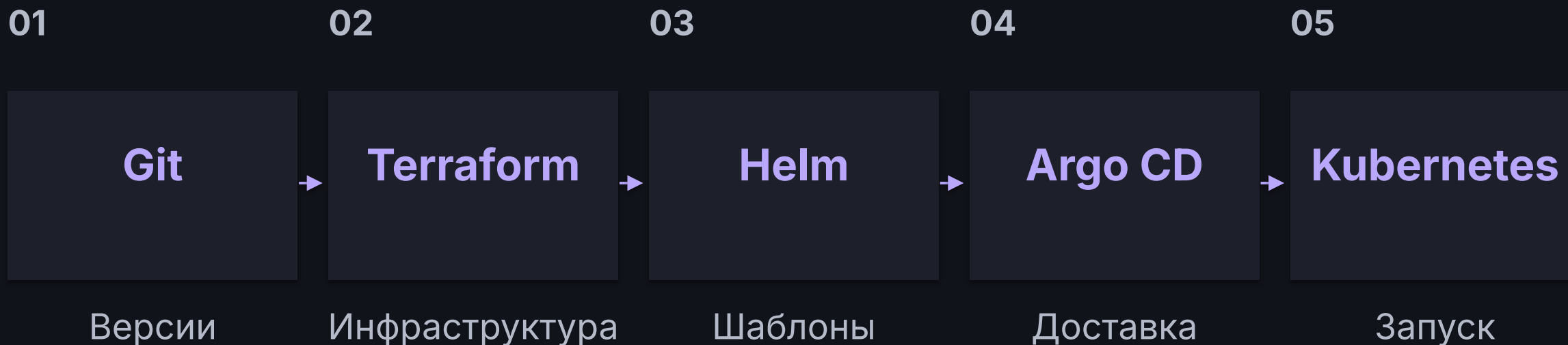
Подход

История потребления, requests / limits, профиль до изменений.

**Не знаешь потребление?
Не управляешь — надеешься**

laC · HELM · GITOPS

Повторяемая доставка начинается в Git



Один описанный путь вместо набора ручных операций.

У переезда есть маршрут и точка решения

Pipelines

Deployments

Environments

Artifacts

Settings

pipelines > migration-platform > #1024

Search runs, commits, or files...

/

AK

#1024 Migrate platform to new infrastructure

Running (7/9 completed)

triggered 14 min ago by alex.k

a3f9c2e

main

Re-run

lint42s

test2m 18s

build1m 03s

package48s

deploy-stage3m 21s

verify1m 12s

deploy-prod

manual approval

smoke-test-

post-checks-

Logs

Summary

Tests124

Artifacts3

Environments

Annotations

All stages

lint42s

test2m 18s

build1m 03s

package48s

deploy-stage3m 21s

verify1m 12s

deploy-prod

Waiting for approval

smoke-test-

post-checks-

114:12:01

▶ Starting production deployment (manual approval required)

214:12:01

① Target environment: production

314:12:01

① Release version: v1.28.0

414:12:01

① Infra changes: 12 to apply

514:12:01

① Application image: registry/app:1.28.0

614:12:02

① Helm chart: platform-0.14.0

714:12:02

814:12:02

⓪ Waiting for manual approval from an authorized user...

914:12:02

⓪ Required approvers: platform-team, sre-team

1014:12:02

⓪ Timeout: 24h

1114:12:02

1214:12:02

① To approve, review the changes and click "Approve" in the UI.

1314:12:02

① You can also add a comment (optional).

Run details

Status

Waiting for approval

Trigger

Push to main

Commit

a3f9c2e

Author

alex.k

Branch

main

Started

Apr 23, 2024 14:12:01

Duration

14m 07s

Estimated remaining

~ 6m

Workflow

.ci/pipeline.yml

Commit message

Migrate platform to new infrastructure

- update Terraform modules
- bump platform chart to 0.14.0
- enable new node pools

Artifacts3

platform-1.28.0.tgz

42 MB

charts-0.14.0.tgz

18 MB

sbom-platform.json

2.1 MB

Environment

production

pending

Region

us-east-1

Cluster

prod-eks-1

Namespace

platform

URL

https://platform.example.internal

- 01
- Один путь
- Проверки и доставка в одном процессе.
- 02
- Решение
- Условие следующего шага известно заранее.

Развернули — ещё не значит переехали

post-deploy / example-prod

```
$ kubectl rollout status deploy/backend-api \
  -n example-prod
deployment "backend-api" successfully rolled out
```

```
$ kubectl get pods -n example-prod
```

NAME	READY	STATUS
backend-api-example	1/1	Running



Технические тесты

Запуск

Приложение работает
в новом контуре

Состояние

Проверено техническое
состояние

Связи

Доступны необходимые
зависимости



Бизнес-сценарии и интеграции

Сценарии

Не только запуск
Но и полезная работа

Интеграции

Не только наш сервис
Но и связи с другими



То, что повторяется, должно быть описано

Ручная работа

- Настройки в интерфейсе
- Ручная настройка серверов
- «Я помню, как это делается»

Terraform · Helm · GitOps



То, что повторяется, должно быть описано

Ручная работа

- Настройки в интерфейсе
- Ручная настройка серверов
- «Я помню, как это делается»

Инфраструктура как код

- Описание в Git
- Проверка изменений коллегами
- Воспроизводимое развертывание

Terraform · Helm · GitOps

Конфигурация и действия — в репозитории

Infra.git

```
$ tree --dirsfirst -F infra
```

```
infra/
```

```
|— charts/  
|— checks/  
|— environments/  
|— runbooks/  
|   |— migration.md  
|   |— rollback.md  
|— terraform/
```

```
5 directories, 2 files
```



«Этот сервис маленький, его можно руками»

01



02



03

Сегодня

Через год

При переезде

Маленький сервис без особого внимания.

