
DVPS
DEVELOPMENT AND OPERATIONS

Технология непрерывной поставки ПО

Практики DevOps

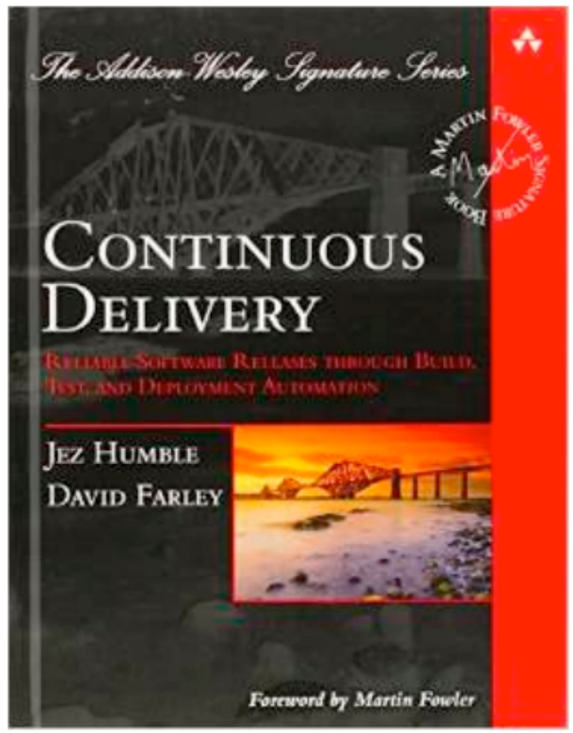
- Непрерывная интеграция (Continuous Integration)
- Непрерывная поставка (Continuous Delivery)
- Тестирование и Управление тестовыми данными
- Непрерывный мониторинг
- Управление версиями и релиз -менеджмент
- Управление конфигурацией и инфраструктура как код

Непрерывная интеграция

- проект забирается из СКВ
- проект собирается
- прогоняются тесты
- проходит выкатка на тестовый стенд (шоты)
- рассылаются оповещения

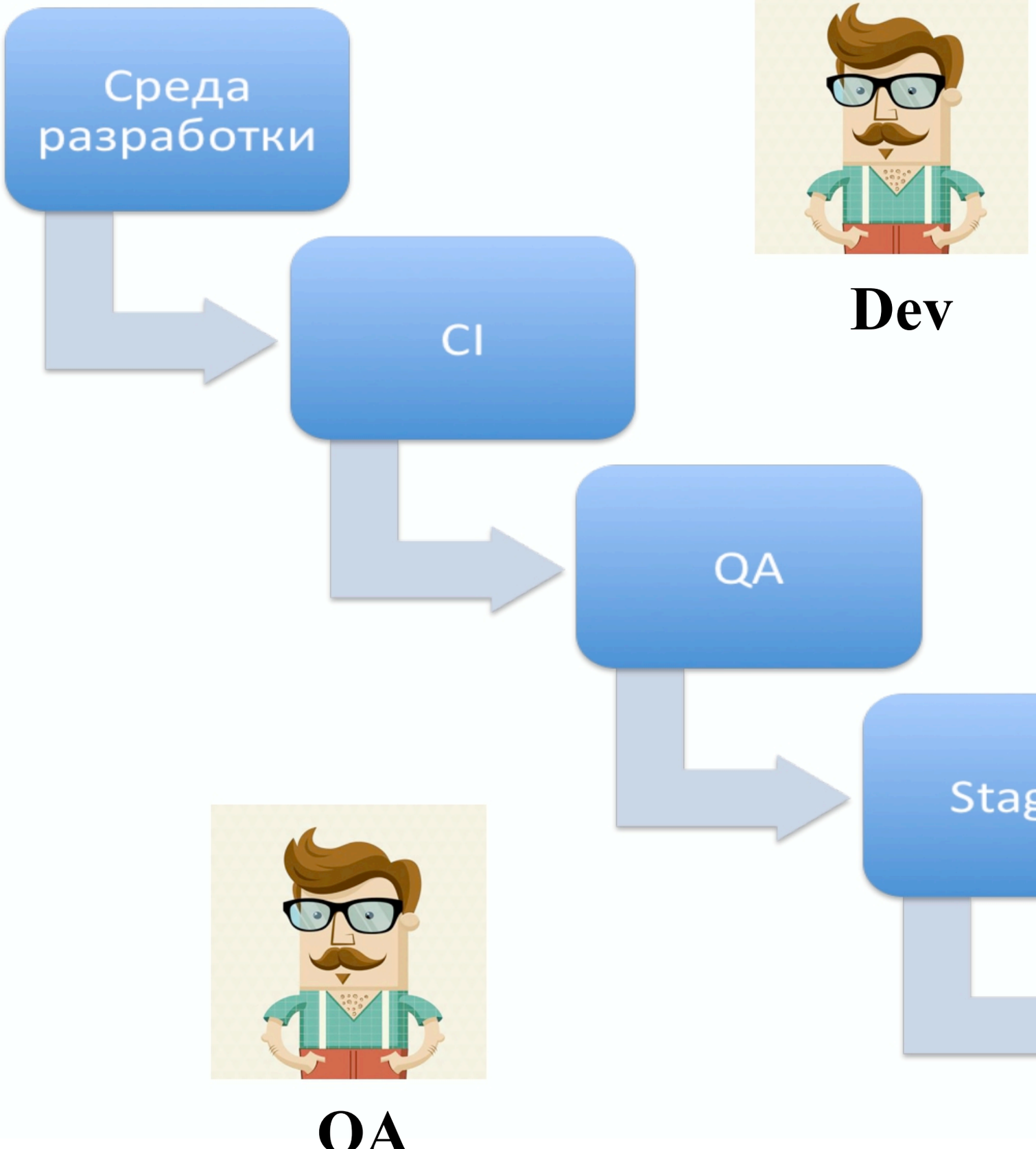
Непрерывная поставка

- Continuous Delivery (CD)



- Непрерывная поставка ПО — это процесс разработки и эксплуатации ПО, при котором каждое изменение может быть выкачено в боевую среду

