

Практики DevOps

Практики

- Инфраструктура как код
- Непрерывная поставка ПО (Continuous Delivery)
- Сбор обратной связи

Практики



Инфраструктура как сервис



Инфраструктура как сервис

- Облачные провайдеры и их API
- Приватные облака и их API
- Остальное можно сделать с помощью Configuration Management Systems (Chef, Puppet, SaltStack, Ansible)

Инфраструктура как код



Инфраструктура как код

- Подход, при котором инфраструктура описана в виде кода и хранится в VCS
- Применимы практики разработчиков:
 - ревью
 - тестирование
- Контроль: предсказуемость и повторяемость

Инфраструктура как код. Примеры

Создание виртуальной машины в AWS с помощью Ansible:

```
1  tasks:
2    - name: create ec2 instances
3      ec2:
4        key_name: brun
5        instance_type: t2.micro
6        image: ami-9a380b87
7        wait: yes
8        region: eu-central-1
9        group: webservers
10       count: 1
```

Важно, что мы описываем желаемый результат, а не конкретные операции

Инфраструктура как код. Примеры

Установка Ruby и зависимостей с помощью Chef:

```
1  ruby_install ruby_version do
2    action :install
3  end
4
5  ruby_set ruby_version do
6    action :set
7    username user
8  end
9
10 %w(libmagickwand-dev libmagickcore-dev libpq-dev imagemagick).each do |pack|
11   package pack
12 end
```

Инфраструктура как код. Примеры

Создание сервиса в Kubernetes:

```
1  kind: Service
2  apiVersion: v1
3  metadata:
4    name: my-service
5  spec:
6    selector:
7      app: MyApp
8    ports:
9      - protocol: TCP
10     port: 80
11     targetPort: 9376
```

Под “капотом” - настройки iptables, записей в DNS, отслеживание экземпляров приложений и еще куча всего

Платформа



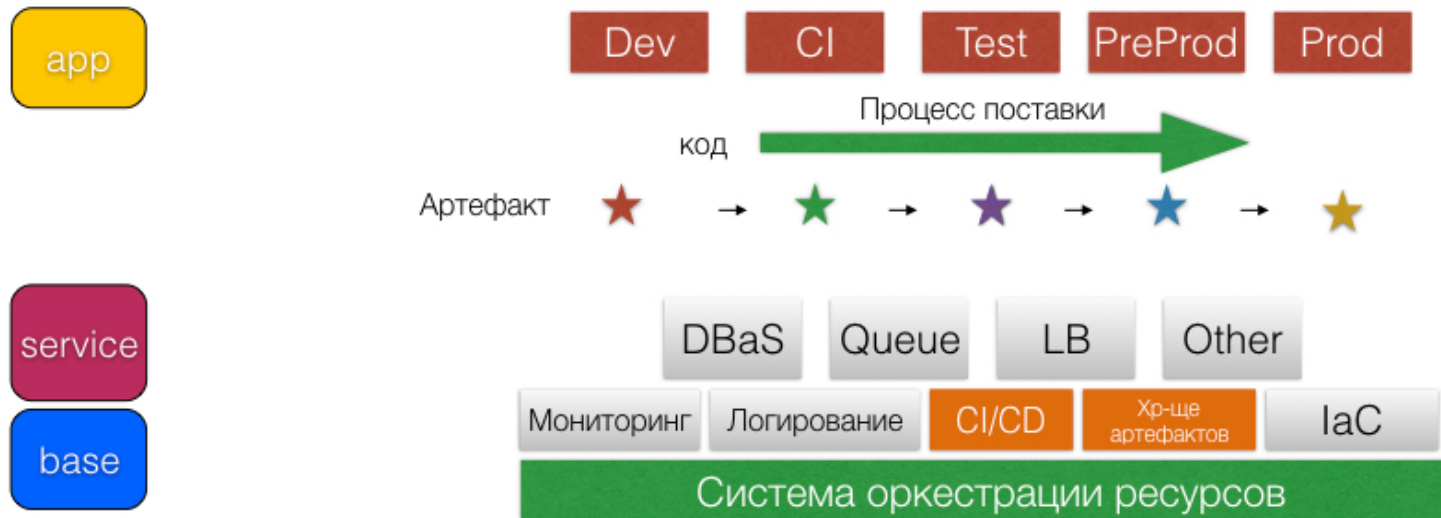
Инфраструктурная платформа

- Инфраструктурная платформа - это такой же продукт. Только внутри компании.
- Задачи платформы - уменьшить время “от коммита до production” и упростить управление окружениями
- Отдельная команда
 - Сбор требований, Backlog, Roadmap
- Фокус на продуктовые команды
- API



