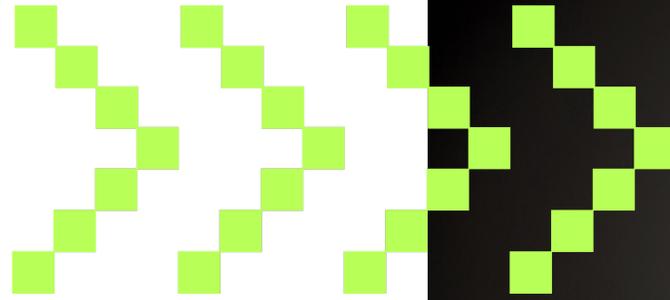


Счастье as a Service

HolyJS, 27 апреля 2024 года

Кто здесь?



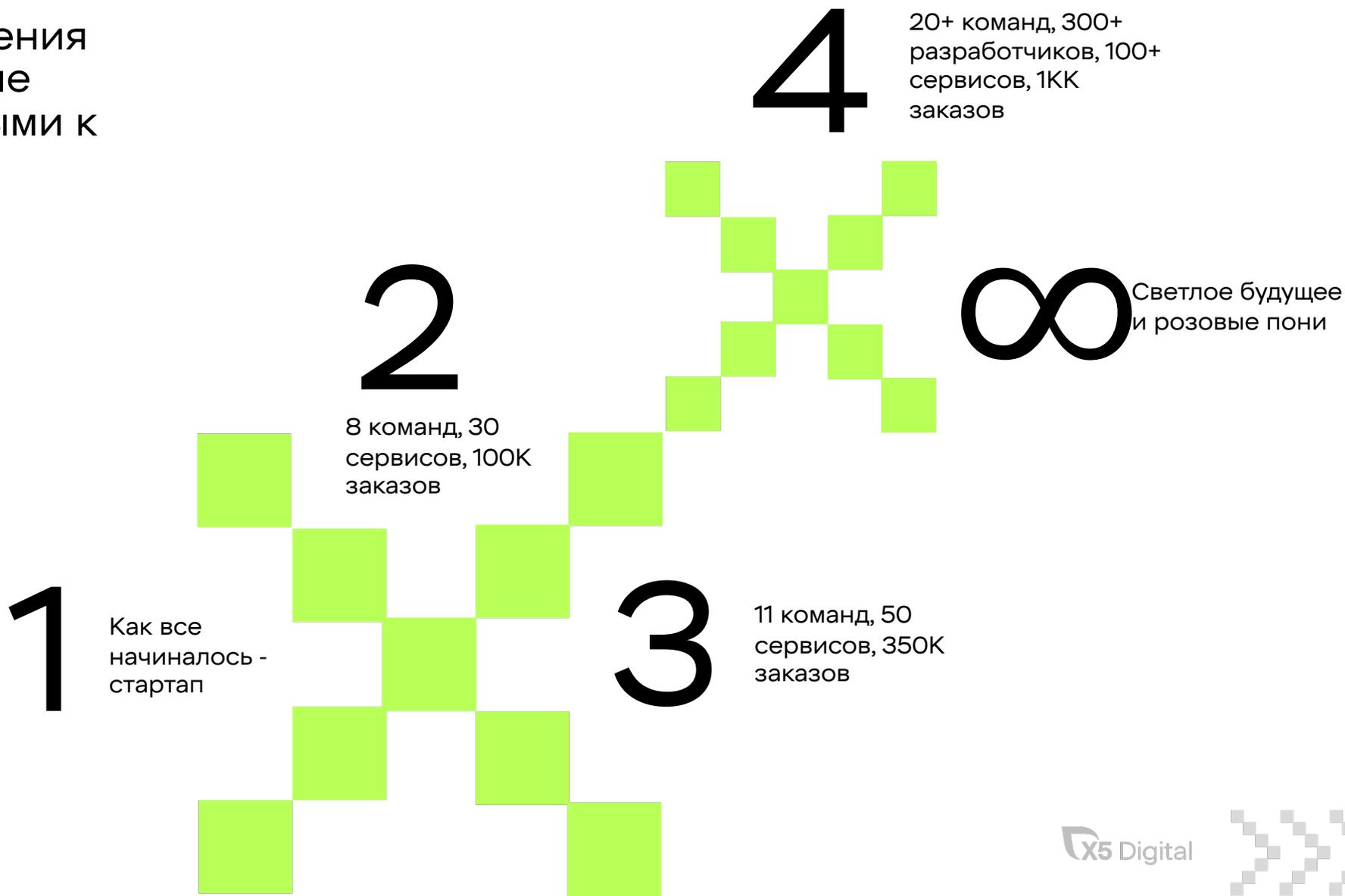
Саша Буланов

- руковожу командой CI/CD
- много общаюсь с разработчиками и тестировщиками
- цель – сделать их жизнь лучше и проще
- иногда имею проблемы с арифметикой

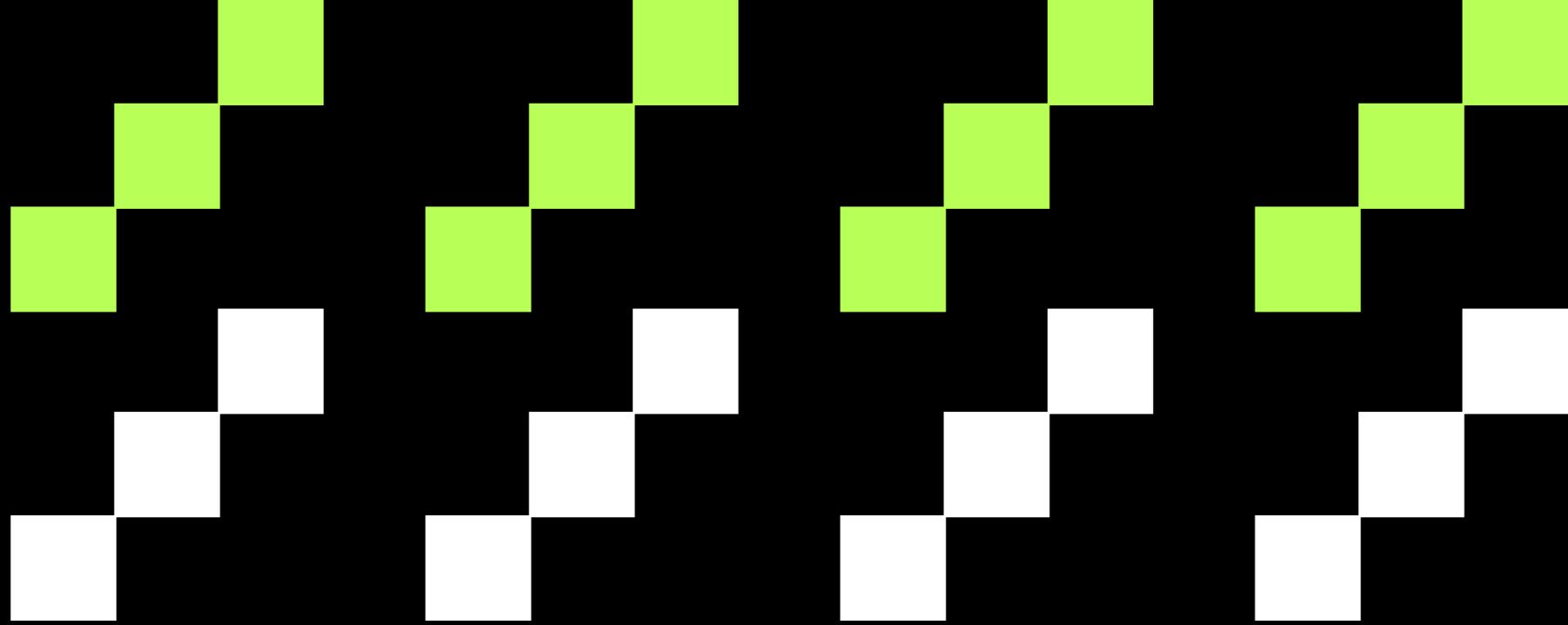


Что будет?

Расскажу, какие решения помогли нам в нужные моменты быть готовыми к кратному росту



0.0



X5 Digital



Мы – часть крупнейшего
ритейлера

Развиваем цифровые продукты для экспресс-
доставки из наших торговых сетей и дарксторов

69

Регионов

100+

Сервисов

7500

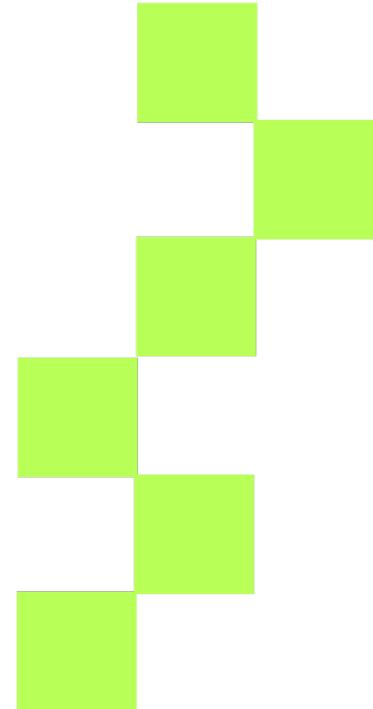
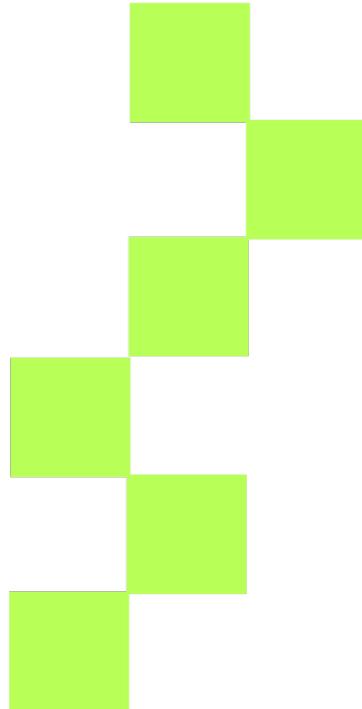
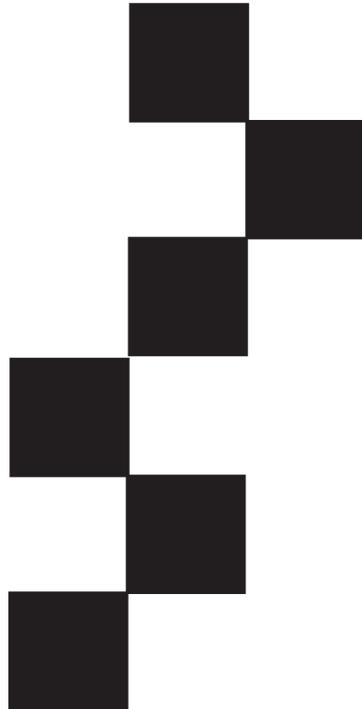
RPS

