

Monthly Logbook

Issue
1102010100011221



Индустрия давно нормализовала
технический абсурд



— Кто вообще решил, что это хорошая идея?

Некоторые алерты переживают эпохи.

Тишина.

Где-то на четвёртом уровне вложенности bash собирал JSON для какого-то генератора, дальше запускался PowerShell, потом появлялся XML для чего-то важного.

Да, это прод.

Да, международной компании.

Да, от количества нулей иногда кружится голова.

И да — оно работало.

Работает до сих пор.

Кто сделал — уже забыли.

Как работает — отдельный разговор.

Некоторые вещи в ИТ переживают эпохи.

Когда-то два системника уже считались стойкой. Ночью спорили в IRC с Томском, Новосибирском, Москвой и кем-то из Далласа, чинили принтеры без мануалов, тянули локалки и обсуждали проблемы с людьми за тысячи километров.

Тогда многие инженерные задачи решались через простое:

| «разберёмся»

Сейчас вместо IRC можно спорить с ИИ до трёх ночи.

Суть всё та же.

Поднять.

Починить.

Понять, кто вообще решил, что это хорошая идея.

А потом — научить других чинить это самостоятельно.

Где-то между этим мы почему-то называем всё происходящее магией.

Инженеры — костылями.

Доклады были.

Статьи были.

Где-то вдохновлял. Где-то учил. Говорят, мною даже детей в сибирских деревнях пугали.

Теперь будет и канал.

Иногда ночью всё ещё кажется, что где-то прозвучало:

| «О-оу!»



— Подождите.

— То есть прод сейчас живёт потому, что cron каждые пять минут проверяет контейнеры и поднимает упавшие?

— Да.

Тишина.

— И давно так?

— Полгода.

42–50 тысяч RPM на балансере.

404–500 тысяч уникальных пользователей в сутки.

Да, это прод.

Да, большой компании.

Да, решение странное.

Работало?

Работало.

Иногда инженерия — это выбор между:

| «сделать правильно»

и

| «сделать так, чтобы бизнес пережил понедельник»

Потом проходит полгода.

Кто-то осторожно спрашивает:

| «а это временное решение?»

А потом рассказываю про это в кулуарах конференции.

И начинается самое интересное.

Обычно уточняют:

| «Подождите. Вы точно про продукт В?»

Потом — длинная пауза.

И где-то в этот момент обычно всплывает:

| «Подождите... то есть это была одна VM, docker-compose
и один балансер?»

К этому моменту кофе обычно уже нужен всем.

Тишина.

А cron — жалко.





Иногда я ставлю встречи на странное время.

Например:

| 09:14:37

Иногда даже живые.

Иногда люди приходят в 09:15.

Иногда — в 09:14.

Ровно в:

| 09:14:37

обычно приходят HR.

Из аутстафферов.





Когда-то моё фото из 2007 года стало мемом.

Часть людей решила:

| «такого не бывает»

Часть — что фотошоп.

Часть — что байка из интернета.

А там всё было честно.

ГорГаз.

Серьёзное предприятие.

Труба под потолком.

Примерно три метра высоты.

Сетевое оборудование и UPS, закреплённые скотчем.

Сеть была.

Работало.

Иногда кажется, что инженерские байки и мемы
происходят где-то в другом мире.

А потом понимаешь:

| иногда — труба.

Команды меняются.

Люди меняются.

Требования меняются.

Мониторинг остаётся.



С годами начинаешь замечать странные закономерности.

Почему-то у многих людей, которые долго в ИТ и уже многое сделали, однажды появляется мотоцикл.

У кого-то Harley.

У кого-то старый японец.

У кого-то внезапно — «Урал».

Когда многое уже построено.

Многое уже ломалось.

Многое собиралось заново.

И почему-то именно в этот момент у некоторых появляется байк.

Чтобы поворот — это поворот.

Дорога — дорога.

Хотя, возможно, дело совсем в другом.

Если выезжаете — берегите себя.

Некоторые инженеры нужны индустрии живыми.

Сегодня в плейлисте — звук старого Harley.

Или Indian.



Только не дисконект.

IRC.

Форумы по ночам, шум обсуждений.

Звук дозвона модема, короткие сигналы.

Местный интернет-провайдер, кабельные сети, транки.

Форумы по ночам — снова, искать решение.

Зима.

Гора.

Проверить передатчик.

Грозы.

Грозы тогда были отдельной формой жизни.

Что-то не работает.

Потом много чего.

Интернет-площадки, ночами чинить форумы,
мониторинг, бэки, веб-приложения.

Позже федеральные провайдеры, сети, кабели, ночь.

После грозы снова что-то поднять.