Конвейер



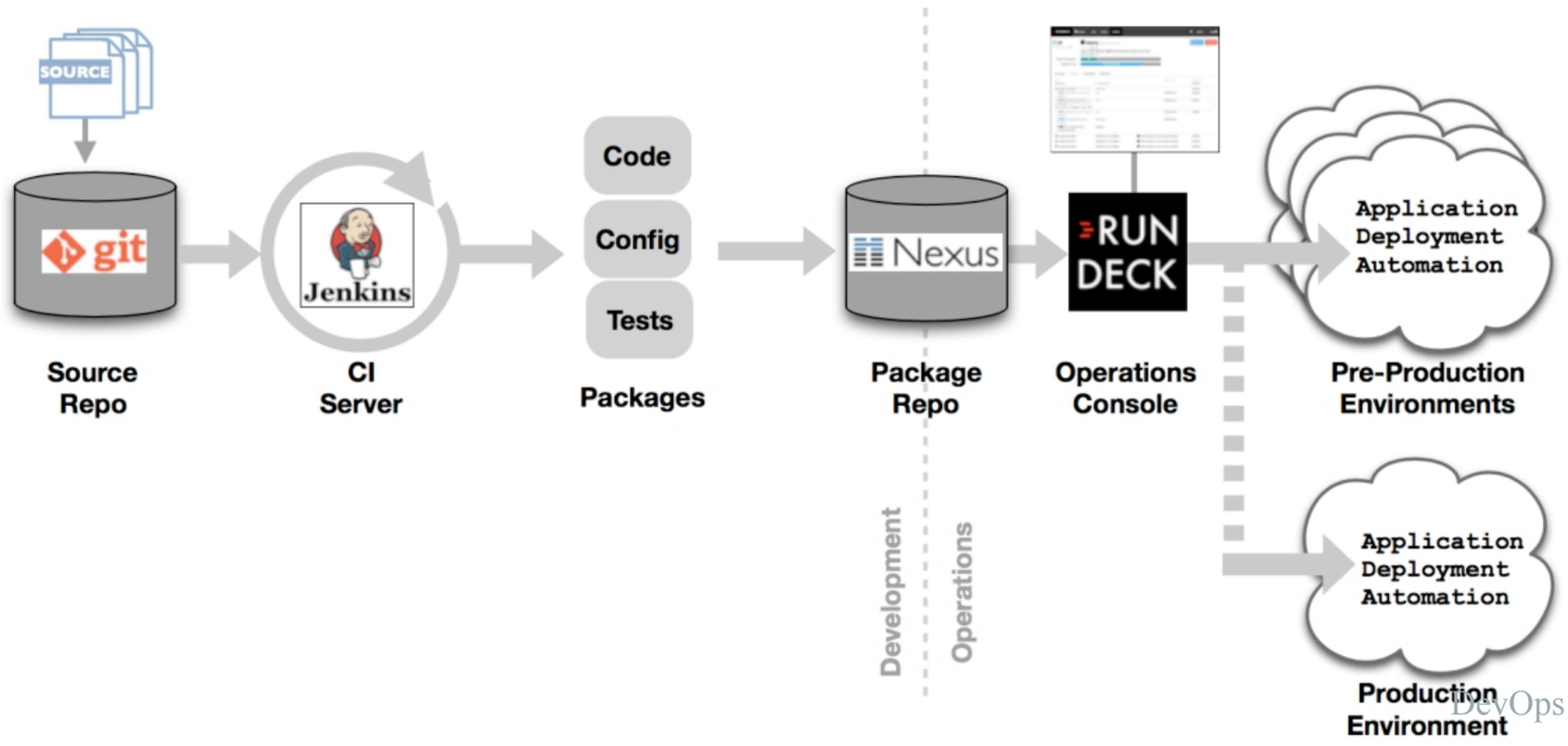
Dev

Ops

QA

CI/CD

- Автоматизация процессов (сборки, деплой, инфраструктура)
- Стандартизация подходов к управлению окружениями
- Повышение прозрачности процессов
- Обеспечение командного взаимодействия



Проблематика

· Разработка и Тестирование нуждаются в независимых окружениях.
· Окружений становиться много, непонятно кто за них отвечает
· Состояние окружений не синхронизировано

dev01.example.com

dev02.example.com

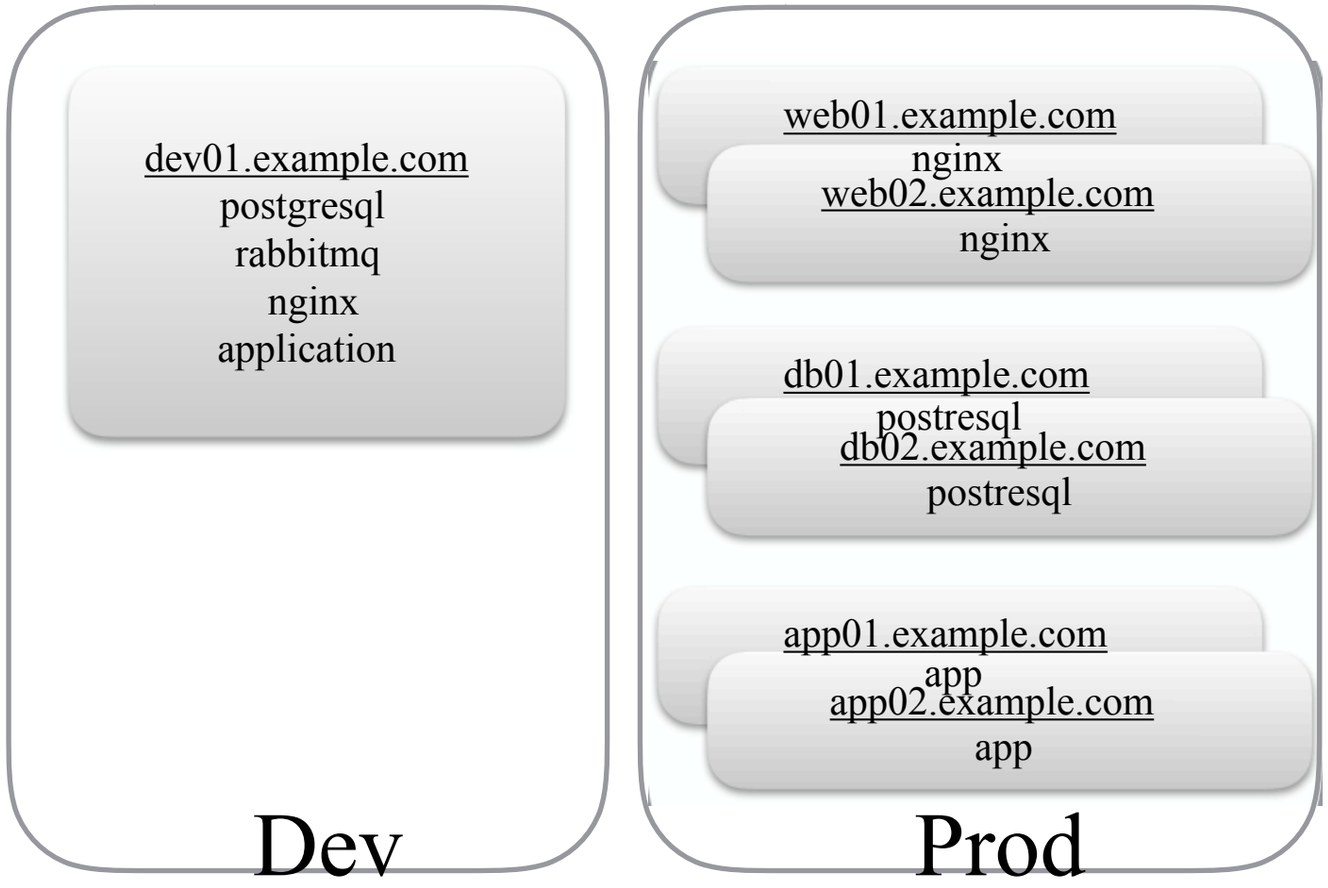
test01.example.com

test02.example.com

На test01 развернута новая фича с изменениями в структуре таблиц БД и мы не можем там тестировать багфикс .
На dev02 кто - то обновил openssl, а на остальных – забыл .