300K+

Заказов



0.1

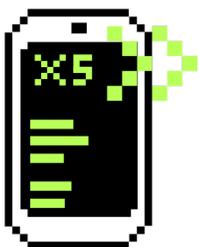


Как принято



Continuous Delivery

Рецепт идеальной
доставки приложений



Собрать и доставить код



Подготовить и определить нужные
инфраструктурные сервисы



Донести и не расплескать секреты

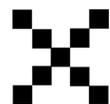
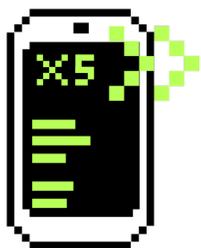


С минимальным участием человека

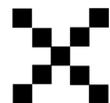


Continuous Delivery

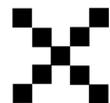
Что может пойти не так?



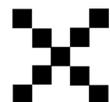
Отдельная область экспертизы для разработчиков



Дополнительные сложности при найме



Лишние усилия на поддержку

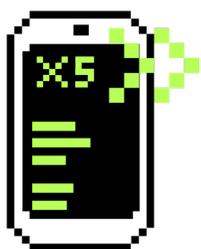


Увеличение Time To Market

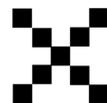


Continuous Delivery

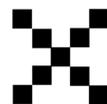
Основная проблема



DevOps ради DevOpsa



Кем-то придуманные абстракции

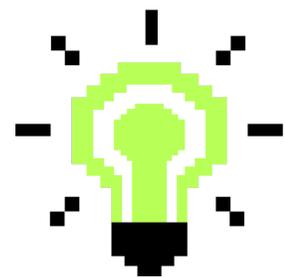


Вы пробовали это отлаживать?



Continuous Delivery

А что решаем?



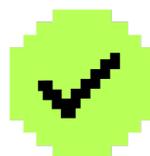
А можно так, чтобы разработчики не знали о том, как развернуть их сервисы?



Чтобы DevOps не знали о различиях в сервисах?



Можно было оперировать понятными нам абстракциями?



И чтобы всем было удобно?

Что, если...

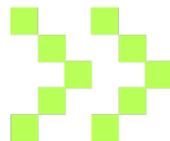


Continuous Delivery

Мы же программисты!



Не любим YAML,
хорошо умеем Node.js

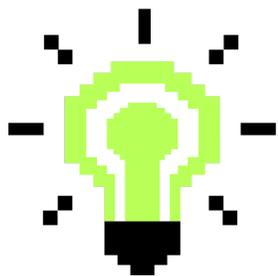


Хорошая архитектура –
взаимозаменяемость разработчиков

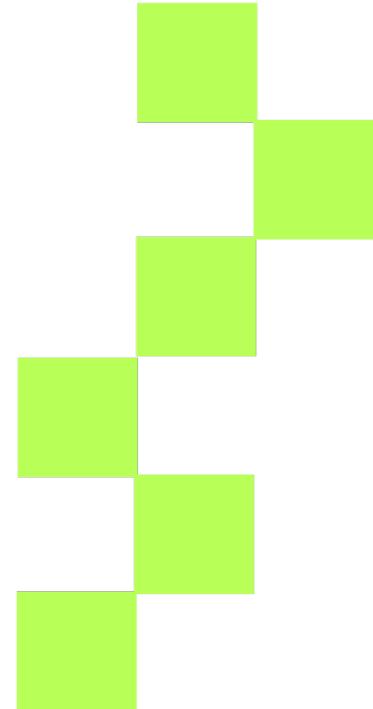
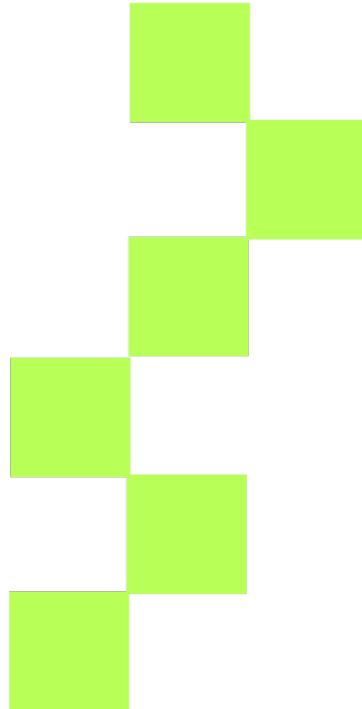
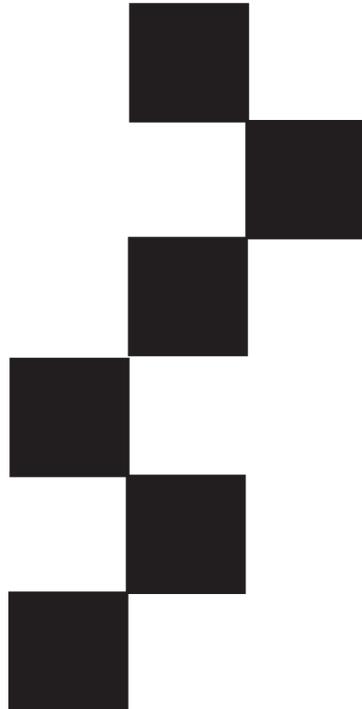


Гибкая логика
и дополнительные абстракции

Конечно же да!



1.0



Кажется,
начинается...



А с чем предстоит иметь дело?



Нас мало, нам хватит

- 1 инстанс БД
- отдельная база для каждого сервиса
- для удобства запускаем в pm2
- 1 стенд со всеми сервисами для всех разработчиков



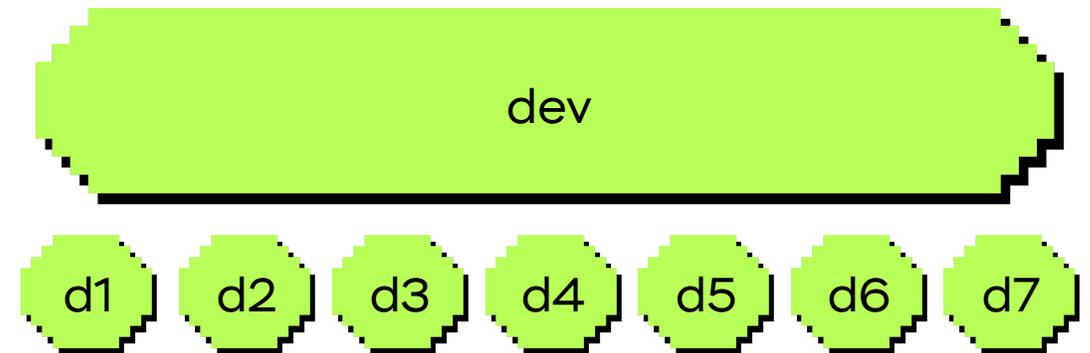
Нас мало, нам хватит

- управление БД – абстракция базы данных, создание баз и секретов для сервисов
- генерация конфига для каждого сервиса
- управление pm2 на серверах – абстракция сервиса и инстанса, агент на серверах
- абстракция стенда



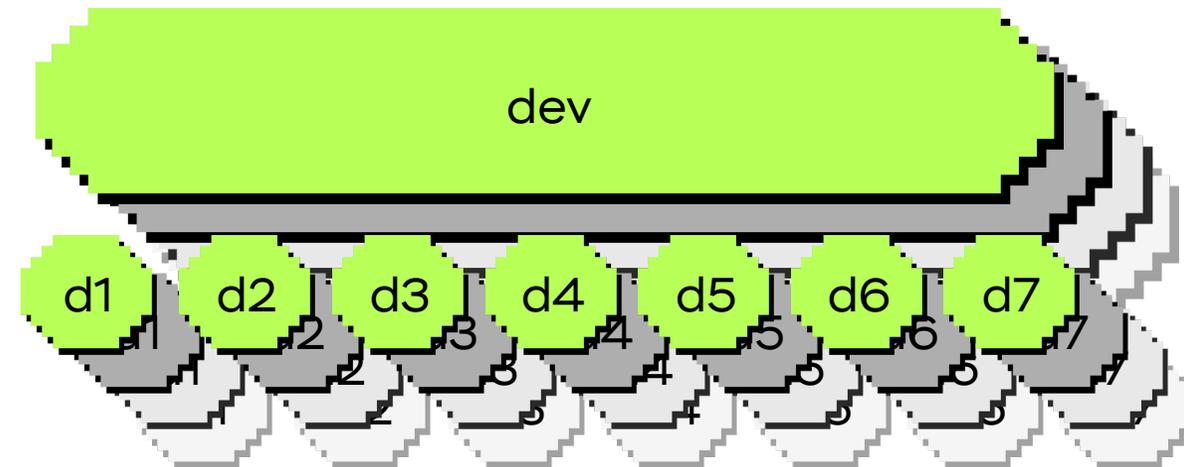
15 человек начали толкаться локтями

- отдельный стенд для каждого разработчика
- 1 инстанс БД
- отдельная база для каждого сервиса на каждом отдельном стенде
- для удобства запускаем в pm2
- можно запускать только нужные сервисы, остальные используются с dev