Интернет как-то оказался рядом всю жизнь.

CDN, Highload.

И принесли пицу.

| Кто-то поднялся в гору.



Зимой на KuBeer Talks я выступал с докладом:

| «Почему вам не нужен Kubernetes»

После выступления несколько человек подошли примерно с одинаковым вопросом:

| «Ты серьёзно считаешь, что Kubernetes не нужен?»

Kubernetes за последние годы стал для индустрии ответом по умолчанию. Проект только стартует, требования ещё меняются, команда небольшая. В архитектуре уже появляются ingress, service mesh, Helm, GitOps, мониторинг, централизованное логирование — современная инженерная молитва.

Вокруг проектов уровня:

| «обычный интернет-портал»

строится инфраструктура так, будто завтра сервис должен пережить многократный рост трафика, несколько серьёзных инцидентов и внезапную пятницу.

Условный «привет, мир» с формой заявки и нагрузкой:

| «два человека в праздники»

Зачем?

Ответ обычно простой:

| «так сейчас делают»

Цена обычно приходит позже — когда кластер начинает жить собственной жизнью, обновление неожиданно ломает работающую систему, а команда понимает:

| Kubernetes — это уже отдельный пласт постоянной работы.

Обычно это выясняется во время инцидента. Или в момент, когда оказывается, что Kubernetes в компании по-настоящему понимают полтора человека.

Доклад в целом был про простую вещь:

| технология должна соответствовать задаче и команде.

Kubernetes отлично работает там, где много сервисов, высокая нагрузка, сложная инфраструктура и большая команда.

Иногда это напоминает попытку завести 300-килограммовую гориллу в маленькую квартиру.

Завести получилось.

Теперь её нужно кормить, сопровождать, объяснять бизнесу, почему вместо новых возможностей команда уже третий месяц разбирается с инфраструктурой.

Особенно интересно это выглядит в небольших компаниях: несколько разработчиков, планы, диаграммы Ганта, встречи про скорость разработки.

Всё это в итоге держит один парень в футболке из Омска, которому теперь ещё и Kubernetes сопровождать.

За 20+ лет эксплуатации я увидел простую закономерность:

| проблема обычно возникает в момент, когда амбиции, реальные задачи, компетенции команды и стоимость сопровождения перестают совпадать.

Иногда достаточно Docker Compose, Nomad, PaaS, managed Kubernetes или обычной VM.

В итоге всё обычно сводится к довольно скучным вопросам:

| работает ли система
| выдерживает ли нагрузку
| умеет ли команда её поддерживать
| придётся ли кому-то чинить всё это ночью

Ночами, если честно, лучше спать.

Для тех, кто любит спорить аргументированно — приложил презентацию с KuBeer Talks ↗.

Если хотите, чтобы система плохо
переживала высокую нагрузку, есть
один проверенный способ.

Открывайте connection к базе.

И не спешите.

Сначала собирайте данные.

Потом ещё данные.

Что-нибудь посчитайте.

Куда-нибудь отправьте.

И только потом закрывайте
connection.

И так каждый request.

Особенно интересно потом слышать:

| «что-то highload не получается»

Тишина.

| Любовь — это принимать любые
недостатки.







Были революционные профессии.

Эксперт по вводу информации в компьютер.

FoxPro.

Человек-сканер.

Распознавание текста в реальном времени.

Десятки записей за день.

Ручная работа.

Клавиатура стучит.

Свет.

Музыка.

Ночной инцидент.

Вместо ночного кофе — отвлекаться
на работу.

После устранения аварий понимаешь:

Надёжность — не для бизнеса.