Критичный сервис под большой нагрузкой.

Никто не помнит, как его развернуть.

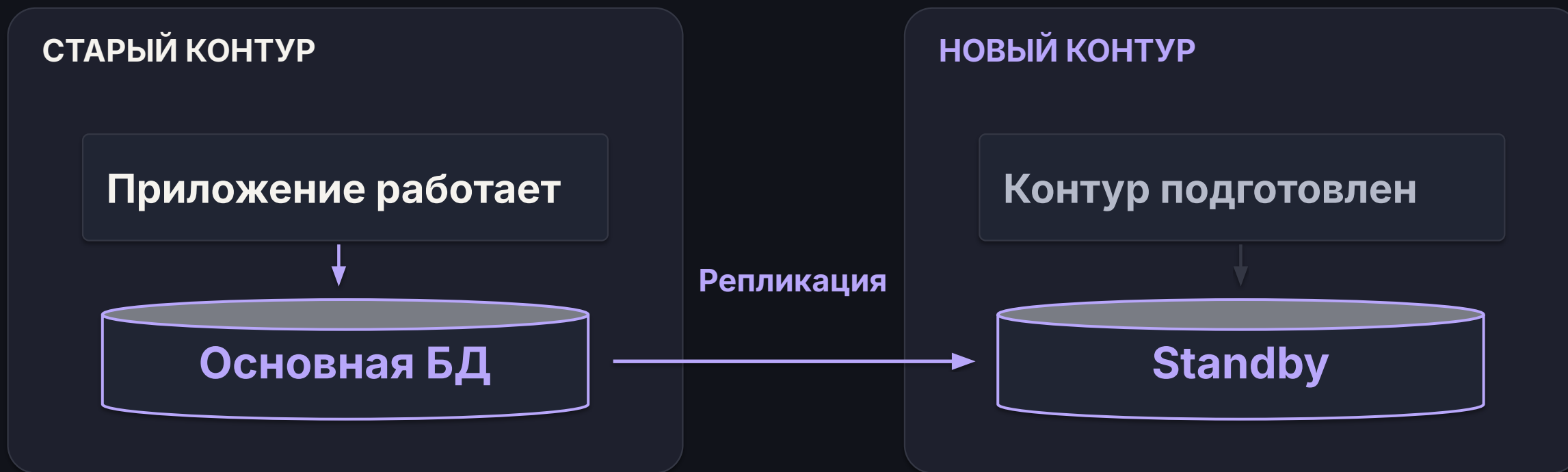
Один неописанный сервис задерживает автоматизированную платформу.