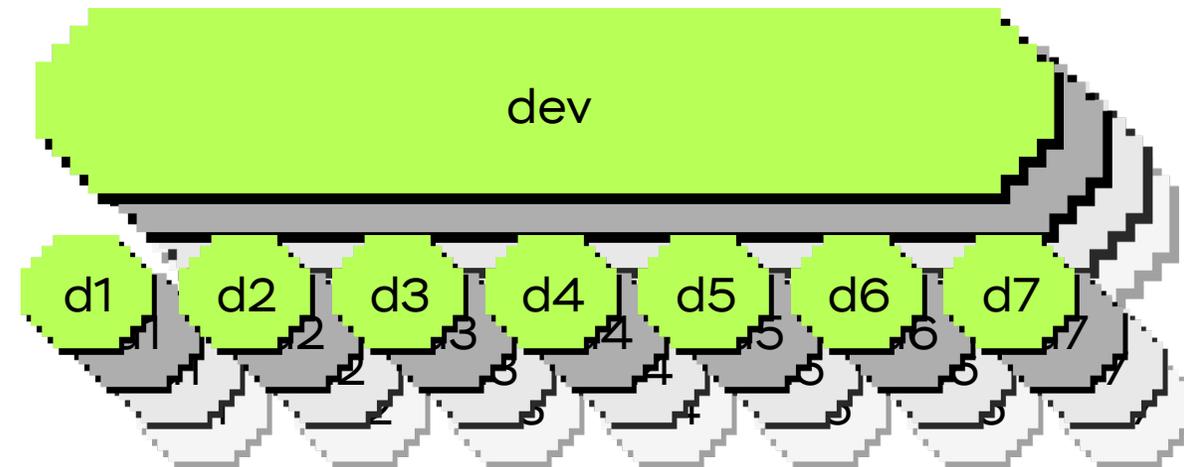
Агрегатор, торговые сети

- слишком разные, сложно смешивать данные
- отдельный стенд для каждого разработчика в каждом контуре
- 3 инстанса БД
- отдельная база для каждого сервиса на каждом отдельном стенде
- для удобства запускаем в pm2
- можно запускать только нужные сервисы, остальные используются с dev



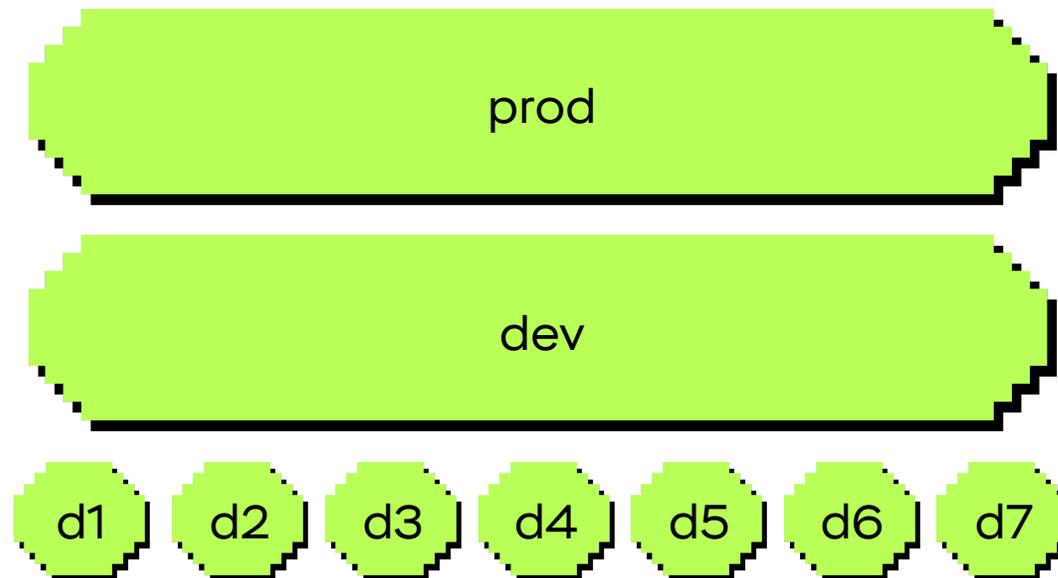
Агрегатор, торговые сети

- работа с тремя контурами – абстракция маркета
 - разделение по сети
 - работа с разными наборами секретов
 - работа с несколькими DNS доменами



В продакшн!

- Повторяем стенд dev с некоторыми ограничениями
- 1 инстанс БД
- отдельная база для каждого сервиса
- хочется повторяемости и масштабирования – docker
- одна реализация для всех трех контуров
- много серверов, много инстансов сервисов – держим первую нагрузку