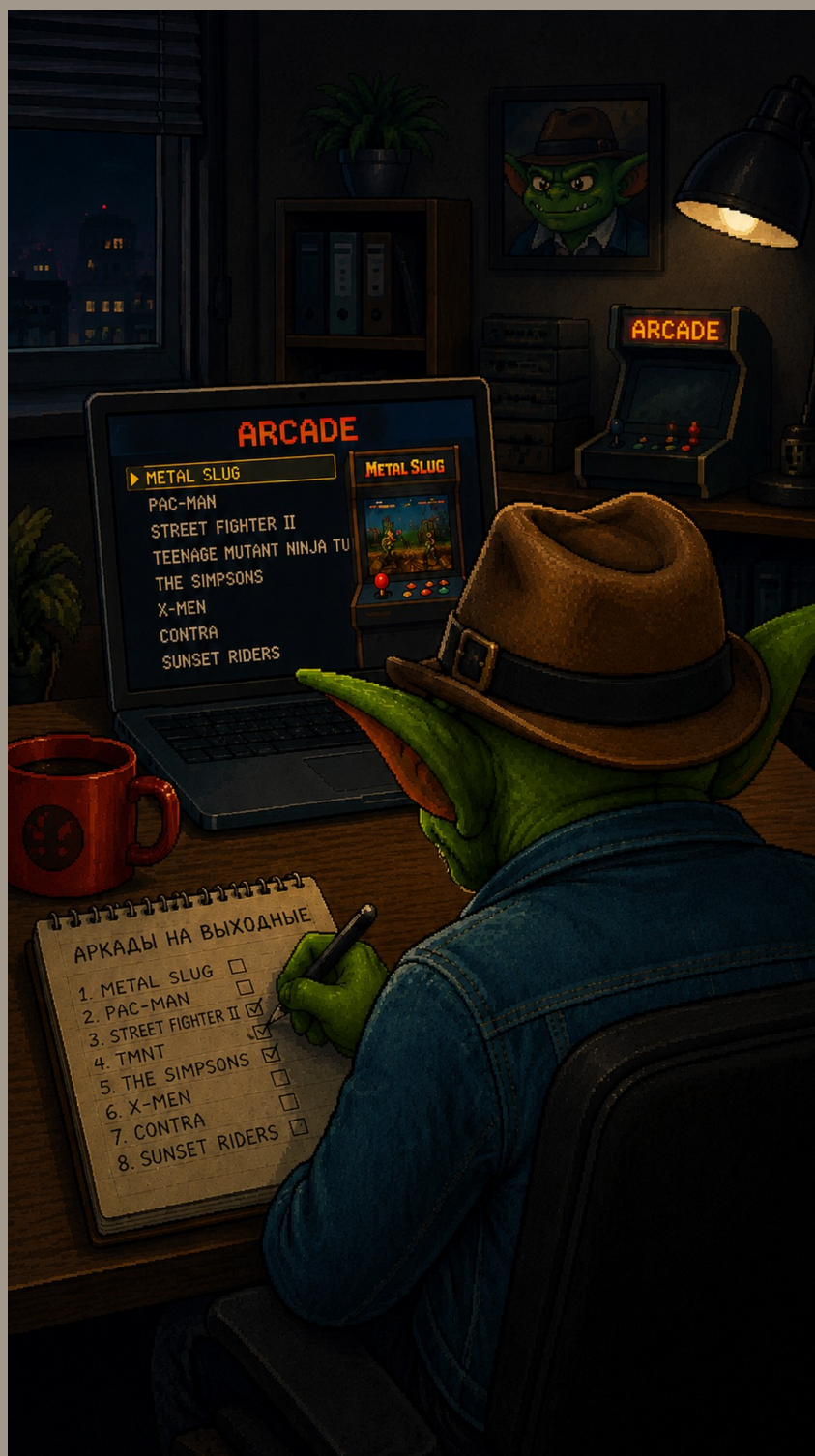
Не для пользователей.

Ночью лучше спать.

И лучше не одному.

«Moscow Calling»







Люблю аркадные игры 70–80-х.
Как инженерную работу.
Сделать нечто интересное в очень ограниченных условиях
— отдельное ремесло.
Памяти мало.
Процессор слабый.
Несколько цветов на экране.
И при этом — лучшая графика своего времени.
Люди стояли в очередях к автоматам.
Никаких патчей.
Никакого «доделаем потом».
Нужно сразу сделать механику, которая работает.
И работает так, что спустя 30–40 лет в это всё ещё играют.
Сейчас местами даже сложно понять — как они это
вообще сделали.
И играть в это до сих пор сложно.
Аркады плохо умели жалеть игрока.
Потому что хорошо сделано.
Честная механика.
Честная сложность.
И ведь местами это были просто игры.
А потом вокруг них выросло новое железо.
Новые возможности.
И новые ограничения.
Забавно.
Когда-то это железо двигало спрайты в аркадных
кабинетах.
Спустя годы — уже считало другие монетки.
Eject Coin



Почему все так боятся пятницы?

Тогда пятница снова становится просто днём недели.

За годы эксплуатации заметил странную вещь.

Бояться обычно стоит чего-то другого.

В ИТ пятница давно стала религией.

| «В пятницу релизы не делаем»

| «После обеда ничего не трогаем»

| «Кто катит в пятницу — сам потом и дежурит»

Логика понятна.

Что-то сломается, кто-то останется ночью, а кто-то будет
искать того самого инженера, который «точно знает как
это работает».

Иногда — уже в субботу.

Но есть один неудобный вопрос.

Почему пятница?

Система знает, какой сегодня день недели?

Если релиз в пятницу опаснее релиза во вторник —
проблема не в календаре.

Обычно всё скучнее.

Rollback есть.