**То, что раньше помнили,
теперь описано**

ДАННЫЕ · ТРАФИК · ОТКАТ

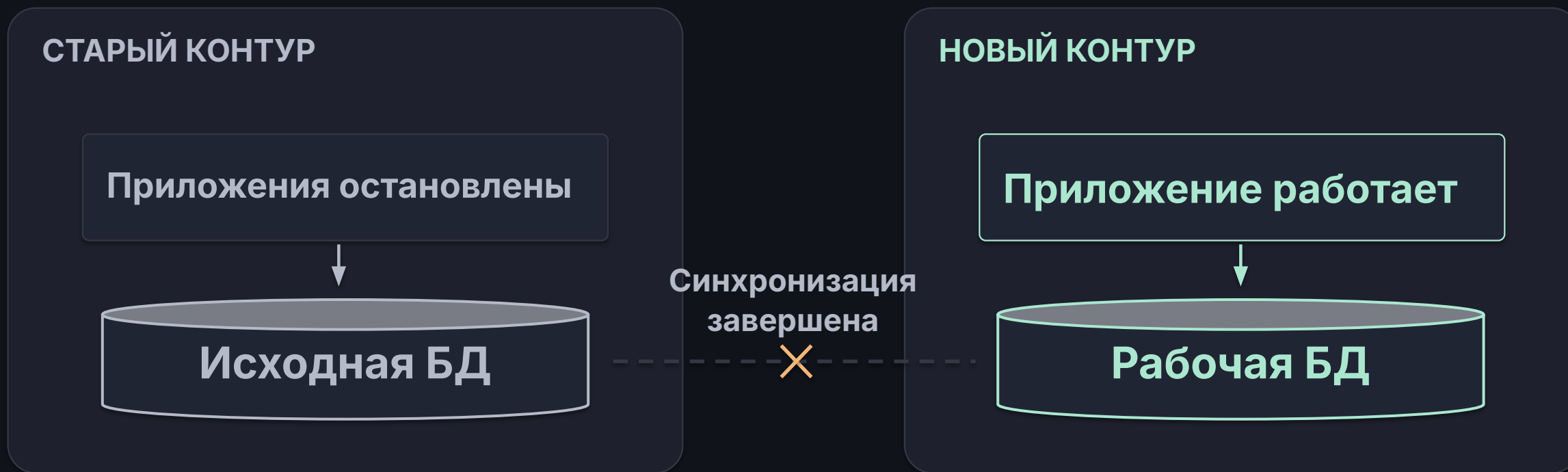
Standby готовим до окна переключения

ДО ОКНА



Переключаемся на подготовленный контур

ПОСЛЕ ПЕРЕКЛЮЧЕНИЯ





Один переезд — один маршрут

ДО ОКНА

01 **Контур
и standby**

02 **Репликация**

03 **Тесты
и интеграции**

04 **Мониторинг**

Критерии и действия обсуждали очень много с DBA до переключения.



Один переезд — один маршрут

ДО ОКНА

01 Контур
и standby

02 Репликация

03 Тесты
и интеграции

04 Мониторинг

В ОКНО

05 Остановить
приложения

06 Проверить
запись

07 Завершить
репликацию

08 Переключить
и наблюдать

Критерии и действия обсуждали очень много с DBA до переключения.

Проверяем пользовательский сценарий

ПРОВЕРКИ НОВОГО КОНТУРА

Техническое состояние

ПРОЙДЕНО

Бизнес-сценарий

ПРОЙДЕНО

Внешние интеграции

ПРОЙДЕНО

Решение о переключении

ОЖИДАЕТ

01

Поведение

Проверили не только запуск, но и сценарий.

02

Дальше

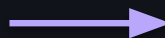
Отдельное решение о переключении.



От записей в БД — к работе бизнеса

Мониторинг БД

Количество записей
Динамика во время работ
Оценка завершения записи



Бизнес-метрики



От записей в БД — к работе бизнеса

Мониторинг БД

Количество записей
Динамика во время работ
Оценка завершения записи



Бизнес-метрики

Работа процессов
Поведение приложения
Результат переключения



Откат проверяется до переключения

runbooks / rollback.md

- ✓ **Предусмотрели сценарий возврата**
- ✓ **Провели репетицию**
- ✓ **Обсудили действия с DBA**
- ✓ **Подключили опытных инженеров**



Мониторинг появляется до миграции

Метрики и логи

Состояние систем
Журналы приложений

Бизнес-показатели

Транзакции
Клиентский путь

Оповещения

Правила тревог
Рабочие панели

Коммуникация

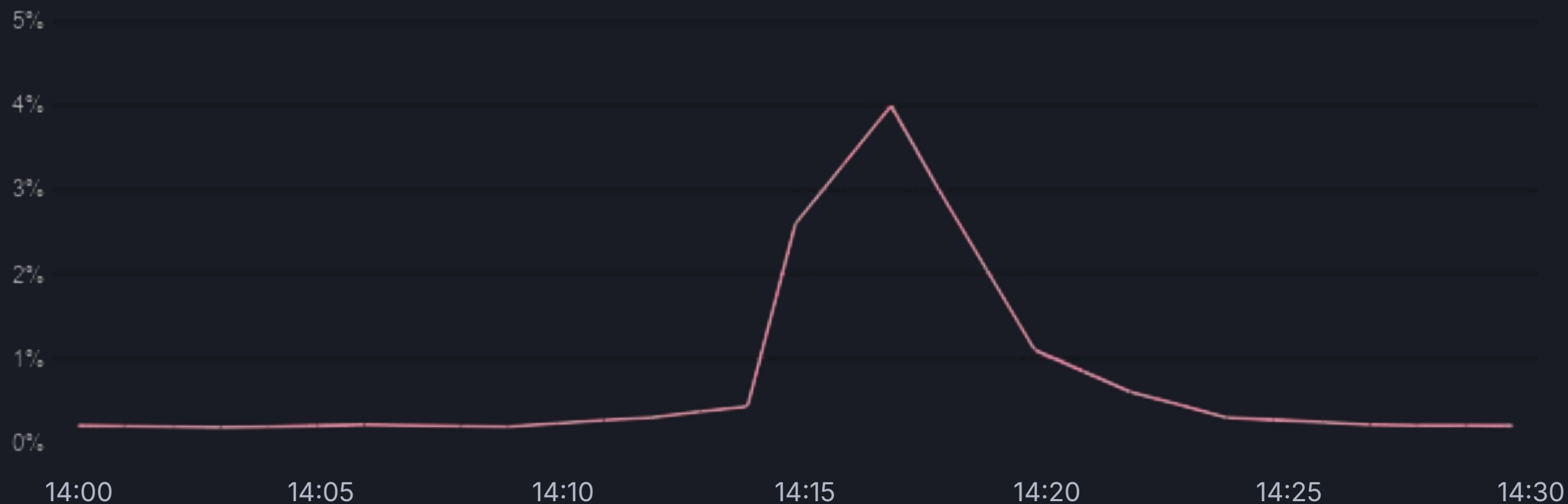
Каналы связи
Дежурные и информирование

Всё проверено до переключения первого процента трафика.