Первые проблемы

Куда без них



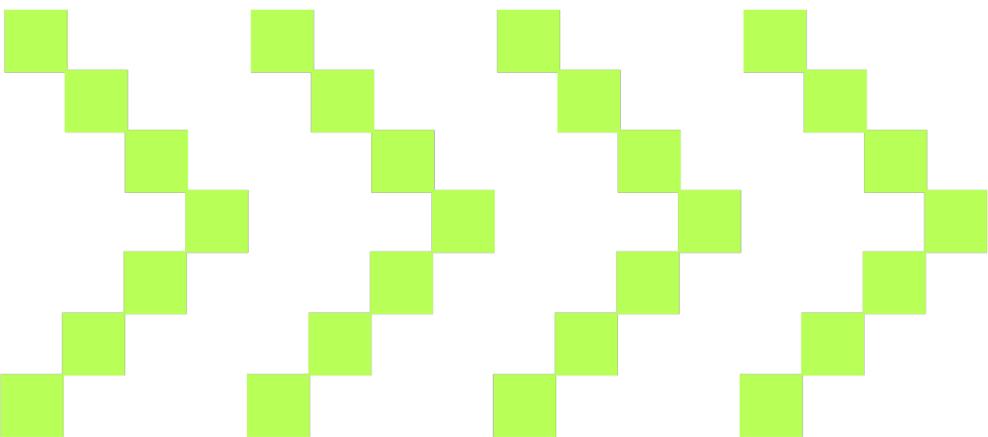
Нужны разные инстансы
БД в проде



Не очень удобно жонглировать
докерами и серверами

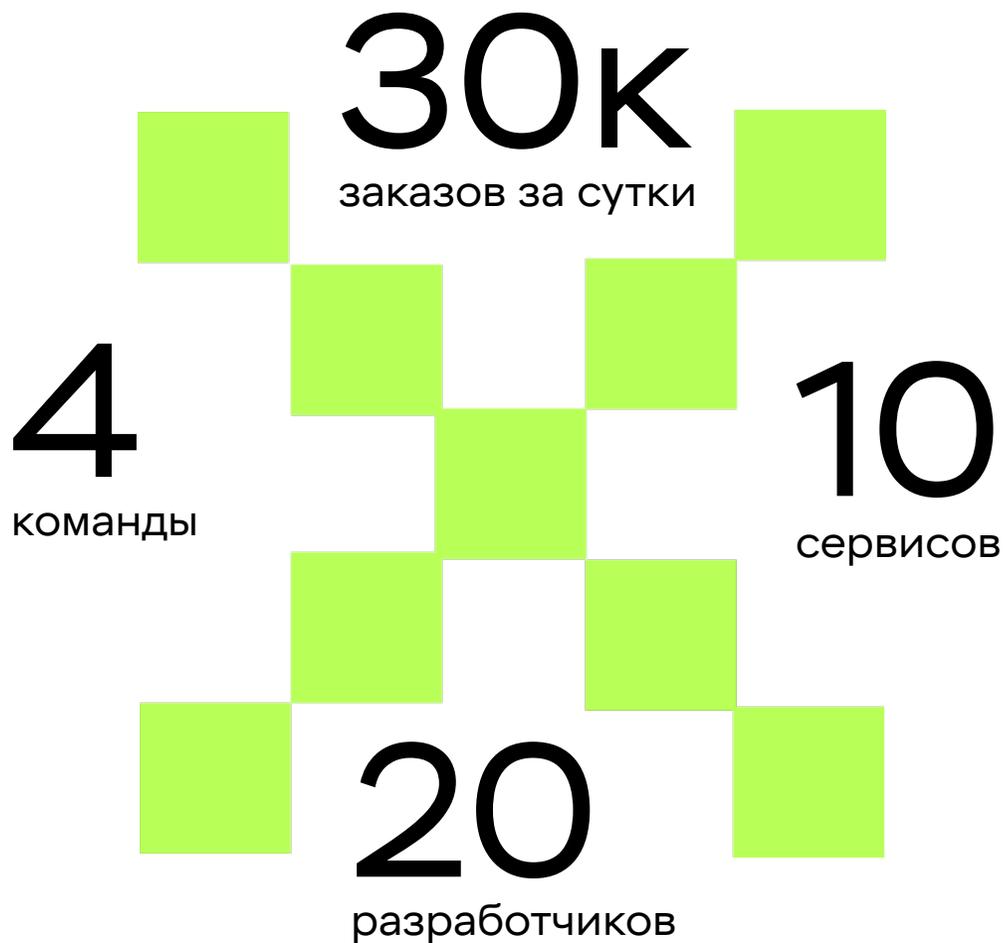


Всплески ошибок при выкатках

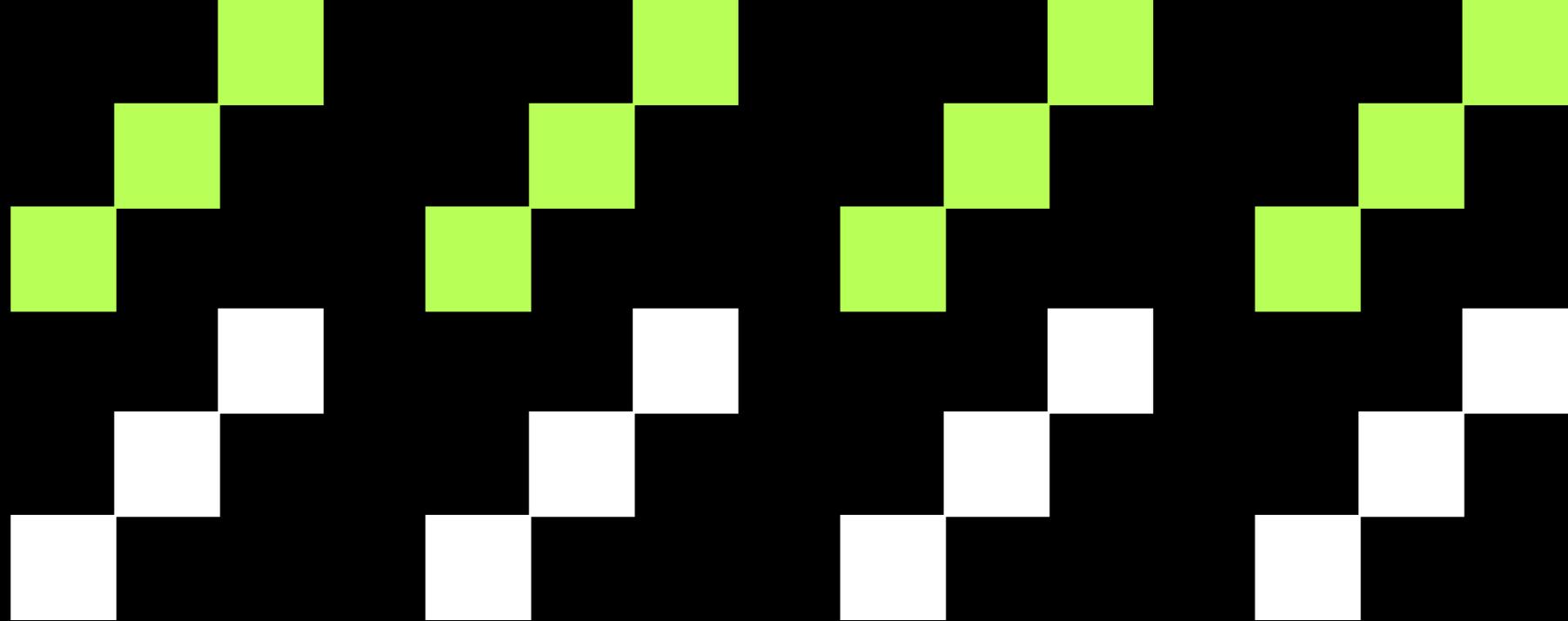


Конец 2021 года

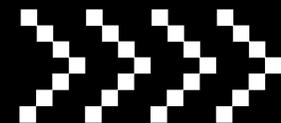
- разработка Deployer без выделенных людей
- 1 общий dev стенд, 20 персональных стендов



2.0

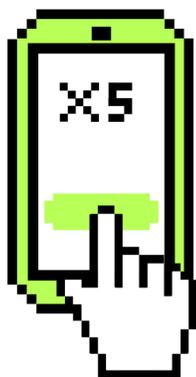
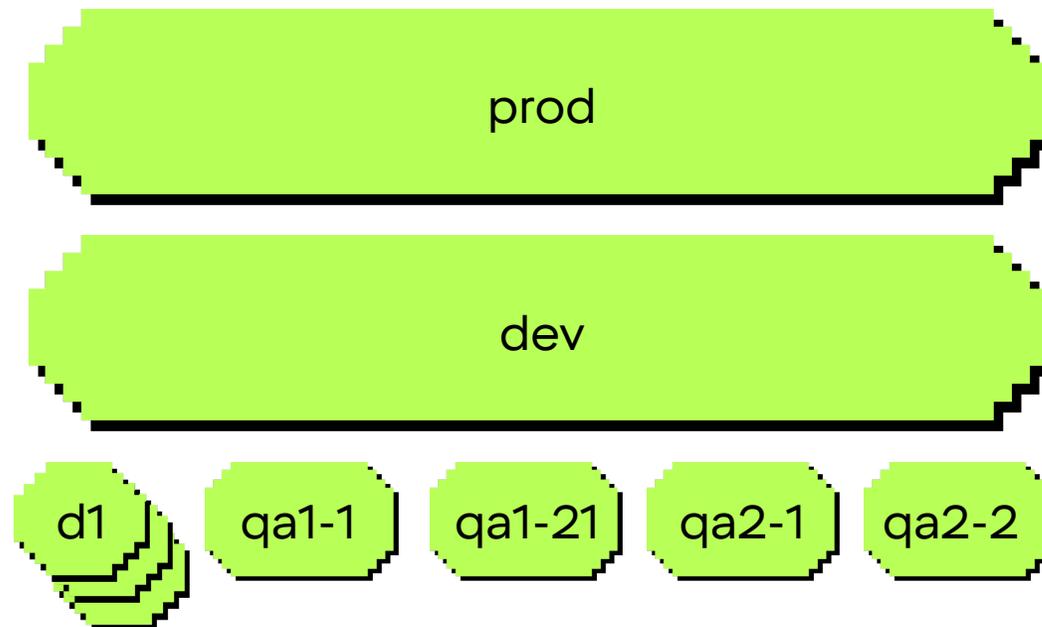


Deployer v2. Пытаемся расти



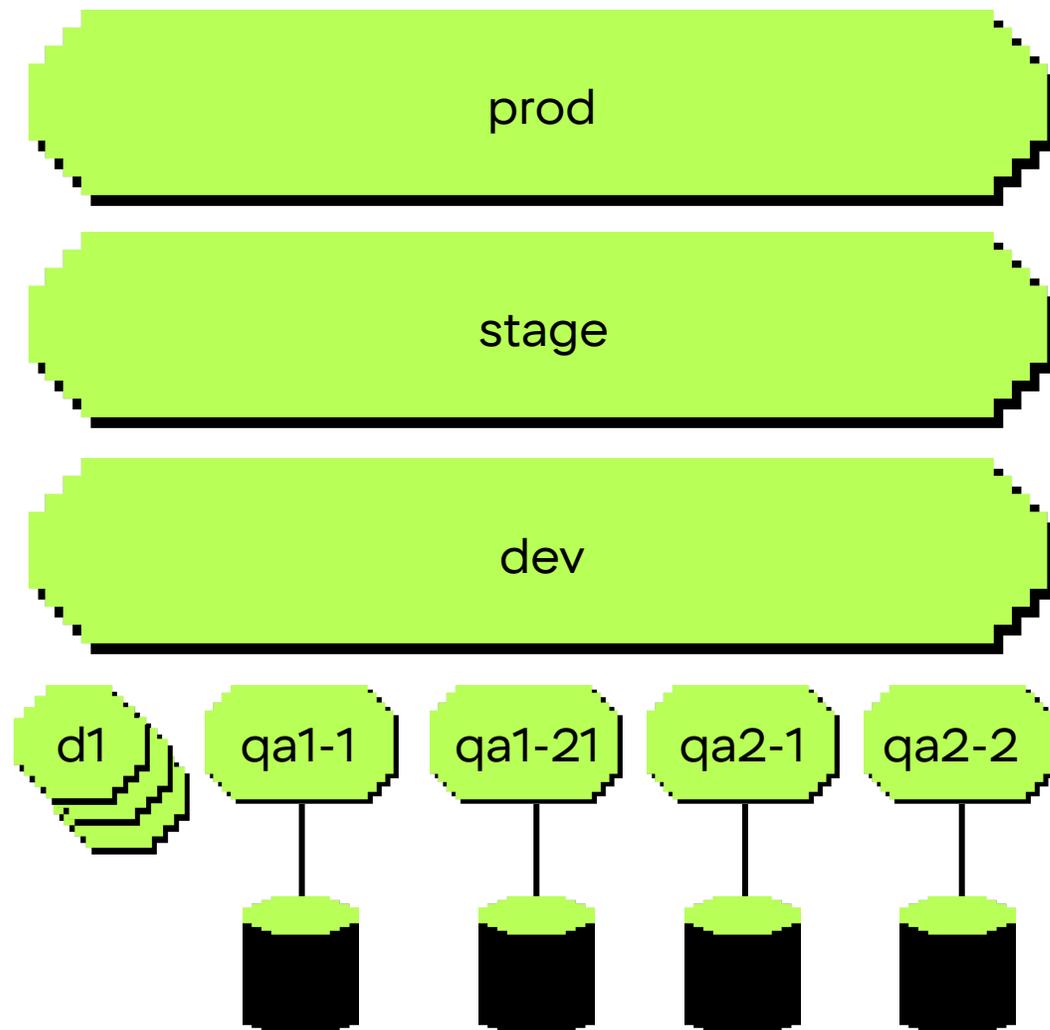
Большое тестирование

- для тестировщиков нужны выделенные стенды
 - шаблоны
 - автоматическое обновление сервисов и данных