Проблематика

· Продукт может состоять из множества компонент
· Много компонент * много окружений = сложно
· Ограничения в ресурсах



Мы планируем запустить рекламную компанию и нужно провести нагрузочное тестирование на окружении аналогичном боевому

Процессы CI/CD

- Сборка приложения
- Контроль качества
- Публикация протестированных артефактов сборки в хранилище

- Создание нового окружения / Удаление более не нужного окружения
- Обновление окружения до состояния, описанного в виде кода
- Выкатка новой версии приложения / Откат до прошлой версии
- Применение изменений к базам данных этого окружения

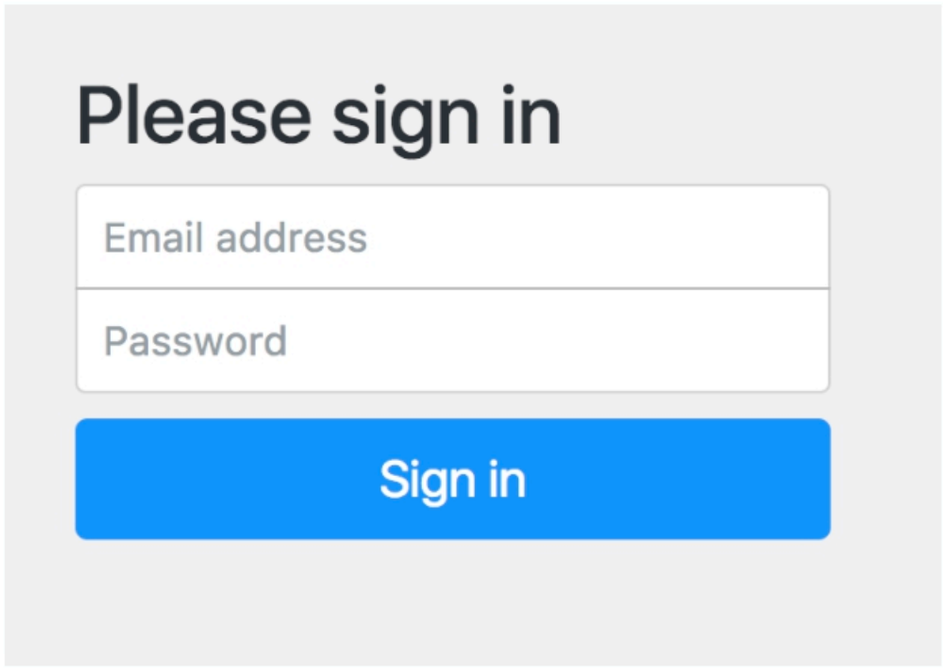
Рассмотрим на примере

Веб приложение
Фронтенд : Angular2
Бекенд : Flask
База : PostgreSQL

web01.example.com
nginx, static files (html, js)

app01.example.com
python (api, flask app)

db01.example.com
postgresql



Пример 1: новый функционал приложения

Задача : добавить кнопку “Запомнить меня”

Фронтенд

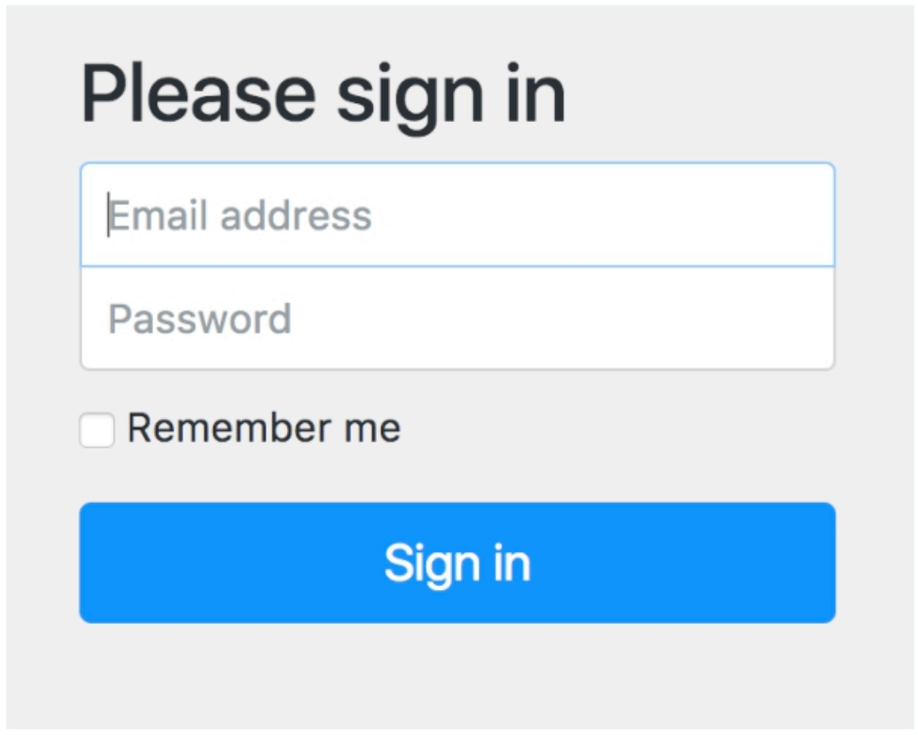
```
<form name="form" (ngSubmit)="f.form.valid && login()" #f="ngForm" novalidate>
  …
      <div *ngIf="f.submitted && !password.valid" class="help-block">Password is required</div>
      </div>
+     <div class="checkbox">
+       <label>
+         <input type="checkbox" value="{{remember_me}}"> Remember me
+
```

Бекенд

```
+ @app. route( "/session", methods=['POST'])
+ def session():
```

БД

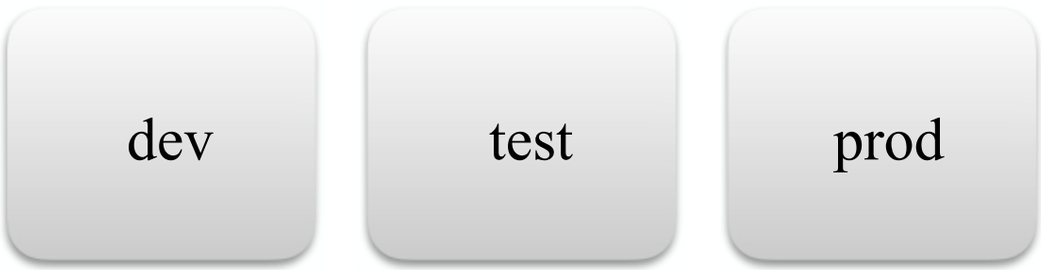
```
+ CREATE TABLE session (
+   ...
+);
```



Пример 2: новая инфраструктура

Задача: добавление функционала рассылок

```
- hosts: email-servers
  roles:
     - role: postfix
       postfix_config_templates:
         -
           src: 'main.cf.j2'
           dest: '/etc/postfix/main.cf'
           owner: root
           group: root
           mode: '0644'
     postfix_tld: 'example.com'
```



Пример 3: Security disaster

Задача: Случился heartbleed. обновить openssl на всех серверах.

Как CI/CD помогает?

Контекст

Контекст ’

Контекст ’’

Dev

Test

Prod

Изменение

2 docker-compose.yml

```
@@ -8,4 +8,4 @@ services:
      volumes:
      - ./future-devops:/future-devops
      - ./static:/static
-     - ./default.conf:/etc/nginx/conf.d/default.conf
+     - ./nginx.conf:/etc/nginx/nginx.conf
```

Принципы

- Создать повторяемый и надежный процесс поставки ПО
- Автоматизировать все , что можно . Интерфейс с кнопками
- Хранить все в системе контроля версий
- Получать раннюю обратную связь
- Совершенствоваться через повторения
- Встроенный контроль качества
- Выполнено , значит зарелизилось
- Релизим именно то , что было протестировано
- Каждый отвечает за процесс поставки ПО

Принципы

- Создать повторяемый и надежный процесс поставки ПО
- Автоматизировать все , что можно . Интерфейс с кнопками
- Хранить все в системе контроля версий
- Получать раннюю обратную связь
- Совершенствоваться через повторения
- Встроенный контроль качества
- Выполнено , значит зарелизилось
- Релизим именно то , что было протестировано
- **Каждый отвечает за процесс поставки ПО**

Среда разработки

CI

QA

Staging

Production

Dev

Я знаю, какие изменения будут в новом релизе, как мне передать это коллегам?