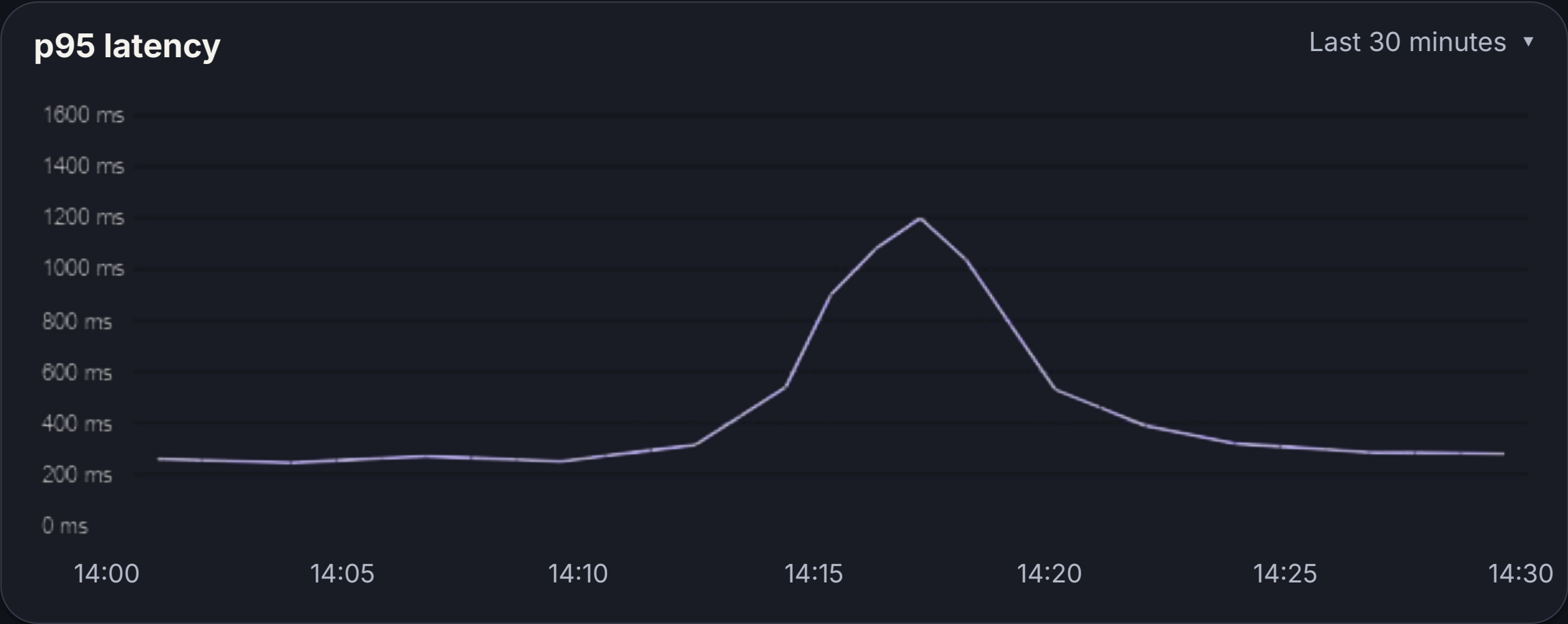
Смотрим ошибки

5xx responses

Last 30 minutes ▾



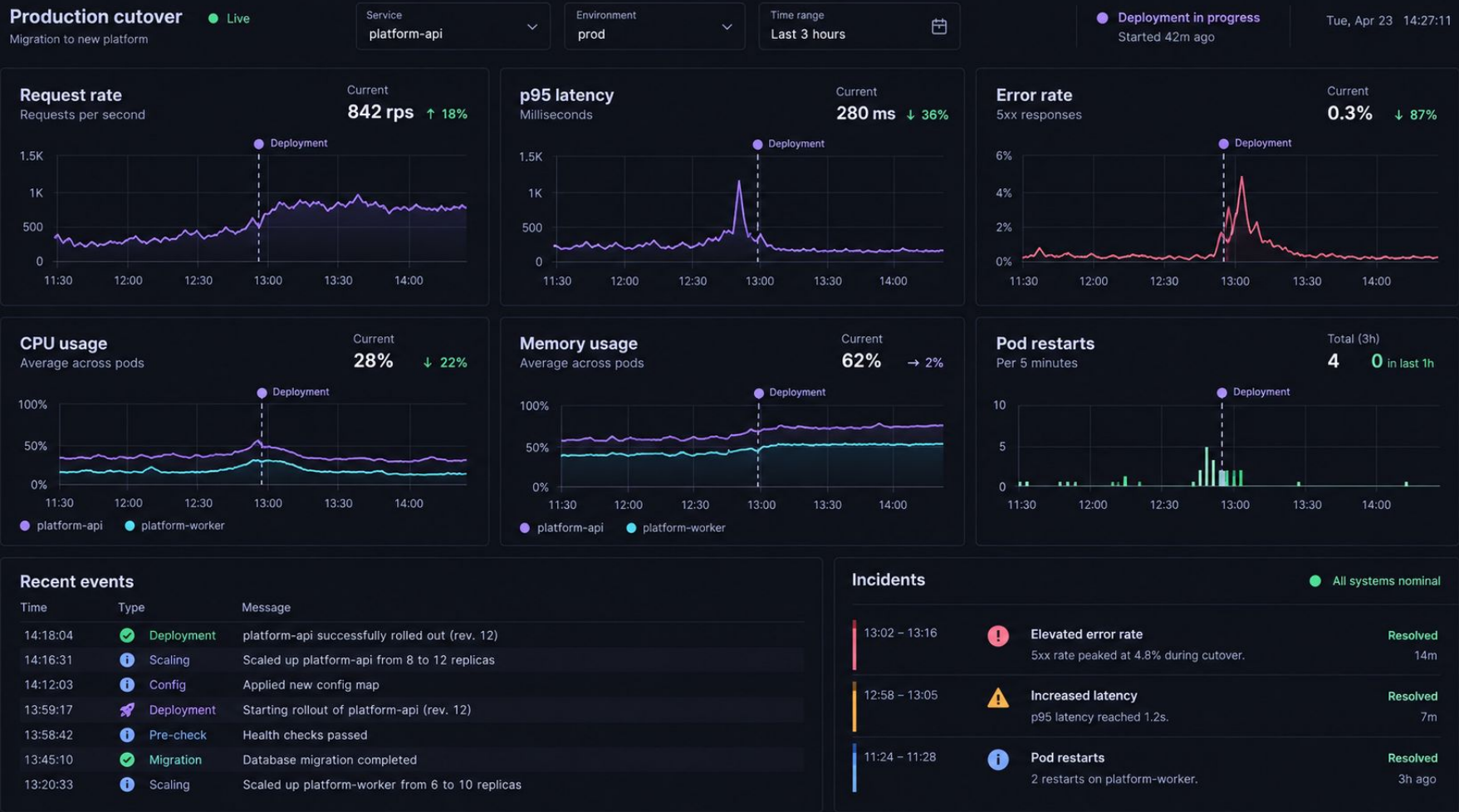
Смотрим задержки



Смотрим разрывы



Общая картина переключения



От изменения на графике — к журналу событий

prod-migration

logs

infrastructure

services

alerts

Last 3 hours

Live

Log explorer

Search and investigate logs across your infrastructure

service:platform-api env:prod (migrate OR sync OR cron)

Last 3 hours

Run

service: platform-api

env: prod

level: error, warn

+ Add filter

1,248 log lines

Auto

400

200

0

11:00

11:15

11:30

11:45

12:00

12:15

12:30

12:45

13:00

13:15

13:30

13:45

TIME

LEVEL

SERVICE

MESSAGE

2024-04-23 12:14:01.231

INFO

platform-api

Starting data migration batch job {batch_id=8472}

2024-04-23 12:14:03.118

INFO

platform-api

Connecting to source cluster eu-1...

2024-04-23 12:14:05.337

WARN

platform-api

Missing labels for 342 resources, defaulting to "unknown"

2024-04-23 12:14:05.338

WARN

platform-api

Label validation failed: key "team" not found on 218 resources

2024-04-23 12:14:07.921

INFO

platform-api

Fetches 1,204 objects from source

2024-04-23 12:14:12.447

ERROR

platform-api

Object sync failed: 23 objects could not be matched (id mismatch)

2024-04-23 12:14:12.448

ERROR

platform-api

Sync aborted: too many unmatched objects

2024-04-23 12:14:12.449

INFO

platform-api

Rolling back partial changes...

2024-04-23 12:14:18.003

ERROR

scheduler

Cron trigger failed: job "migrate-daily" (exit code 1)

2024-04-23 12:14:18.004

ERROR

scheduler

