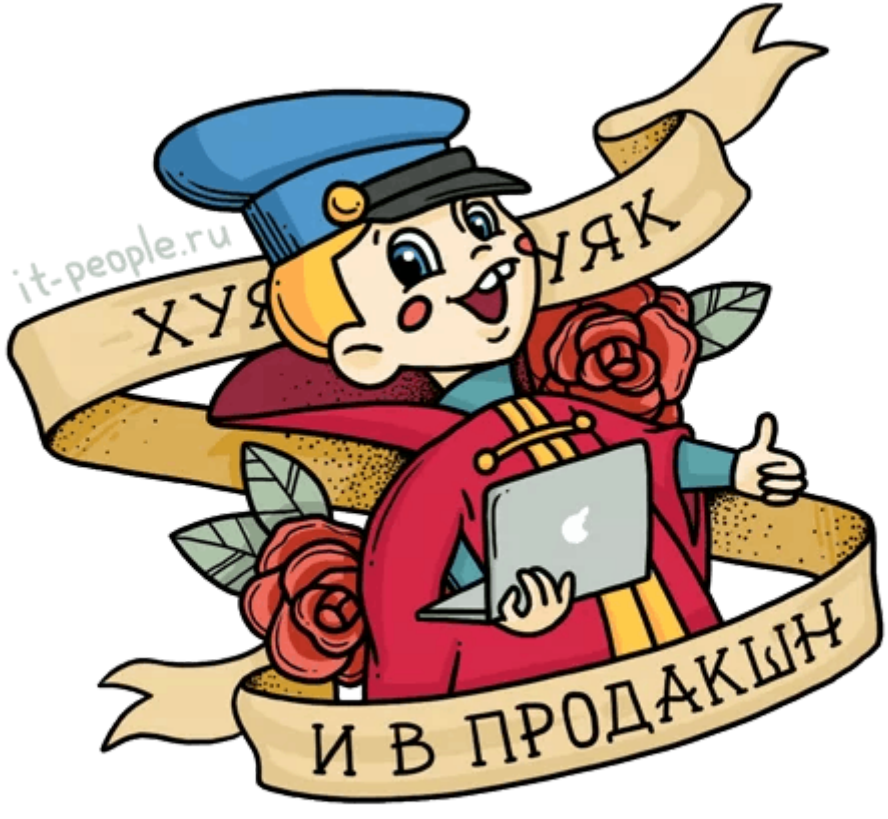
# Достаточно хороший конвейер поставки

# Не забудь включить запись!

# Что такое достаточно хороший?



# Что такое достаточно хороший?

## Удовлетворяет текущие потребности команд

- Быстрая выкатка и тестирование изменений
- Быстрая и сфокусированная обратная связь
- Крепкий и здоровый сон

# Что такое достаточно хороший?

## Не дает слишком много свободы

- Есть шаблоны (проектов, манифестов, документации и процессов)
- Есть стандарты (линтеры, опять же, документация и процессы)
- Есть контроль (тесты, RBAC, security check-up)

> "- Так это же бардак! - Бардак. Зато ты - главный!" (с)

# Что такое достаточно хороший?

## Есть куда развиваться дальше

- Требования и технологии меняются
- Всегда можно сделать лучше (надежнее, быстрее, чаще)
- Развитие платформы тесно связано со структурой команд и процессом разработки

Поэтому нам нужен понятный **план развития**.

# Разработка

- Разработчикам должно быть удобно.
  - Удобнее всего им разрабатывать локально.
- Рано или поздно, это начинает тормозить:
  - Монолит распух
  - Микросервисов стало слишком много
- С 12FA неудобно жить локально:
  - Доступ к внешним сервисам (разные registry)
  - Сервисный слой
  - JSON-лог в `stdout` - это неприятно
- Вообще не похоже на production

# Разработка || Куда идти?

## Локальные сборка и запуск в Kind/Minikube

- Лучше, чем просто локально
- Но медленнее (сборка и ВМ)
- Для динамических языков - нет live reload
- Сложнее отлаживать процессы в контейнерах

# Разработка || Куда идти?

## Сборка и запуск в Dev-кластере

- Примерно те же проблемы, что и при локальной разработке
- CI-сервер должен собирать и выкатывать код в кластер быстрее, чем ноутбук разработчика
- Если можем выкатывать только "по коммиту":
  - Много мусора в истории коммитов (нужны конвенции по commit messages, auto-squash не значащих коммитов)
  - Или коммитить мелкими и частыми изменениями (TBD)

# Разработка || Куда идти?

- Skaffold
  - Save -> Build -> Deploy
  - Не забывайте чистить Registry
- Упомянутые в лекциях Ksync и Telepresence
  - Только не используйте их в Production
- WebIDE
  - RedHat CodeReady
  - Theia и прочие

# Dev-кластер

## Namespace-per-project

- Общий namespace для проекта
- Общие ресурсы, непредсказуемые версии компонентов
- Общая БД и проблемы с миграциями

> Спонтанные обновления сервисов могут испортить тесты, а миграции БД - сломать все. DB per service - OK

# Dev-кластер

## Namespace-per-component

- Нужный нам сервис и зависимости
- Mocks, stubs
- Легко запустить интеграционные тесты и тесты нагрузки
- Своя БД

> "Выкатка без коммита" возможна, если только один разработчик меняет код сервиса

# Dev-кластер

## Namespace-per-branch

- Параллельная разработка нескольких фич или багфиксов
- Легко сопоставляется с Git flow

# Personal Dev-cluster

- Максимальная свобода действий для команд
- Можно воссоздавать namespace схемы с Production
- Нет драк за ресурсы
- Managed Kubernetes или Gardener

# К чему идем?

- Независимость сборки и поставки от локального окружения
- Общий набор инструментов для диагностики и мониторинга
- Условия, максимально близкие к Production
- "Окружение" по кнопке

# Тестирование

## Стандартная пирамида тестирования

- Параллельное выполнение тестов
- Требует "интеграционных стендов", которые близки к production (в том числе и по ресурсам)
- Можно тестировать вручную, но лучше этого не делать
- В составе набора тестов должны быть тесты безопасности

# Тестирование

Тестирование должно захватывать не только приложение.

Все три слоя инфраструктуры (Base-Service-App) должны быть протестированы

> Например, так

# Progressive Delivery

- После минимальных проверок код попадает в продуктивную среду, но недоступен для пользователей
- Нужно продумать управление данными и маркеры "тестовых" запросов
- В коде должны быть feature toggles
- Платформа должна уметь управлять трафиком пользователей (зеркалирование, перенаправление)
- Нужны инструменты для управления такими "сложными" развертываниями
- и мониторинг тоже нужен

Обзор практики PD

# Полезные ссылки

- Тех. радар от ThoughtWorks
- Ландшафт CNCF
- Блог Cindy Sridharan


# Спасибо за внимание!

**Время для ваших вопросов!**