Ops

QA

Среда разработки

CI

QA

Staging

Production

Dev

Я знаю, какие проблемы бывают в продакшене , а еще могу сделать мониторинг для всех тестовых окружений

Ops

QA

Среда разработки

CI

QA

Staging

Production

Dev

Ops

Ну держитесь !…

QA

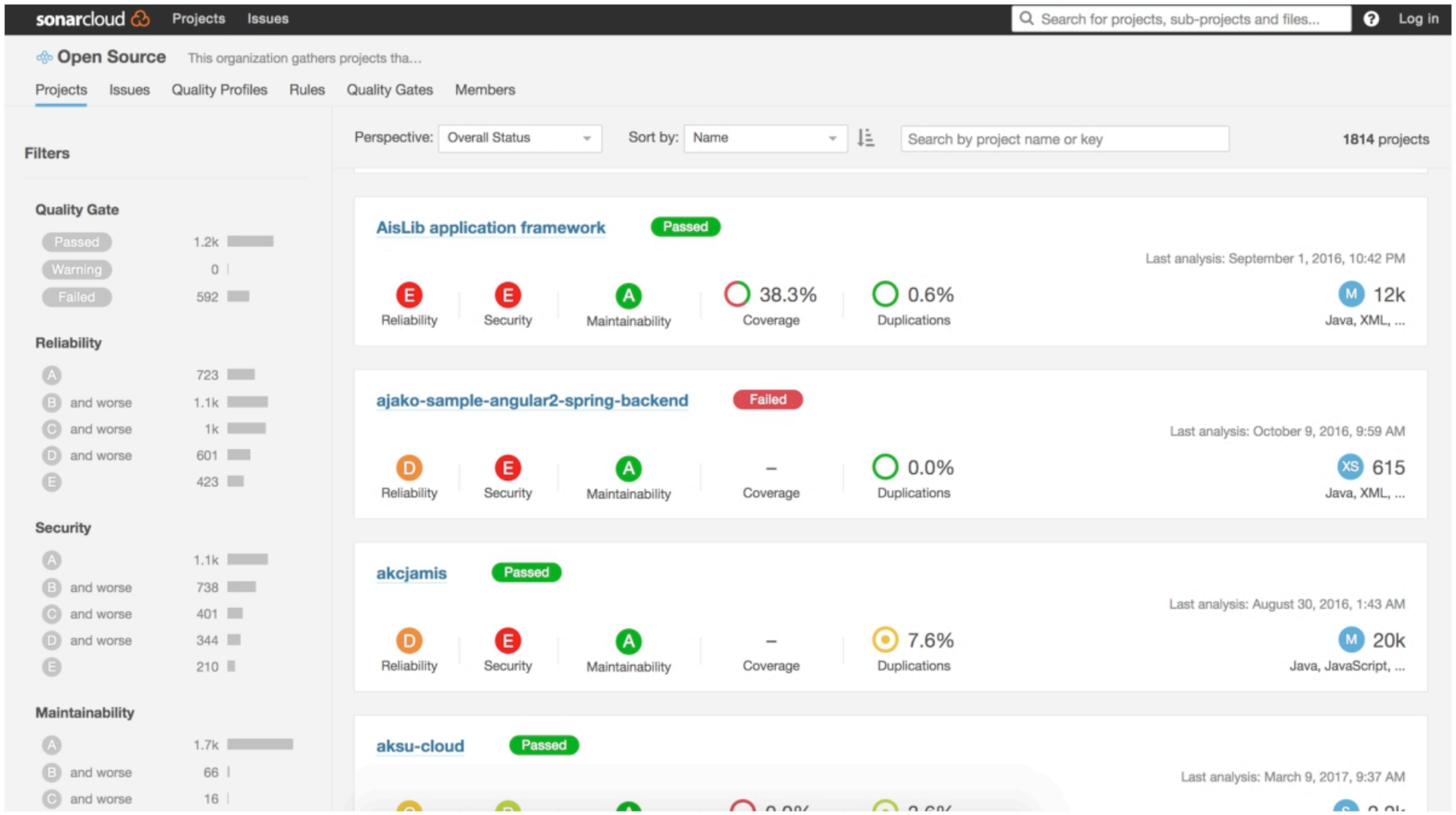
Обратная связь

- Светофор
- Отчеты о тестировании
- Контроль качества кода
- Уведомления о статусах в чат или почту
- Интеграция с таск-трекером

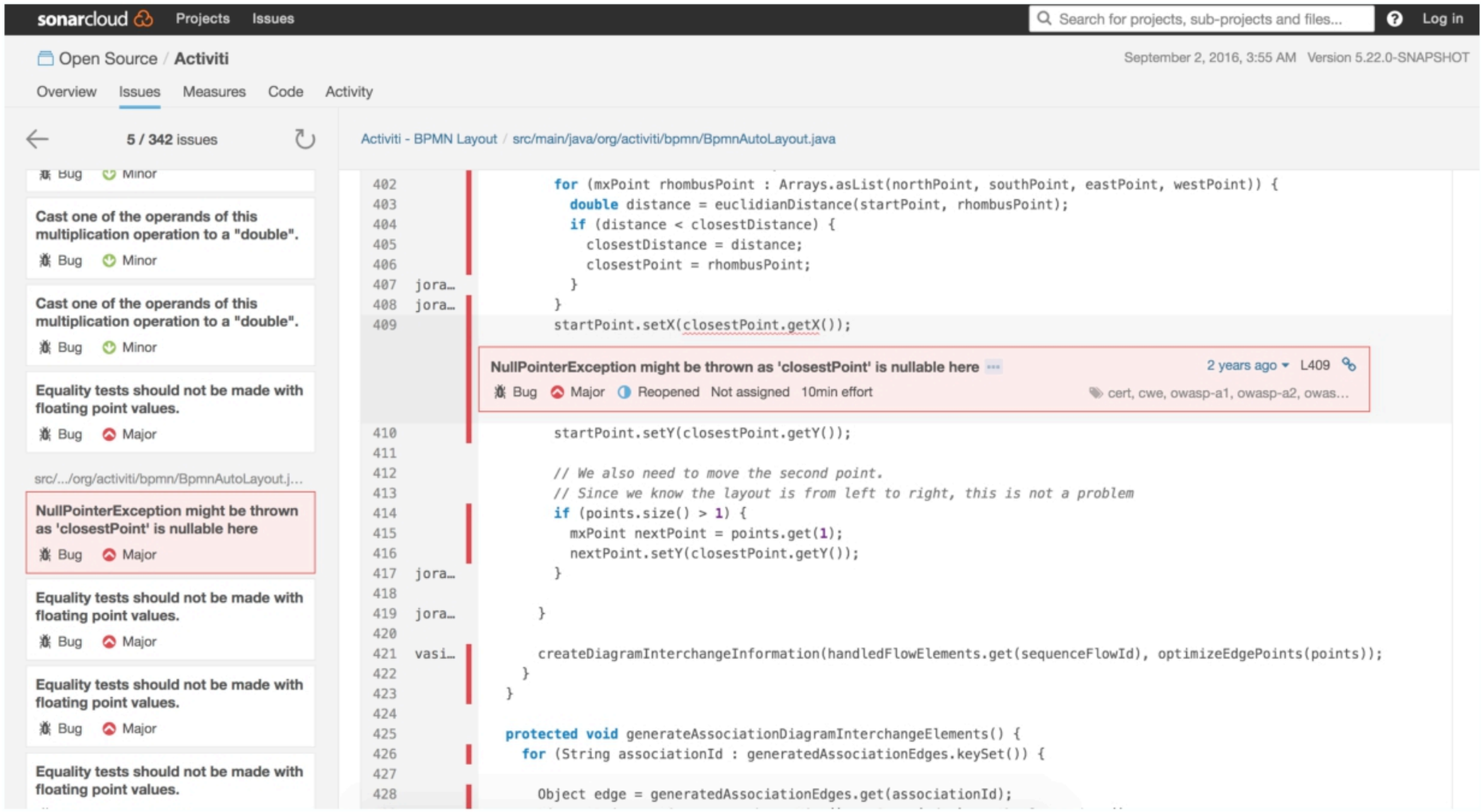
Контроль качества кода

- Измерения и визуализации
  - Покрытие тестами
  - Технический долг
  - Поиск потенциальных багов и уязвимостей
- Помогает найти потенциальные проблемы
- Помогает отслеживать тренды