Failed to enqueue migration job: previous run still marked as failed

2024-04-23 12:14:22.119

INFO

platform-api

Clenaup completed, removed 17 temporary resources

2024-04-23 12:14:46.231

INFO

platform-api

Retrying sync with id mapping...

2024-04-23 12:15:02.773

WARN

platform-api

Skipping 5 objects with invalid metadata

2024-04-23 12:15:18.662

INFO

platform-api

Sync completed: 1,181/1,204 objects synced (skipped: 5, failed: 0)

2024-04-23 12:15:19.004

INFO

scheduler

Cron job "migrate-daily" rescheduled for 2024-04-24T00:00:00Z

2024-04-23 12:15:19.105

INFO

platform-api

Migration run finished successfully

INC-4827

Resolved

Migration sync failed

Data sync encountered object mismatches and missing labels during production migration.

Severity

Start time

End time

Duration

Affected service

Environment

Related services

High

Apr 23, 2024 12:14:05

Apr 23, 2024 12:15:19

1m 14s

platform-api

prod

scheduler, object-sync

Timeline

12:14:05

Issue detected

Increased error rate in platform-api

12:14:12

Object sync failures

23 unmatched objects, sync aborted

12:14:18

Cron trigger failed

migrate-daily (exit code 1)

12:15:02

Retry completed

Sync succeeded with 5 skipped objects

12:15:19

Incident resolved

All systems operational

Probable cause

Resources in the source cluster were missing required labels ("team"), which led to object matching failures during sync. A subsequent cron trigger also failed because the previous migration run had not completed successfully.

Action taken

Added label defaults, fixed id mapping logic, and rescheduled the migration job. Sync completed successfully.