Pivotal **Web Services**

Схема платформы



Непрерывная поставка



Continuous Integration

Это практика разработки, в которой все участники команды часто интегрируют свою работу в общую кодовую базу. Каждая интеграция автоматически проверяется на наличие ошибок.

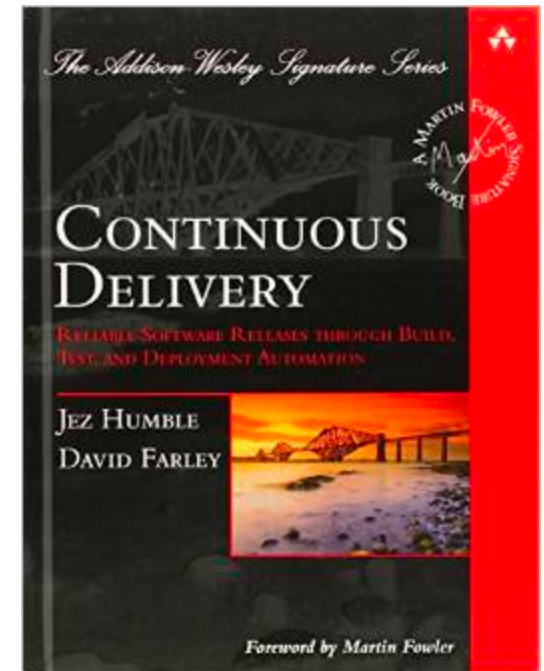
Критерии наличия CI

- Все ежедневно мержат свой код в мастер
- На каждый коммит запускаются юнит-тесты
- Быстрая починка сломаного билда, например, за 10 минут

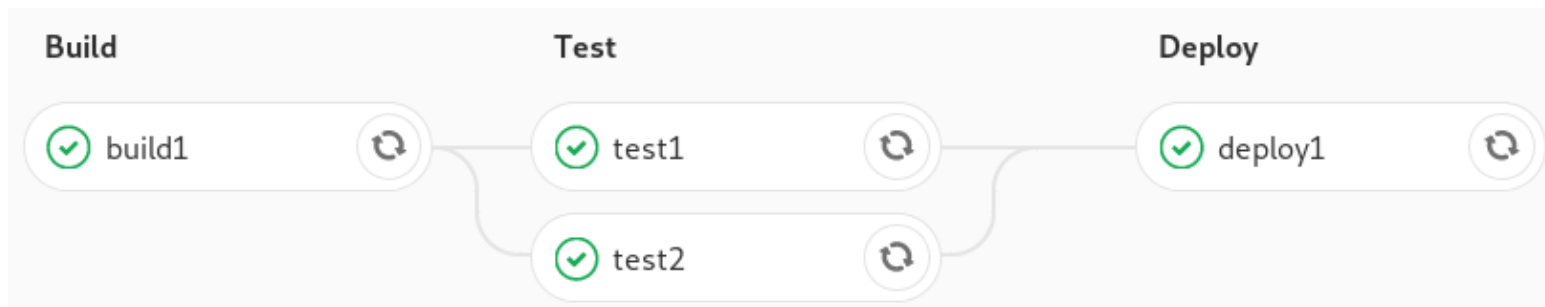
Continuous Delivery

Непрерывная поставка ПО

это процесс разработки и эксплуатации ПО, при котором **каждое изменение** может быть выкачено **в боевую среду**