SonarQube


sonarcloud
Projects
Issues
Search for projects, sub-projects and files...
Log in
Open Source
This organization gathers projects tha...
Projects
Issues
Quality Profiles
Rules
Quality Gates
Members
Filters
Perspective: Overall Status
Sort by: Name
Search by project name or key
1814 projects
Quality Gate
Passed 1.2k
Warning 0
Failed 592
Reliability
A 723
B and worse 1.1k
C and worse 1k
D and worse 601
E 423
Security
A 1.1k
B and worse 738
C and worse 401
D and worse 344
E 210
Maintainability
A 1.7k
B and worse 66
C and worse 16
AisLib application framework
Passed
Last analysis: September 1, 2016, 10:42 PM
E Reliability
E Security
A Maintainability
38.3% Coverage
0.6% Duplications
M 12k
Java, XML, ...
ajako-sample-angular2-spring-backend
Failed
Last analysis: October 9, 2016, 9:59 AM
D Reliability
E Security
A Maintainability
– Coverage
0.0% Duplications
XS 615
Java, XML, ...
akcjamis
Passed
Last analysis: August 30, 2016, 1:43 AM
D Reliability
E Security
A Maintainability
– Coverage
7.6% Duplications
M 20k
Java, JavaScript, ...
aksu-cloud
Passed
Last analysis: March 9, 2017, 9:37 AM

SonarQube


sonarcloud
Projects
Issues
Search for projects, sub-projects and files...
Log in
Open Source / Activiti
September 2, 2016, 3:55 AM Version 5.22.0-SNAPSHOT
Overview
Issues
Measures
Code
Activity
5 / 342 issues
Activiti - BPMN Layout / src/main/java/org/activiti/bpmn/BpmnAutoLayout.java
Bug Minor
Cast one of the operands of this multiplication operation to a "double".
Bug Minor
Equality tests should not be made with floating point values.
Bug Major
src/.../org/activiti/bpmn/BpmnAutoLayout.j...
NullPointerException might be thrown as 'closestPoint' is nullable here
Bug Major
Equality tests should not be made with floating point values.
Bug Major
Equality tests should not be made with floating point values.
Bug Major
Equality tests should not be made with floating point values.
for (mxPoint rhombusPoint : Arrays.asList(northPoint, southPoint, eastPoint, westPoint)) {
double distance = euclidianDistance(startPoint, rhombusPoint);
if (distance < closestDistance) {
closestDistance = distance;
closestPoint = rhombusPoint;
}
startPoint.setX(closestPoint.getX());
NullPointerException might be thrown as 'closestPoint' is nullable here
2 years ago L409
Bug Major Reopened Not assigned 10min effort
cert, cwe, owasp-a1, owasp-a2, owas...
startPoint.setY(closestPoint.getY());
// We also need to move the second point.
// Since we know the layout is from left to right, this is not a problem
if (points.size() > 1) {
mxPoint nextPoint = points.get(1);
nextPoint.setY(closestPoint.getY());
}
}
createDiagramInterchangeInformation(handledFlowElements.get(sequenceFlowId), optimizeEdgePoints(points));
}
}
protected void generateAssociationDiagramInterchangeElements() {
for (String associationId : generatedAssociationEdges.keySet()) {
Object edge = generatedAssociationEdges.get(associationId);

Хранение артефактов

Результатом билда иногда является артефакт
Например : jar, war, rpm, zip, docker image

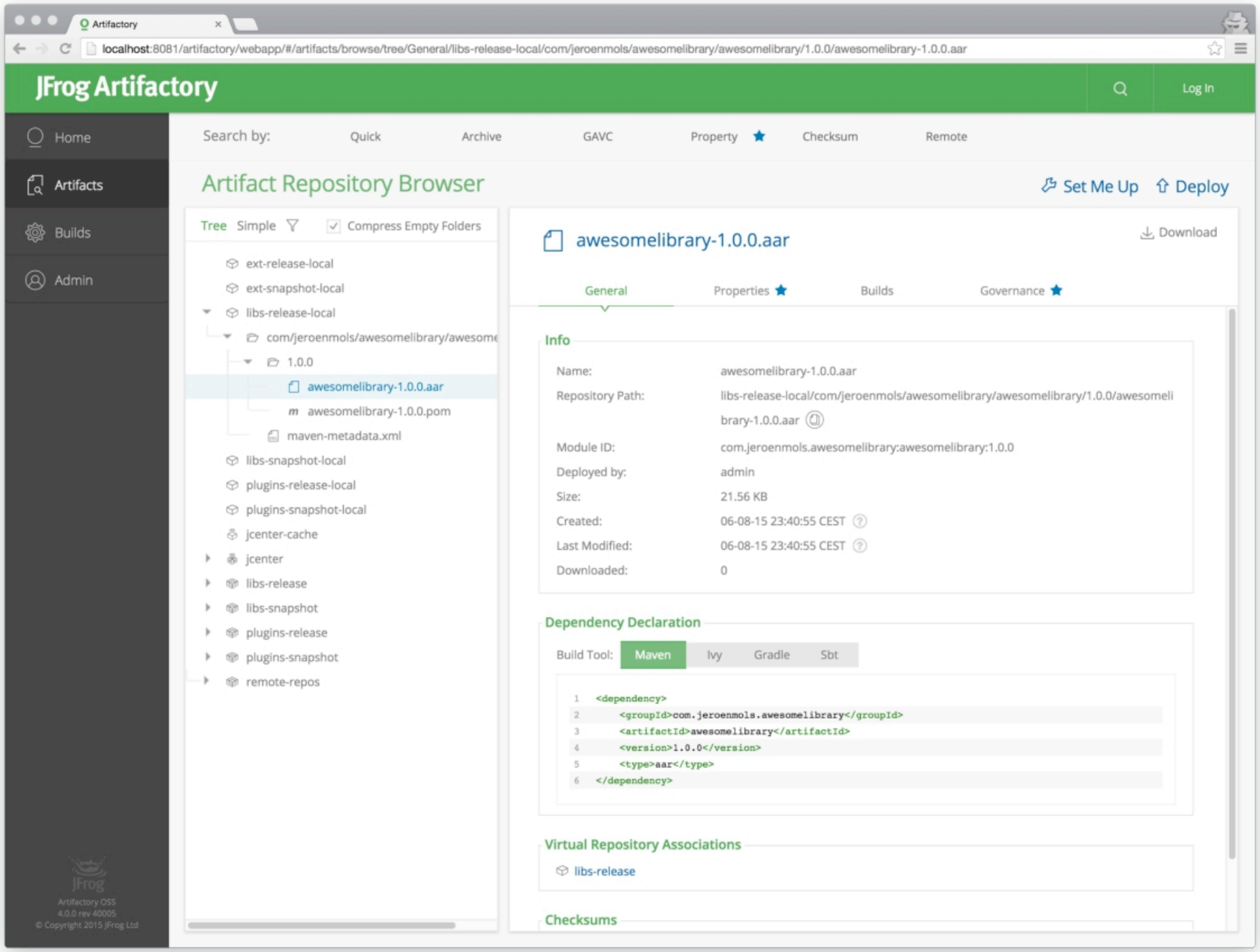
Исходный код	Артефакт
Текст	Бинарник
Diffable	Not diffable
Версионируется по содержимому	Версионируется по имени
Перезаписывается	**Не должен перезаписываться**