ТРАФИК · ДАННЫЕ · ОТКАТ

76

**Откат должен быть
проверен до переключения**

ПРАКТИКИ КОМАНДЫ



Семь практик

01 Говорите с разработчиками

02 Ведите паспорта стендов

03 Фиксируйте ресурсы

04 Поддерживайте документацию

05 Передавайте знания

06 Говорите о проблемах сразу

07 Не перевозите ненужное легаси



Говорите с разработчиками

Поведение

Что приложение делает
на самом деле

Зависимости

Что его перезапускает
и поддерживает

Исправление

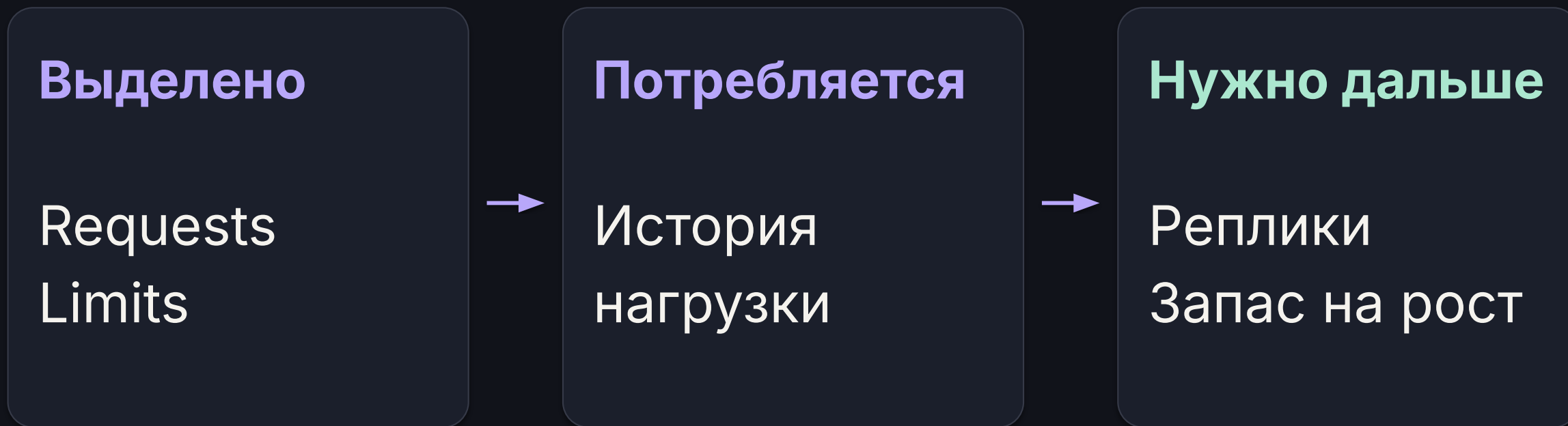
Дефект — в разработку,
а не в НОВЫЙ КОСТЫЛЬ



Ведите паспорта стендов

Кто	Владелец и критичность
Где	Контур и состав
С чем	Зависимости внутри и снаружи
Как проверить	Мониторинг и сценарий возврата

Фиксируйте ресурсы





Поддерживайте документацию

runbooks / migration.md

Действие

Что сделать

Проверка

Что увидеть

Остановка

Когда остановиться

Ответственность

Кто поможет

Документ должен помогать выполнить шаг

docs > runbooks > migration-to-k8s.md

Изменено 23 апр 2024, 14:27

Редактировать

...

Документация

Поиск...

🔍

Обзор

Архитектура

Руководства

Runbooks

Миграция в Kubernetes

Восстановление сервиса

Ротация секретов

Масштабирование

Аварийные сценарии

Справочники

Частые вопросы

Инфраструктурная команда

Все системы в норме

RUNBOOK

Миграция платформы в Kubernetes

Пошаговая инструкция по безопасной миграции сервисов в Kubernetes с проверками, возможностью отката и валидацией данных.

1. Подготовка

Убедитесь, что выполнены все предварительные шаги перед началом миграции.

✓ Резервная копия базы данных создана и проверена

✓ Конфигурации подготовлены (values, секреты, configmaps)

✓ Проведено тестирование в staging-окружении

✓ Утверждено окно миграции и уведомлены заинтересованные команды

☐ Проверена доступность rollback-скриптов

2. Выполнение миграции

Применяем инфраструктуру и деплоим приложение в целевой кластер.

bash

Применяем инфраструктуру

terraform apply -var-file=prod.tfvars

Деплоим приложение

helm upgrade --install platform ./charts/platform \

--namespace prod \

--set image.tag=1.28.4 \

--wait --timeout 10m

3. Smoke-тесты

Проверяем, что основные сценарии работают после миграции.

✓ Сервис отвечает по основному endpoint

✓ Проверены ключевые API (health, /api/v1/*)

✓ Фоновые задачи (cron, workers) выполняются

✓ Проверен доступ к внешним зависимостям (БД, очереди, S3 и т.д.)