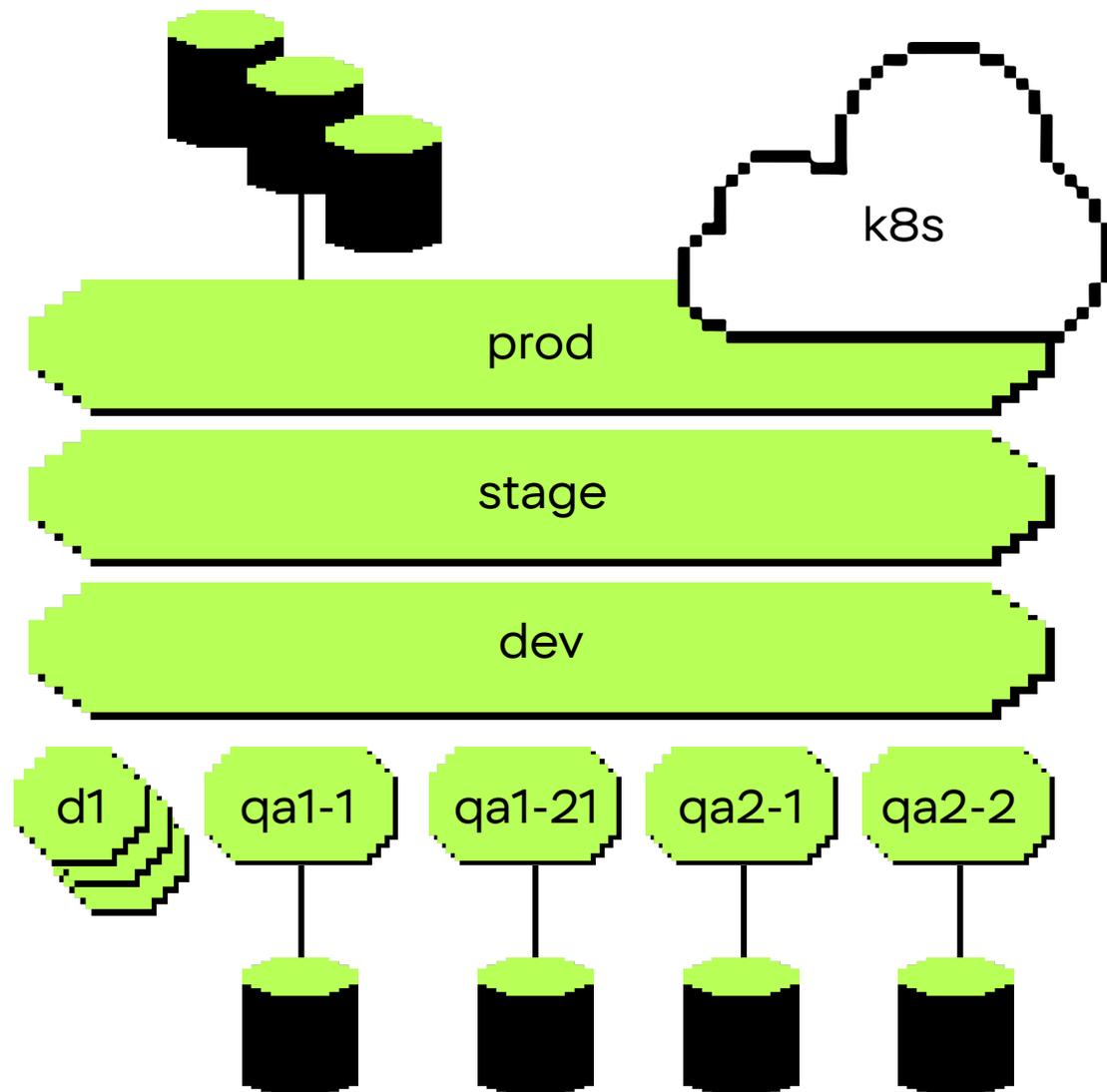
Не мешать друг другу

- на пустых базах не получается тестировать
 - MasterDB
 - первые попытки сброса данных



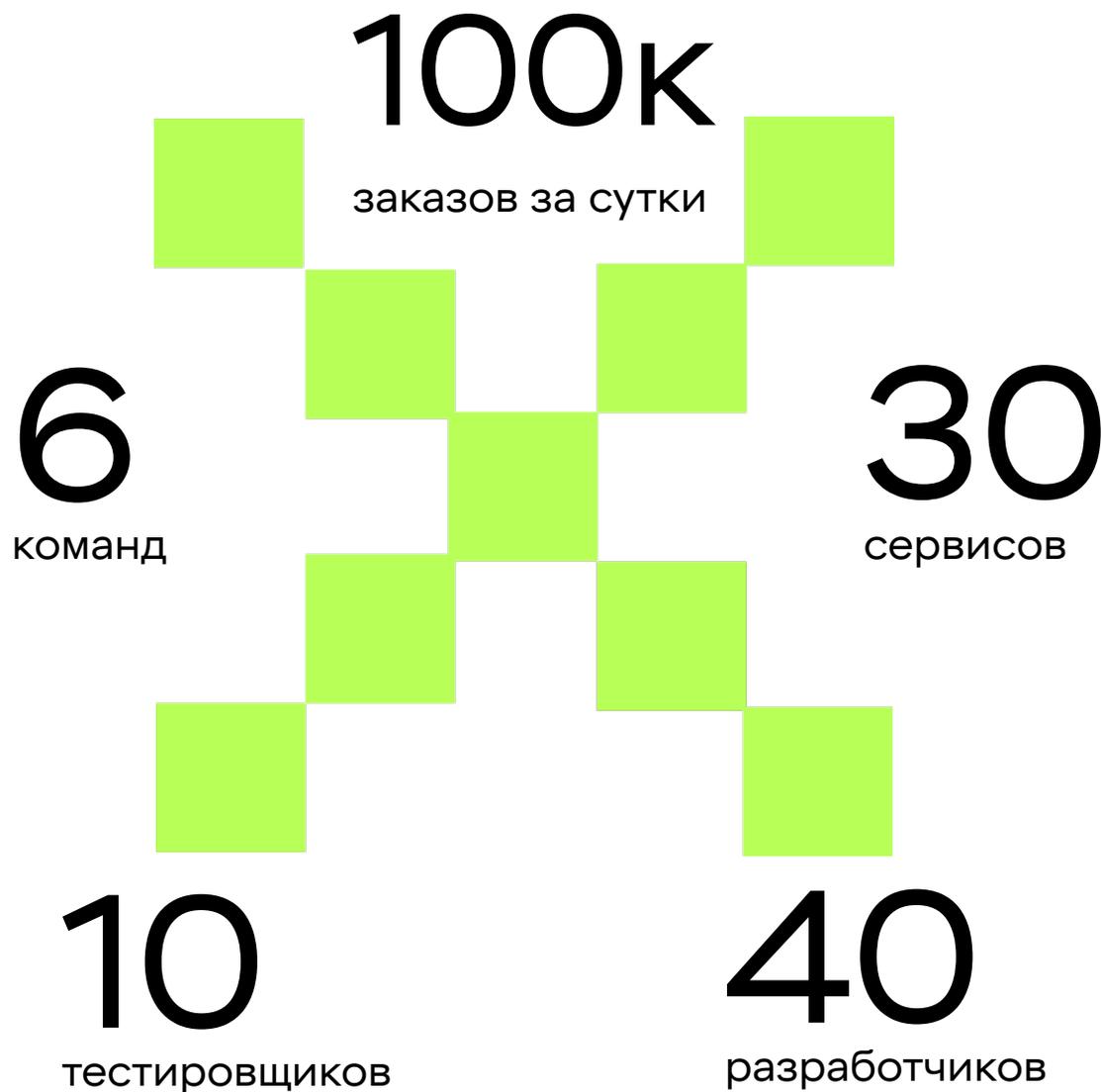
Больше нагрузки

- отдельные базы для сервисов в проде
- kubernetes в проде
 - просто отдельный тип инстанса

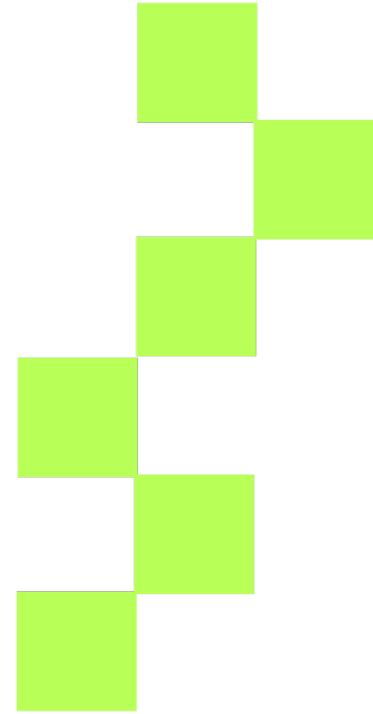
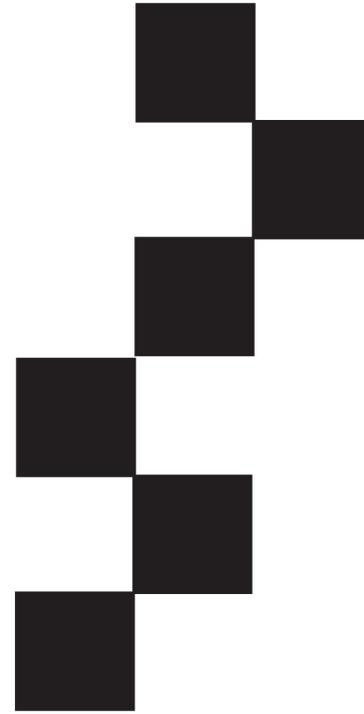
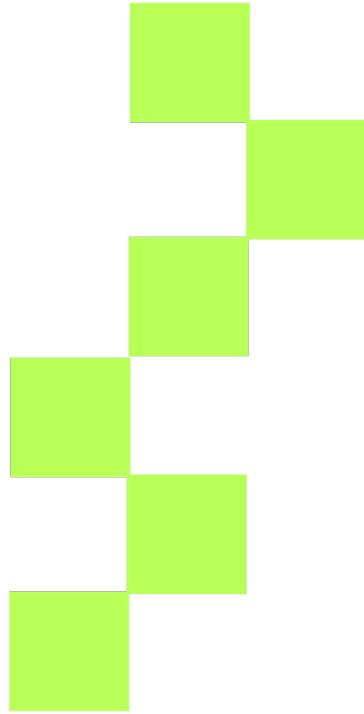
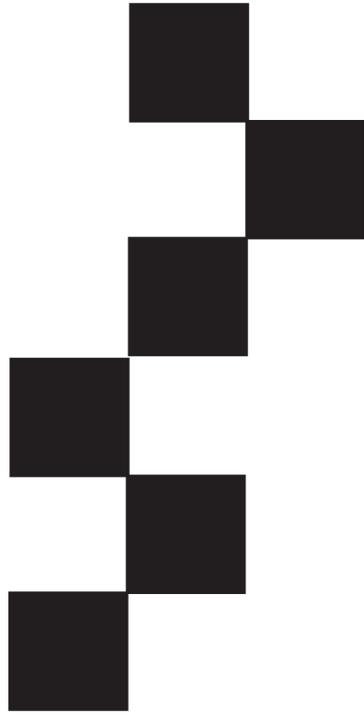


Конец 2022 года

- один выделенный разработчик Deployer
- 2 общих стенда – dev и stage
- 40 персональных стендов
- 20 стендов для тестирования