Хранение артефактов

- Artifactory
- Nexus
- Archiva

- Прокси для внешних репозиториев
- Управление жизненным циклом артефакта
- Модель управления доступом
- Поиск
- REST API
- Single source of truth

Хранение артефактов

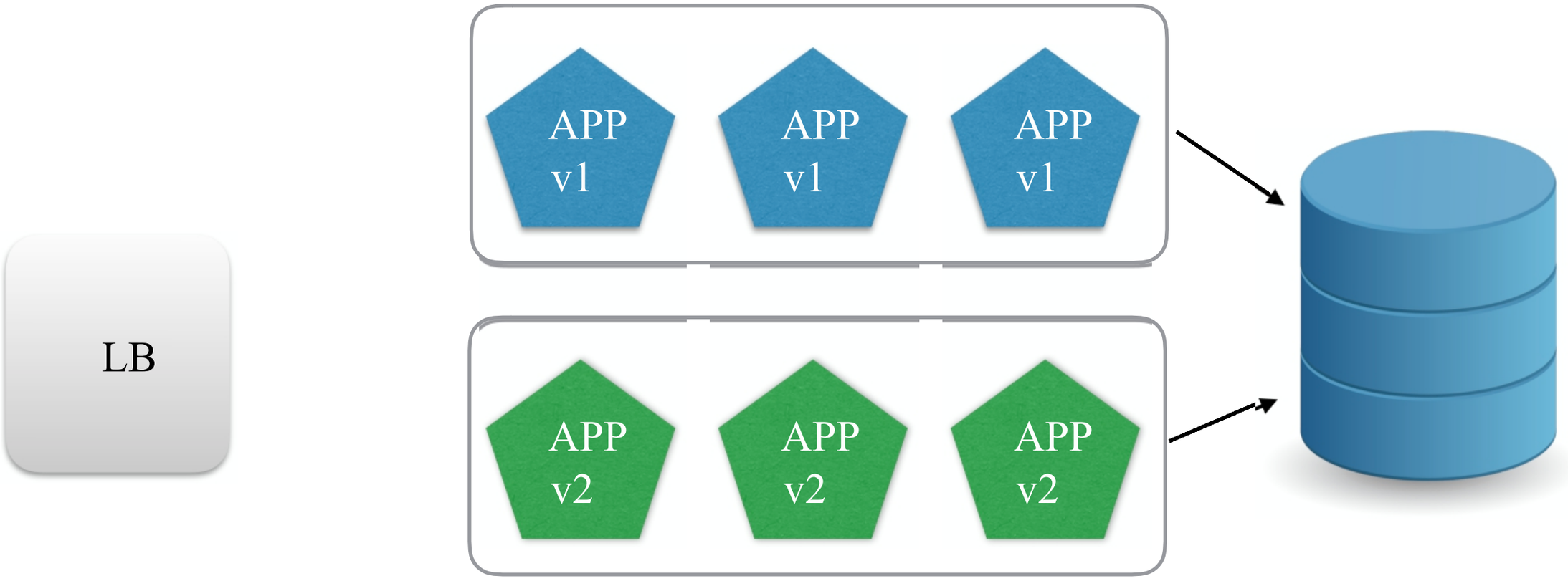


Стратегии выкатки

Для того , чтобы нам автоматически выкатывать изменения нужно реализовать обновление без простоев (zero-downtime deploy)