Прогресс выполнения

4 из 5 шагов завершено

80%

Содержание

1. Подготовка

2. Выполнение миграции

3. Smoke-тесты

4. Валидация данных

5. Откат (rollback)

6. Постмиграционные проверки

7. Возможные проблемы

8. Полезные команды

Параметры

Окружение

production

Владелец

SRE Team

Последнее обновление

23 апр 2024, 14:27

Версия

1.4

Связанные страницы

Архитектура платформы

Чек-лист релиза

Аварийное восстановление

Мониторинг и алерты

ПРАКТИКА 4 / 7 · КОМАНДНЫЕ ПРАКТИКИ

84



Передавайте знания

Один всё знает	Архитектура и действия в одной голове
-----------------------	--

Он недоступен	Отпуск, болезнь или уход из команды
----------------------	--

Процесс встал	Команда не может продолжить сама
----------------------	-------------------------------------



Вместо незаменимости — передаваемые знания

Документация

Шаги в общей базе знаний. Изменения проверяют коллеги.

Парная работа

Сложные этапы проходят минимум два инженера.

Обмен опытом

Регулярные демонстрации и разборы архитектуры.

Переезд должен пережить отпуск ключевого инженера.



Говорите о проблемах сразу

До миграции

Нет доступа

Неясная зависимость

Нет сценария отката

В ночь миграции

Те же проблемы

Но уже срочно

И с риском простоя

Риски — в задачи до миграции

MIGRATION

ДоскаСписокОтчёты

Фильтр ▾

● К ВЫПОЛНЕНИЮ2

Найти владельца

Нет ответственного

■ MIG-101Ops

Уточнить зависимость

Связь и способ проверки

■ MIG-102Dev

● В РАБОТЕ2

Проверить доступы

Права до окна миграции

■ MIG-103Ops

Описать откат

Сценарий возврата

■ MIG-104DBA

● ГОТОВО2

Документация

Доступна всей команде

■ MIG-105Ops

Состав контура

Сверен с паспортом

■ MIG-106Dev

Не перевозите ненужное легаси

Старая площадка

Мёртвые переменные
Старые секреты
Обходные решения

Без аудита

Та же конфигурация
Те же зависимости
Тот же технический долг

**Знания о системе
должны принадлежать
команде**

ОШИБКИ

ОШИБКА 1

Один релиз — две разные роли

API

name: backend
instance: stage
component: api

Бизнес-API

WORKER

name: backend
instance: stage
component: worker

Служебный HTTP

— name и instance одинаковы. Роль задаёт компонент.

Service выбирает не только API



У обоих есть /health. Бизнес-маршруты — только у API.

Одна строка меняет набор получателей

charts / service.yaml · diff

```
# Service backend-api
spec:
  selector:
    app.kubernetes.io/name: backend
    app.kubernetes.io/instance: stage
+   app.kubernetes.io/component: api
```

API + worker



Только API

Проверяем выбор селектора

selector / demo

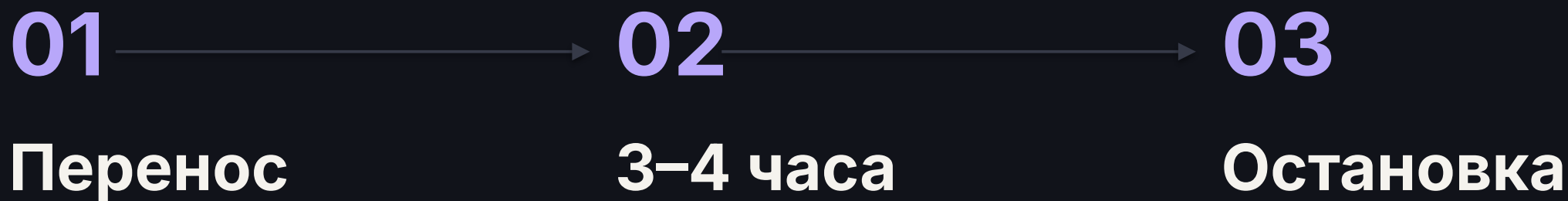
```
$ kubectl -n demo get pods \
  -l app.kubernetes.io/name=backend \
  -L app.kubernetes.io/component
```

NAME	COMPONENT
api-01	api
api-02	api
worker-01	worker

Сверяем полный selector Service и EndpointSlice

ОШИБКА 2

На dev контур приложение «заснуло»



Новый контур.
Приложение
запущено.

Некоторое время
всё работало.

Процессы встали.
Новых логов нет.

Симптом — нет полезной работы и нет новых логов.



Переехали без внешнего cron

Jump host
cron + kubeconfig



Периодический
перезапуск



Приложение
со старым дефектом

ПЕРЕДАЧА

Об обходном решении
не сообщили при передаче.

ПОСЛЕ ПЕРЕНОСА

Дефект больше
не маскировался.



Исправили дефект в тот же вечер

**Проблема передана
разработчикам**



**Дефект исправлен
в тот же вечер**

ЧТО ПОМОГЛО

**Сразу обсудили поведение
с командой разработки.**

ВЫВОД

**Внешние зависимости
нужно передавать явно.**

ОШИБКА 3



Strapi — система управления контентом

Headless CMS

Контент управляется отдельно
и передаётся приложениям через API

strapi.io

Повторный запуск затёр контент

01**02****03****Перенос****~4 часа****Повторный
запуск**

Контент перенесли
в новый контур.

Всё работало
как ожидалось.

Контент оказался
затёрт.

Strapi: точный механизм потери контента не установили.