Но спокойствия это не добавляет.

Мониторинг зелёный, пока клиент уже пишет, что всё
плохо.

Тестирование было.

Но уверенности в изменениях мало.

Радиус проблем становится понятен уже после релиза.

Тогда да.

В пятницу лучше ничего не трогать.

Потому что иногда последствия становятся заметны
только в понедельник.

Вместе с потерянными клиентами.

И такое бывало.

А иногда всё работает иначе.

Изменение выкатывается спокойно.

Ошибка остаётся локальной.

Инженер спокойно уходит отдыхать.

И релиз перестаёт зависеть от того, ответит ли в субботу
тот самый парень в футболке.





Гонка, где скорость — только часть задачи.

Доехать быстро — мало.

Утро встречают другие.

У финишной черты.



Как-то подходит архитектор:

— Тут есть VM, которые давно никто не использует. Давай удалим.

Например, dev-стенд команды, которая ушла пару лет назад.

11:57 — выключили.

12:00 — обед.

12:13 — звонок:

— У нас час пик. Всё встало. Очереди.

12:20 — включаем.

12:21 — сервис восстановлен.

Root cause: Часть интеграций обращалась к dev среде.

Impact:

Решилась одна загадка — почему багфиксы ничего не меняли.

На тестах — всё хорошо. И на «прод стенде» — тоже.

Современное АйТи решает задачи,
которые вчера казались
невозможными.

Мир растёт, нагрузки растут.
Всё, что вчера казалось неподъёмным,
сегодня — рутинная операция.

QR-код в SSH-терминале.

На другом континенте.

Для авторизации пользователя.

Работает.

К этому уже привыкли.





Милые существа вокруг.

Пока:

не намочишь;

не покормишь после полуночи;

не нарушишь правила.

— Да там маленькое изменение.

В пятницу.

После полуночи.





31 мая 2021 года я впервые выступал на большой конференции.

К тому моменту сами конференции уже давно были частью жизни.

Программный комитет.

Доклады.

Помощь спикерам.

Обсуждения.

Споры про содержание.

Та сторона процесса, где обычно пытаешься сделать так, чтобы после доклада у человека осталось полезное.

DevOpsConf тогда была большой историей.

Во многом — главной конференцией по теме.

И долгое время всё происходило с другой стороны сцены.

Так даже было привычнее.

Есть люди, которые выступают.

Есть люди, которые помогают сделать так, чтобы всё состоялось.

Однажды оказываешься уже с микрофоном.

Перед залом.

И рассказываешь:

| «Зачем DevOps в шахте?»

Само название тогда звучало немного странно.

Для многих DevOps всё ещё был про интернет-компаний.

CI/CD.

Облака.

Быстрые релизы.

Красивые схемы.

А тут — металлургия.

Шахта.

Производство.

Процессы, часть из которых местами не менялась десятилетиями.

И разговор о довольно простой вещи.

Иногда некоторые вещи уже существуют.

Просто никто ещё так это не называет.

Тогда сама идея многим звучала странно.

| DevOps в шахте.

Время тогда вообще было интересное.

Все вокруг обсуждали майнинг.

Если честно, сам я в эту историю тогда верил сильно меньше, чем люди рядом.

Иногда рядом оказываются те, кто почему-то раньше тебя понимает, что пора выходить из-за кулис.

Без длинных разговоров.

Без мотивационных речей.

Просто однажды звучит:

| «Иди уже. Нормально будет.»

И почему-то идёшь.

Тогда это казалось обычным разговором.

Одной из многих историй вокруг сообщества.

Только спустя годы начинаешь понимать, что некоторые люди иногда дают тебе сильно больше, чем кажется в тот момент.

Части людей, которые были рядом тогда, уже нет.

А некоторые разговоры почему-то помнятся очень подробно.

Потом были другие доклады.

Другие сцены.

Модерация залов.

Но тот первый момент всё равно помнится отдельно.

Крупнейшая DevOps-конференция.

Полный зал.

И доклад про production.

Сегодня в памяти — звук конференционного микрофона.

И немного шум цеха.

Доклад ↗

ШКОЛА ИМПЕРИИ

РЕГЛАМЕНТ РАСПРОСТРАНЕНИЯ КОНТЕНТА



TELEGRAM

Инженер школы Империи



VK

Инженер школы Империи



YANDEX DISK

Yandex Disk Archive



GOOGLE DRIVE

Google Drive Archive



CONTENT SYNC

Monthly Logbook Content Sync



ЛИЦЕНЗИЯ

CC BY-NC-ND 4.0



Monthly Logbook

Мастерская Империи по заказу Школы Империи

© Дмитрий Малыхин, 2026