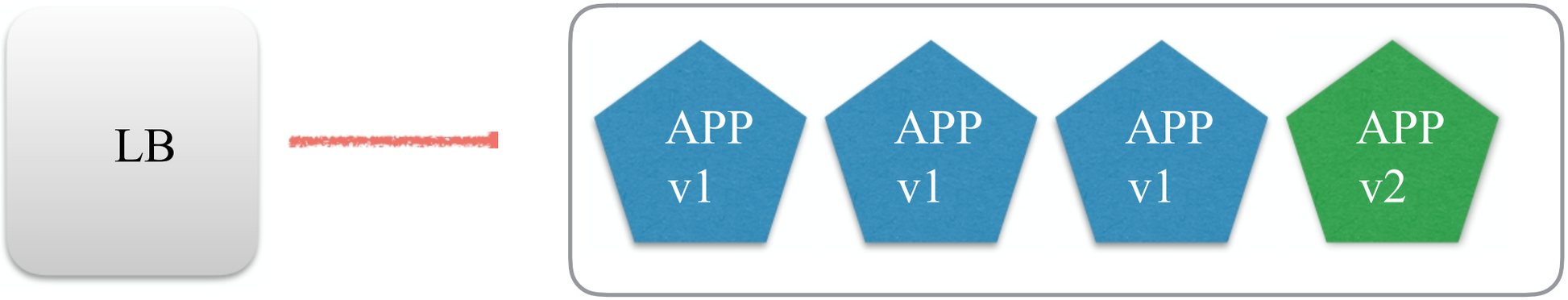
- Blue-green
- Canary
- Rolling

Blue-green deployment

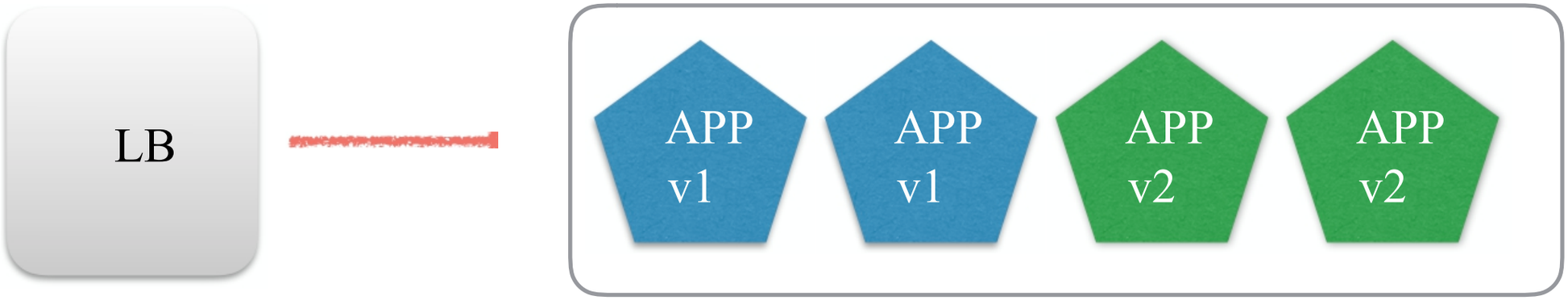


Blue-green deployment

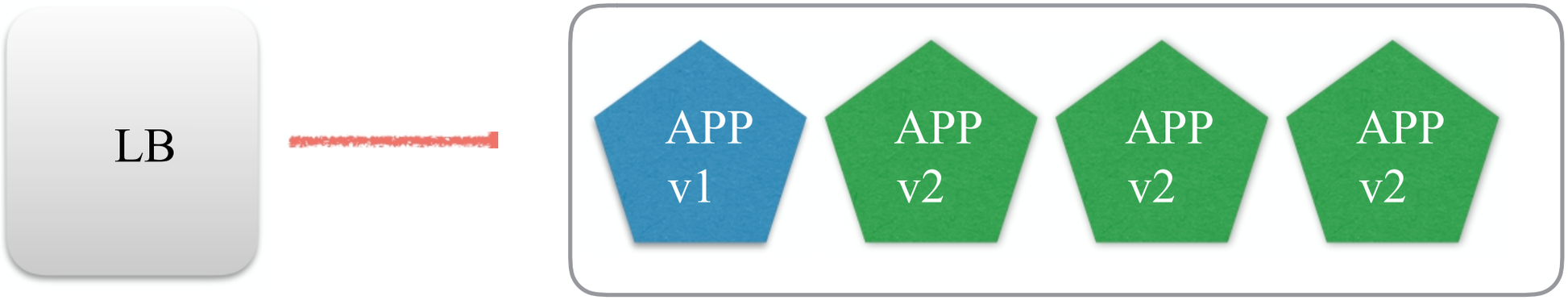
Canary deployment



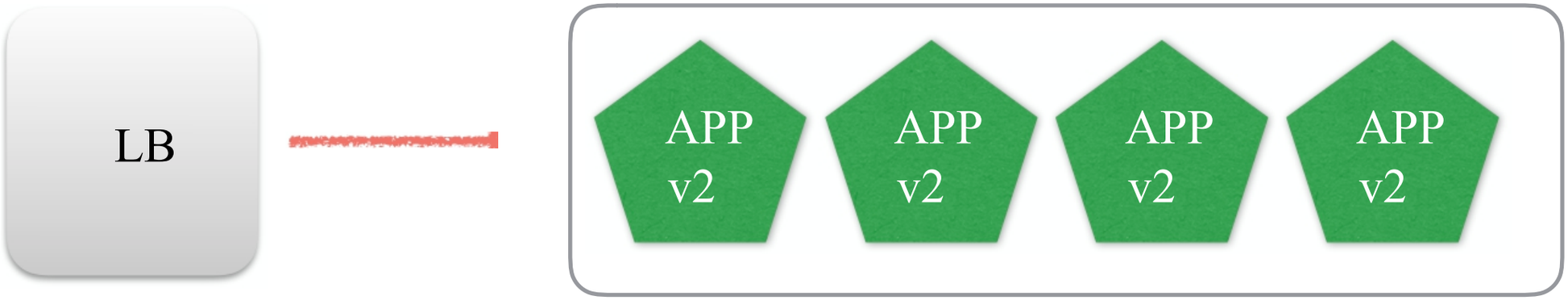
Rolling deployment



Rolling deployment



Rolling deployment



Нюансы

- Интеграция инструментов в процссы CI/CD
  - Как по созданию ветки в git создать окружение ?
  - Как будет запускаться ansible, настраивающий окружения?
  - Как подключать новые окружения к мониторингу ?
- CI-сервис в больших организациях
  - Windows / Linux / AIX / Docker
  - Много продуктов со своей спецификой
  - Состояние сборочных агентов уплывает
- CI/CD для компонент, поставляемых вендором

Метрики

- Скорость
- Качество

Метрики скорости

- Mean Lead Time – время на билд, тесты, деплой
- Daily Change Rate – успешных изменений в день
- Mean Time To Environment – время на подготовку нового окружения
- Mean Time To Detect – как долго баг может быть необнаруженным

Метрики качества

- Build Failure Rate – процент неуспешных сборок
- Deployment Failure Rate – процент неуспешных выкаток
- Infra-Related Failure Rate – процент неуспещных выкаток / сборок из -за проблем инфраструктуры
- Automated Detection Rate – процент проблем, выявленных на этапах тестирования

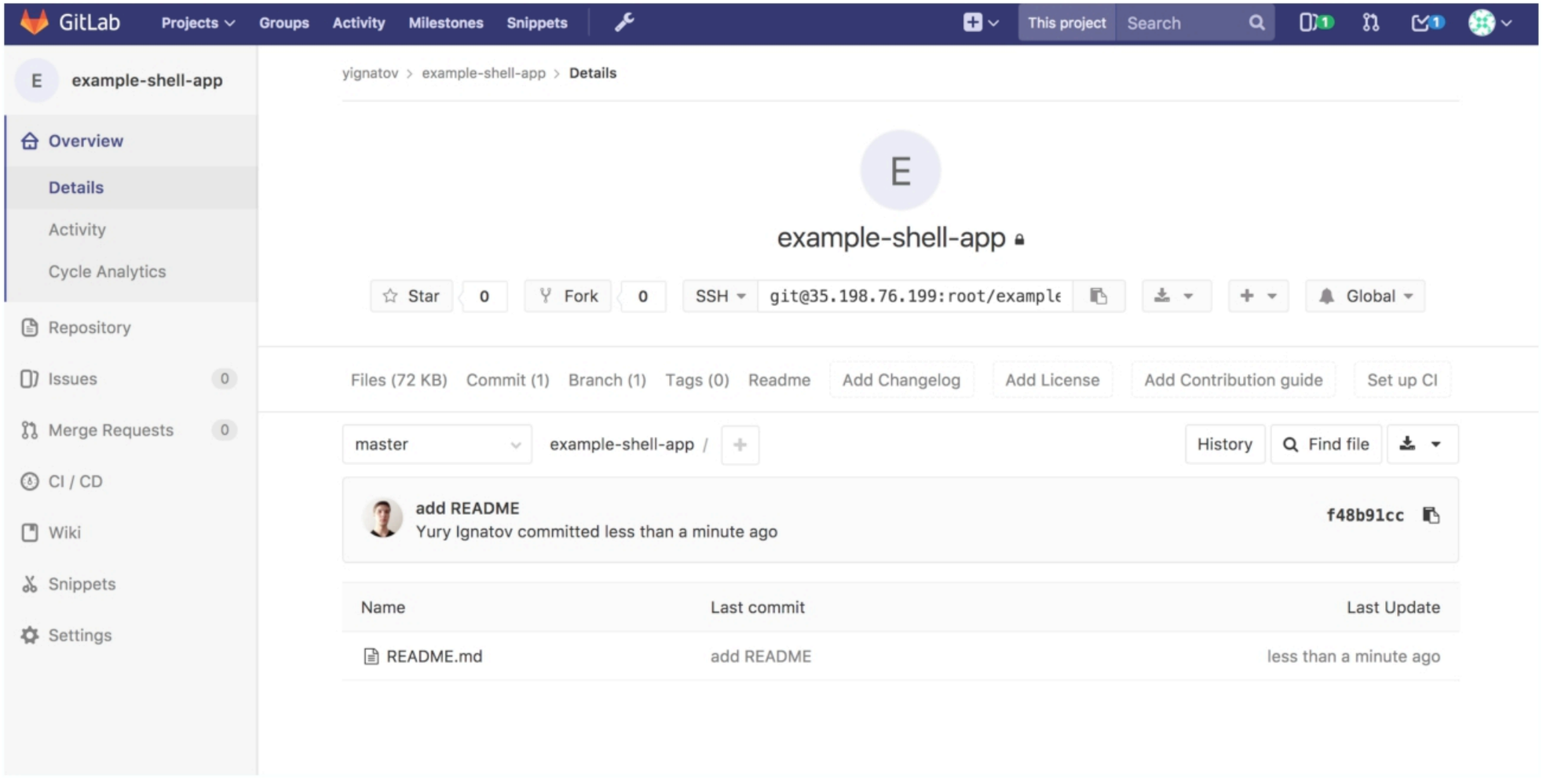
Инструменты

- Jenkins
- Bamboo от Atlassian
- TeamCity от JetBrains
- GitLab CI
- TravisCI
- Concourse CI
- Go CD
- Rundeck