Контент вернули из старого бакета

СТАРЫЙ БАКЕТ

- ☐ content / object-01
- ☐ content / object-02
- ☐ content / object-03

Оставили в резерве

ВОССТАНОВЛЕННЫЙ КОНТУР

- ☐ content / object-01
- ☐ content / object-02
- ☐ content / object-03

Повторно перенесли

— Старый бакет не удалили — из него восстановили контент.



Репетиция включает повторный запуск

Первый перенос

Проверить результат и поведение приложения.

Повторная операция

Понять, что меняется при повторе миграции.

Восстановление

Сохранить источник до окончания проверок.

Первый успешный запуск не проверяет все сценарии.

**Ошибка полезна,
когда после неё
меняется процесс**

ИТОГИ



Набор для следующего переезда

Знание системы

- 01 Техническая документация
- 02 Паспорт стенда
- 03 Карта интеграций
- 04 Карта зависимостей
- 05 Ресурсный профиль



Набор для следующего переезда

Знание системы

- 01 Техническая документация
- 02 Паспорт стенда
- 03 Карта интеграций
- 04 Карта зависимостей
- 05 Ресурсный профиль

Управление процессом

- 06 План миграции
- 07 План отката
- 08 Дорожная карта
- 09 Владельцы
- 10 Критерии успеха и отката

После команды остаётся рабочий набор

migration / readiness

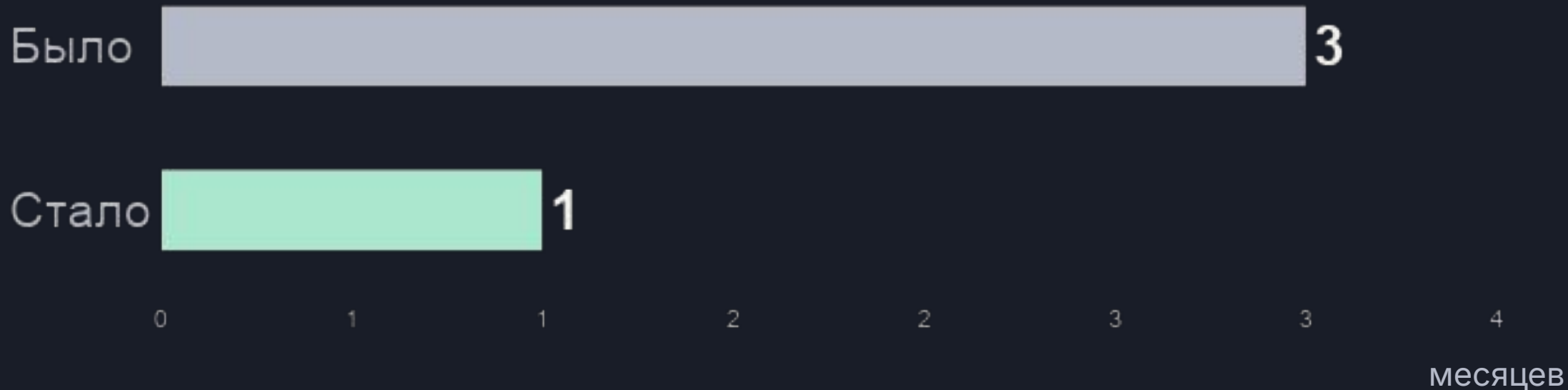
01 Знание системы Документы · паспорта · зависимости

02 Ресурсы Профиль фактического потребления

03 План действий Миграция · дорожная карта · владельцы

04 Решение Критерии успеха и отката

Раньше — ~3 месяца, позже — ~1 месяц



Оценка длительности технического переезда из опыта команды.



Изменились не только сроки

Было

- Ручные действия и проверки
- Знания в головах
- «Переключим и посмотрим»
- Героизм на развертываниях



Изменились не только сроки

Было

- Ручные действия и проверки
- Знания в головах
- «Переключим и посмотрим»
- Героизм на развертываниях

Стало

- IaC / Helm / GitOps
- Паспорта и чек-листы
- Автотесты и план отката
- Предсказуемый процесс



Что действительно изменилось

Автоматизация

Единый деплой.
Автопроверки.
Меньше ручной
рутины.

Знания

Паспорта, карты
интеграций и общий
набор документов.

Предсказуемость

План отката,
чек-лист и понятные
критерии решения.



Что хочется сказать после переезда?

«Мы герои!»

- Стресс и ночные дежурства
- Тушение пожаров
- Спасение личным временем

Хороший переезд не должен становиться испытанием для команды.



Что хочется сказать после переезда?

«Мы герои!»

- Стресс и ночные дежурства
- Тушение пожаров
- Спасение личным временем

«Было скучно»

- Понятная последовательность
- Автоматизация и проверки
- Без необходимости подвига

Хороший переезд не должен становиться испытанием для команды.



Что проверить в своём контуре завтра?

Зависимости

Что управляет сервисом снаружи?

Повторяемость

Сможет ли другой инженер повторить действия?

Возврат

Есть ли проверенный сценарий неудачи?

**Не копировать архитектуру
Забрать полезные практики**

А что в вашем контуре может сорвать переезд?

Зависимости

Данные

Безопасный откат

Пять переездов. Ошибки, выводы и практики, которые можно адаптировать.