Конвейер



Принципы

- Создать повторяемый и надежный процесс поставки ПО
- Полная автоматизация
- Хранить все в системе контроля версий
- Совершенствоваться через повторения
- Получать раннюю обратную связь

Контроль качества

- Линтеры
- Статические анализаторы
- Юнит-тесты
- Интеграционные тесты

Политики деплоя

- Blue-green
- Canary
- Rolling

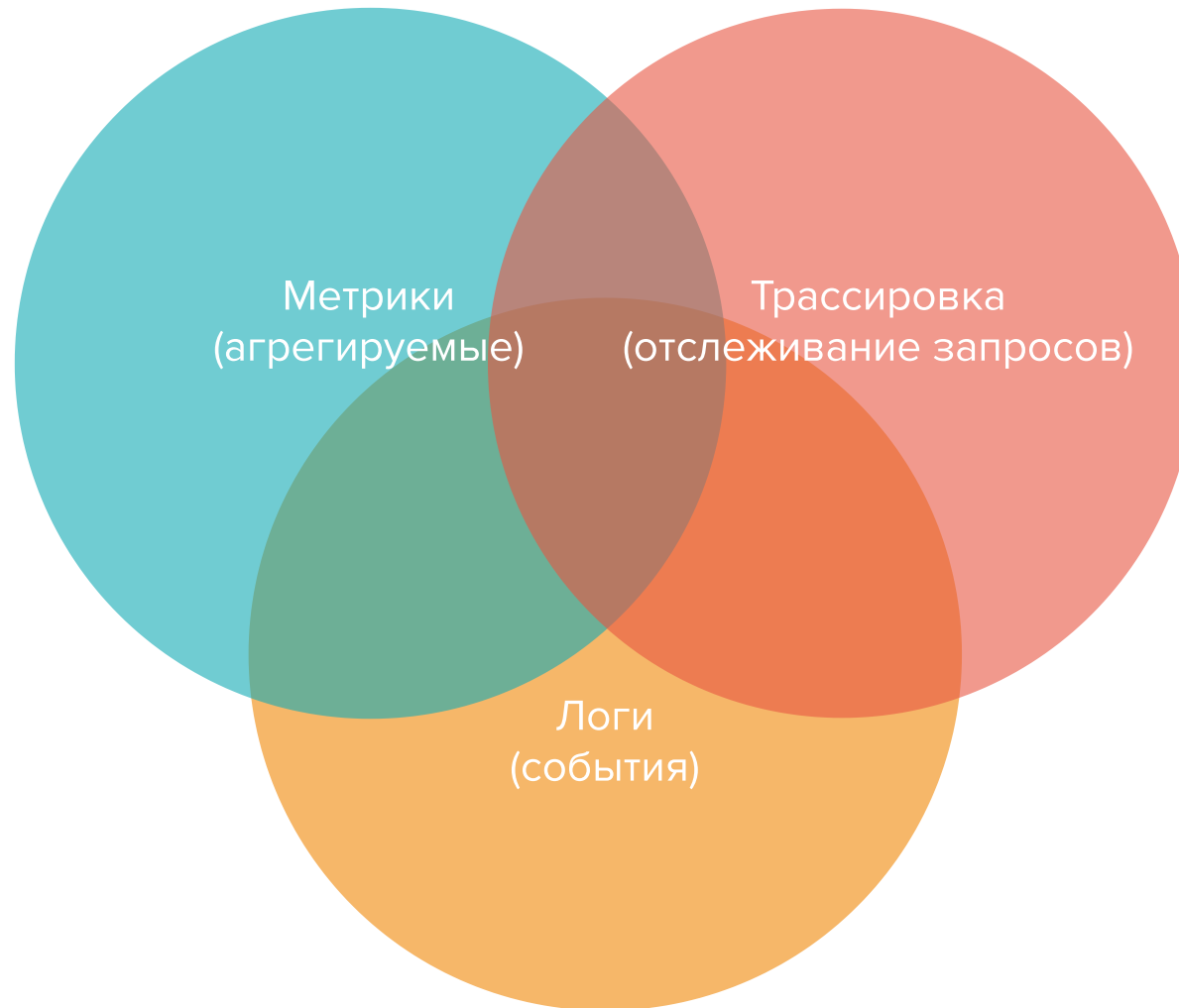
Инструменты для CI/CD

- Jenkins
- Jenkins X
- Concourse CI
- GitLab CI

Обратная связь



Три столпа обратной связи



Логирование

- Все приложения логируются автоматически
- Инструмент для всех
- Анализируем бизнес-события, проверяем гипотезы
- Логирование пишут разработчики
- Система логирования предоставлена как сервис