3.0

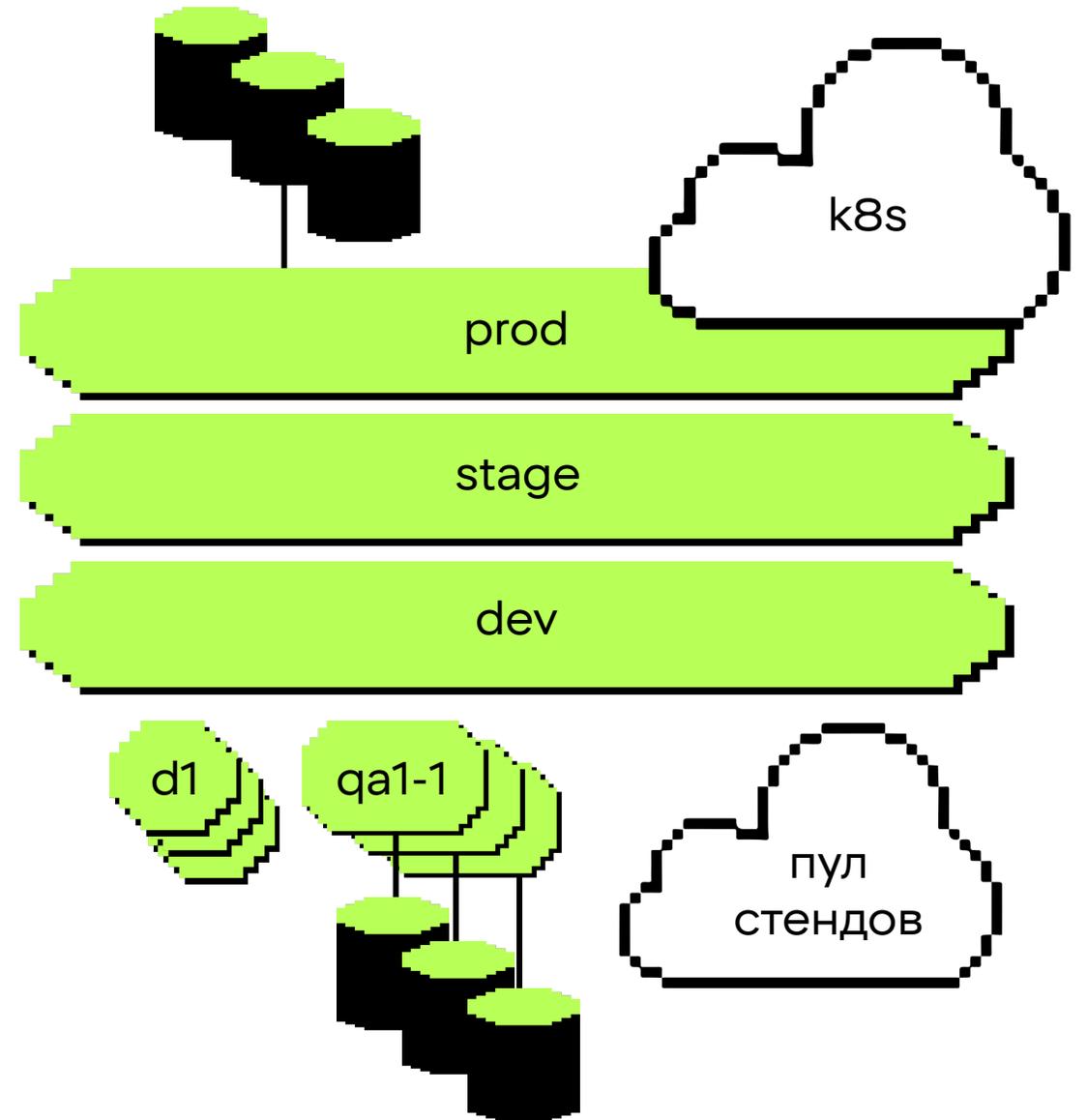


Deployer v3. Объять необъятное



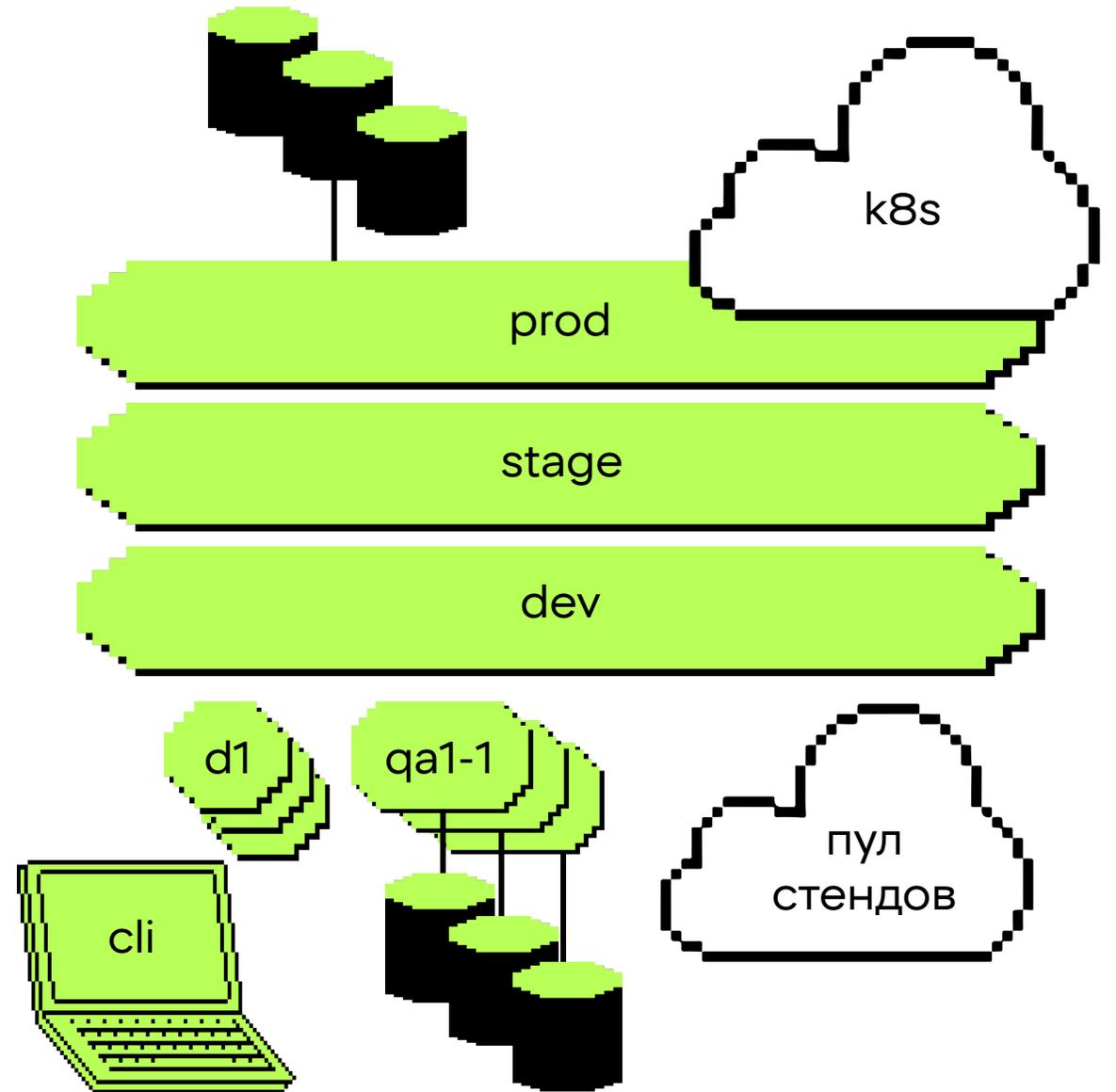
Автоматическое тестирование

- интеграционные тесты
- повторяемость, надежность и быстрота выполнения
- пул стенов
 - время выдачи стенов примерно равно скорости сборки самого сервиса
 - не нужно беспокоиться о сбросе данных
- большая часть dev окружения – в своем кластере k8s



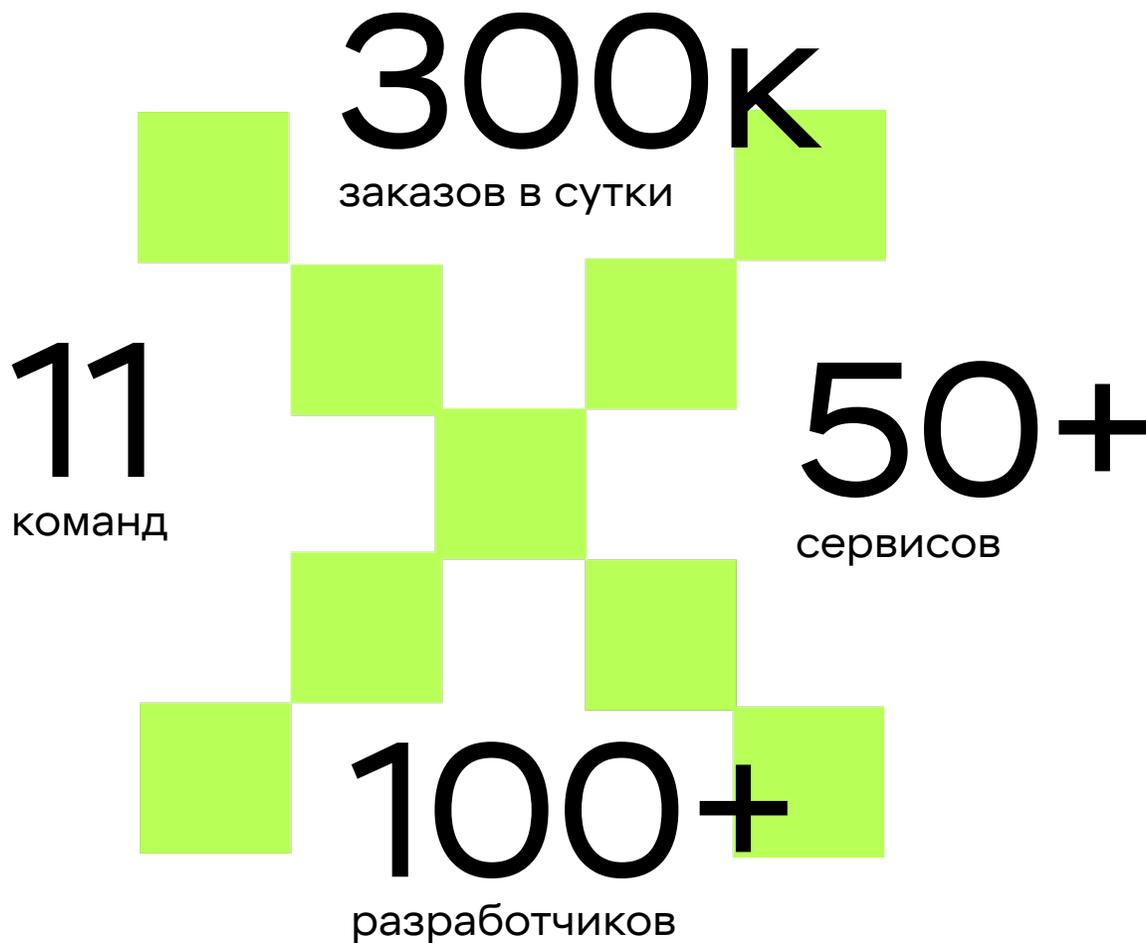
Размножаемся почкованием

- сервисы начинают дробиться на мелкие
- добавляем инструмент кодогенерации
- добавляем автоматическое тестирование для контрактов данных