Почему Gitlab CI

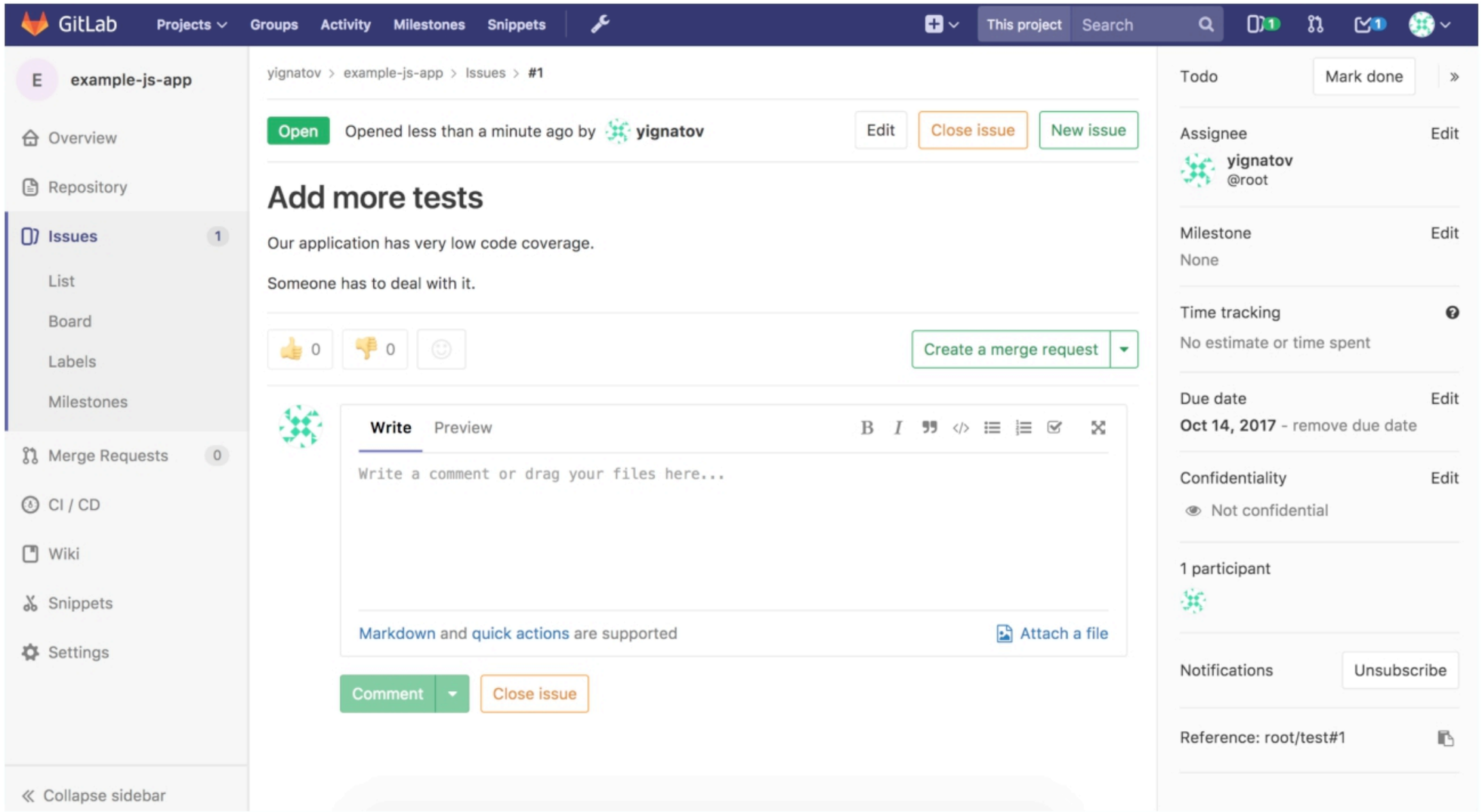
- Уже интегрирован в экосистему Gitlab (VCS, Issues, Docker registry!)
- Отдельный CI процесс под репозиторий (компонент)
- Быстрая обратная связь
- Визуализация процесса
- Достойная поддержка Docker
  - Тестирование внутри контейнеров
  - Тестирование с учетом других сервисов в контейнерах
- Обратная связь из коробки (много метрик , мониторинг)

Gitlab VCS

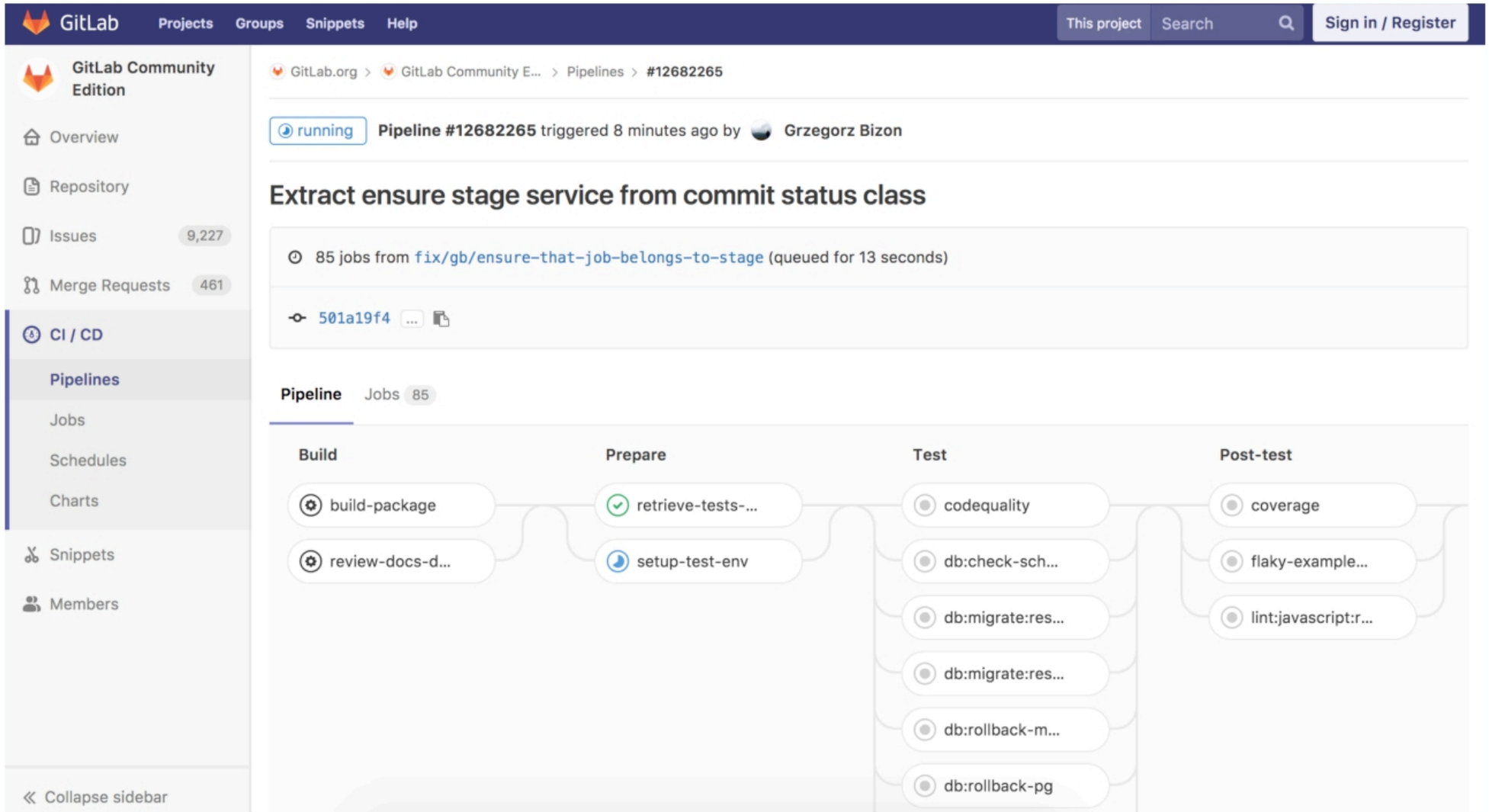


DevOps консультанты

Gitlab Issues



Gitlab Pipelines



Gitlab Pipelines