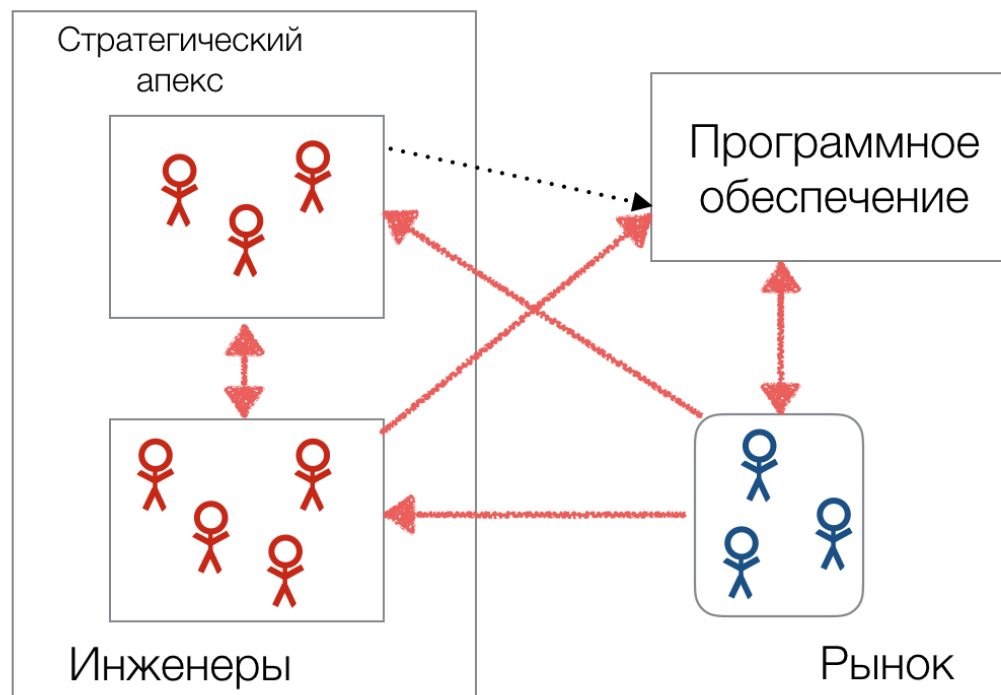
Мониторинг

- Собираем метрики
- Автоматически мониторим приложения на всех окружениях
- Мониторинг пишет разработчик
- Мониторинг предоставляется как сервис

Тре́йсинг

- Пишем все запросы
- Видим общение наших сервисов
- Анализируем долгие запросы
- Тре́йсинг пишет разработчик
- Тре́йсинг предоставляется как сервис

Производство цифровых продуктов



Информация ←

Модель DevOps



DevOps компетенции

- Разработчик с представлением об архитектуре и работе софта в продакшене (пишет тесты и инфраструктурный код)
- Инфраструктурный инженер (пишет обвязки для инфраструктуры, предоставляет разработчику платформу)
- Разработчик инфраструктурных сервисов (DBaaS, Monitoring as Service, Logging as Service)
- Релиз-менеджер (управляет процессом и зависимостями)
- SRE

Что почитать

- Книги Lean Startup и Lean Enterprise
- Книга Continuous Delivery
- Книга Practical Monitoring <https://www.practicalmonitoring.com>
- Книга Refactoring Databases: Evolutionary Database Design
- DevOps for DBA <http://devopsfordba.com/>
- Статьи Мартина Фаулера <https://martinfowler.com/design.html>
- Team Topologies <https://teampologies.com/>