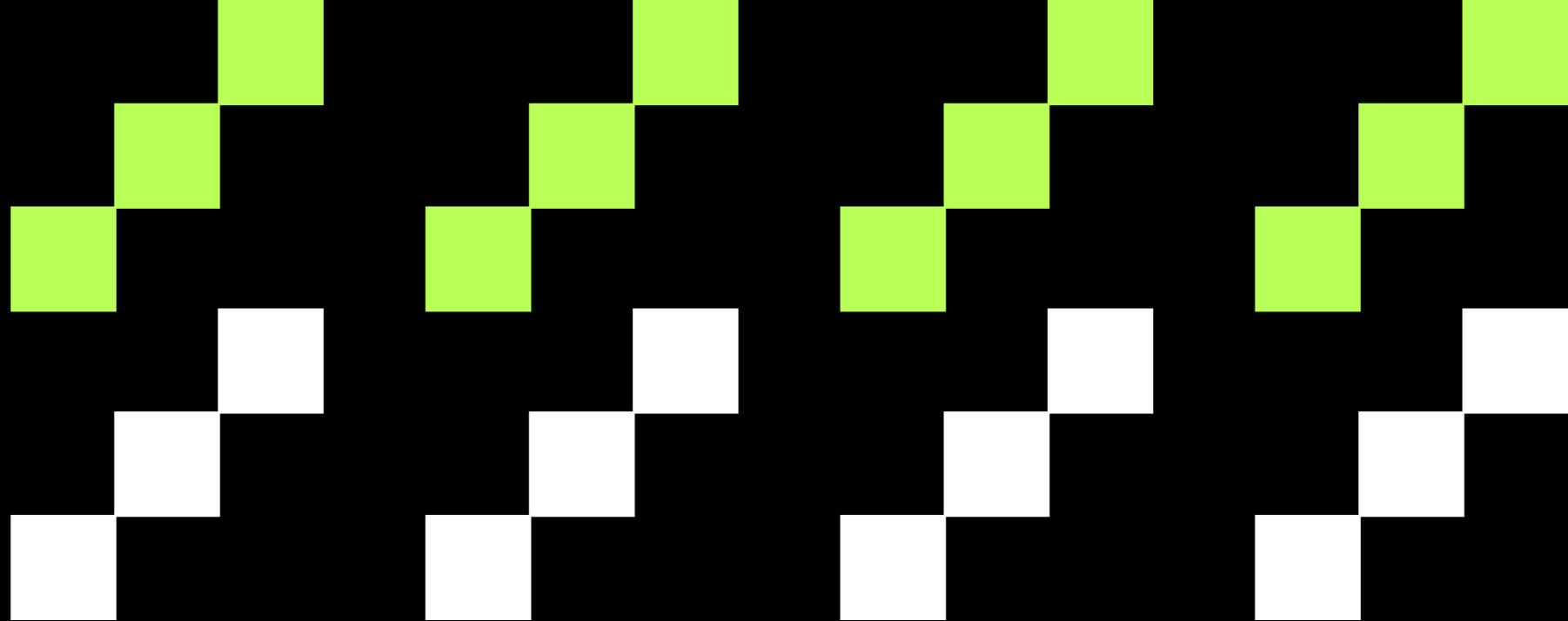


Конец 2023 года

- три выделенных разработчик Deployer
- 3 общих стенда
- 70 персональных стендов
- 70 стендов для тестирования



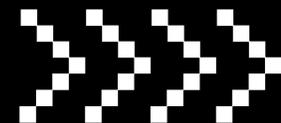
4.0



А что дальше?

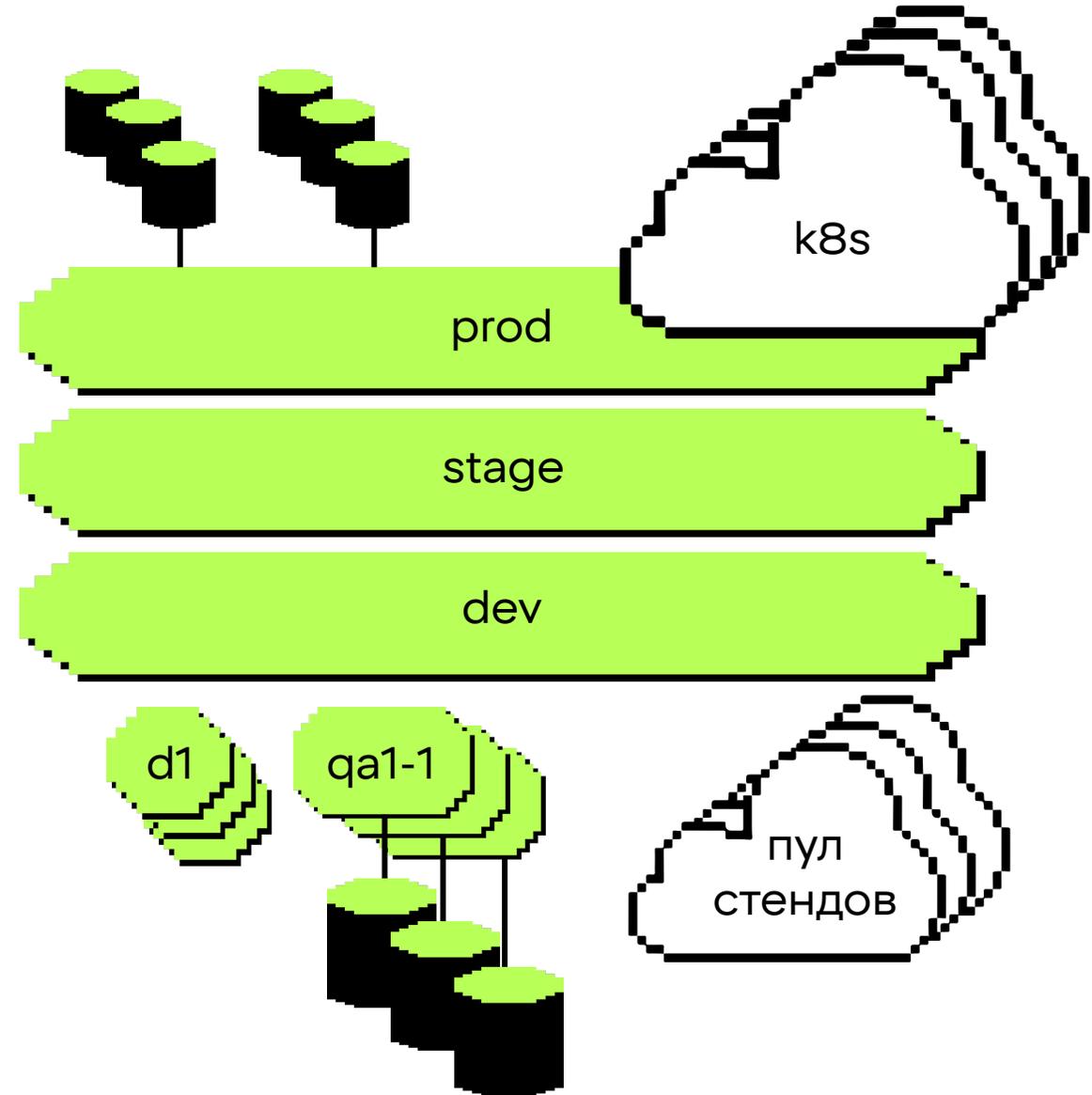
Еще расти?

А мы готовы!



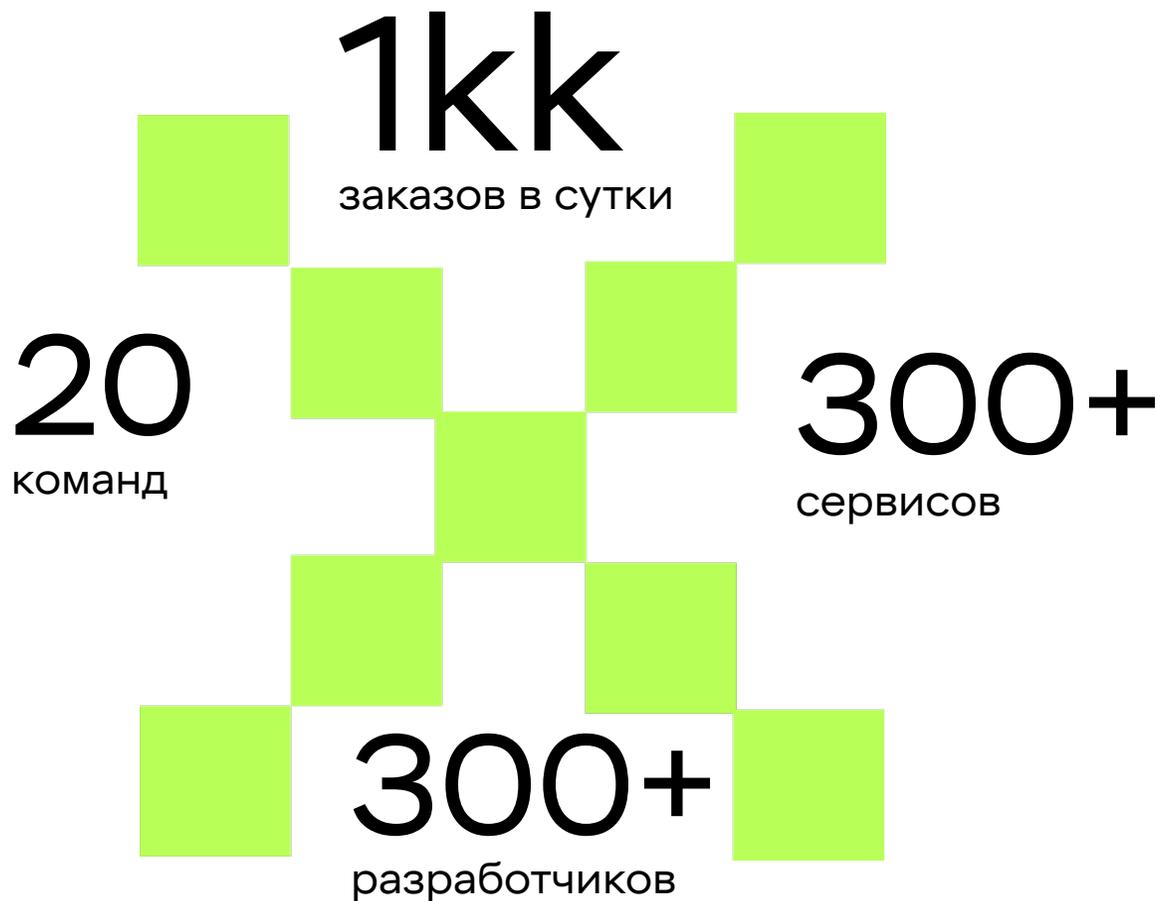
Увеличение объемов и нагрузки

- вертикально доросли, учимся шардировать
- отдельные динамические стенды для людей
- двигаем кубы – поддержка нескольких кластеров k8s

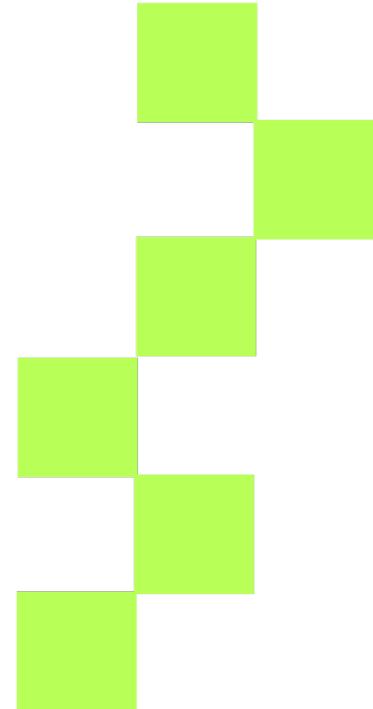
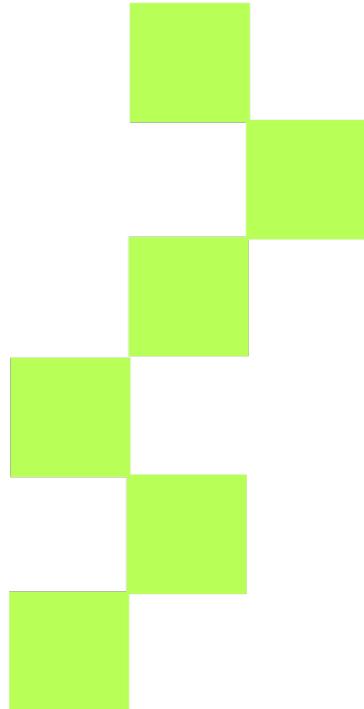
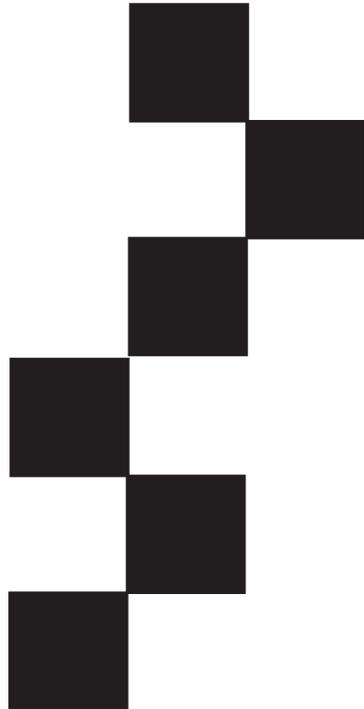


Готовимся к пока страшному

- четыре выделенных разработчиков Deployer
- инженер подготовки тестовых данных
- 3 общих стенда
- полный переход на динамические стенды



5.0



Немного о ЖИВОТНЫХ.
Я обещаю пони



Мечты, мечты



Локальная разработка,
доступные сервисы



Повелевать наборами
тестовых данных



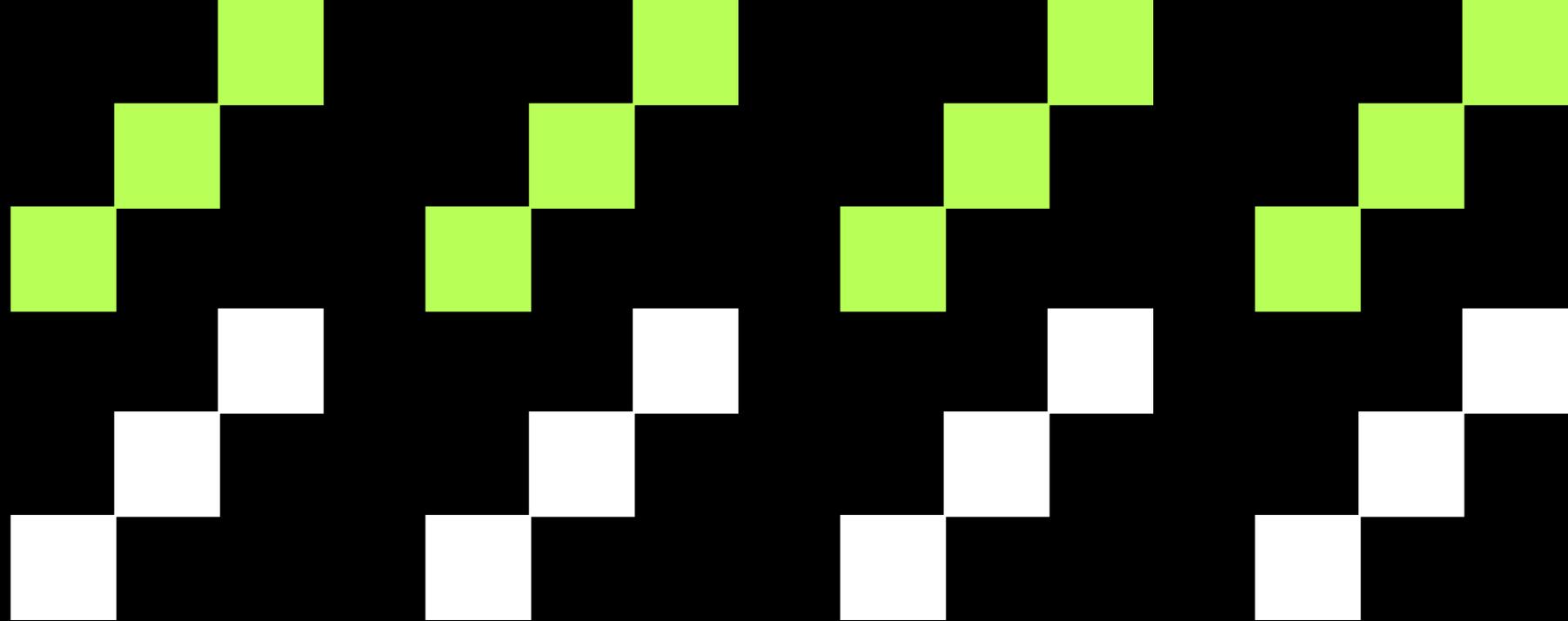
Улучшить observability



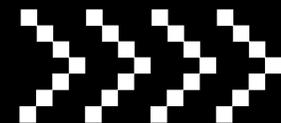
И все это – прозрачно
и незаметно для разработчиков



6.0



Плюсы и минусы



Чего хорошего?



Убрали необходимость дополнительной экспертизы разработчиков



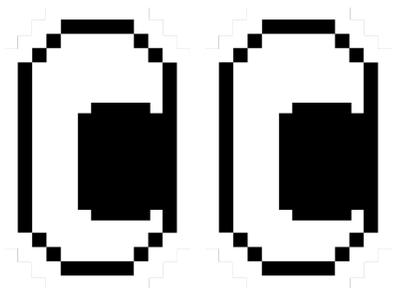
Есть удобные высокоуровневые абстракции



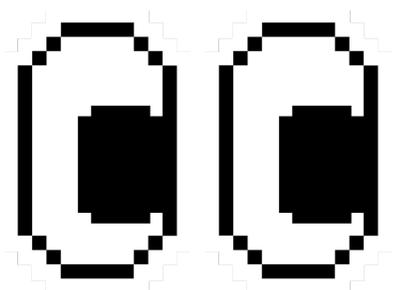
Лучше гранулированность прав доступа, сложно сломать



Умеем расти во всех направлениях без каких-то сложностей



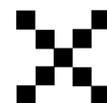
Бочка дегтя



Это нужно писать самостоятельно



Пути интеграции бывают сложнее
обычного использования инструментов



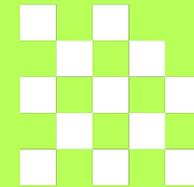
Скорость изменений немного
ниже – сначала нужно реализовать



Разработчики без Deployer
потом не очень любят жить



Огромное
спасибо!



@salsaly4

