



УНИВЕРСИТЕТ ИТМО

Выпускная квалификационная работа

Автоматизации управления жизненным циклом веб-сервиса

Федюкович Семён Андреевич
Р3555

2. Цель и задачи



Цель исследования:

Развернуть и автоматизировать управление жизненным циклом веб-сервиса на базе Node.js

Задачи исследования:

1. Провести обзор необходимых средств для автоматизации управления жизненным циклом веб-сервиса
2. Рассмотреть полученный для развёртывания веб-сервис и проанализировать требования
3. Провести проектирование механизмов автоматизации управления жизненного цикла веб-сервиса
4. Составить план тестирования механизмов развёртывания веб-сервиса
5. Провести практические работы по развёртке и автоматизации управления жизненного цикла веб-сервиса
6. Провести тестирование механизмов развёртывания и обосновать полученные результаты

3. Требования к системе



1. Использовать современные DevOps методологии
2. Взаимодействие с конечным пользователем по методологии SaaS
3. Возможность добавления в систему не описанных заранее сервисов по методологии IaC
4. Использование методологии CI/CD для взаимодействия с инфраструктурой и организации контроля качества
5. Наличие конфигурации прав доступа (приватизации исходного кода)
6. Наличие хранилища пакетов и образов
7. Стоимость установки и обслуживания системы - не более 2 500 рублей за установку 2 500 рублей в месяц за обслуживание на момент написания данной работы

4. Сравнение Git хостингов

Функция	GitHub	BitBucket	GitLab	Space
Настройки доступа	Ограничены	Включены в подписку	Включены в подписку	Включены в подписку
Хранилище контейнеров и пакетов	500 Mb	Отсутствует	10 Gb	10 Gb
Инструменты CI/CD	Ограничены	Ограничены	Ограничены, дополнительные функции	Ограничены
Количество пользователей	Не ограничено	5	Не ограничено	Не ограничено

Таблица 1. Сравнение Git хостингов

5. Ключевые преимущества GitLab

1. Гибкие настройки прав доступа без ограничений
2. Расширенные возможности и функции CI/CD
3. Хранилище пакетов и контейнеров 10 Gb
4. Неограниченное количество пользователей в группах и проектах
5. Крупное сообщество пользователей, доступная и понятная документация

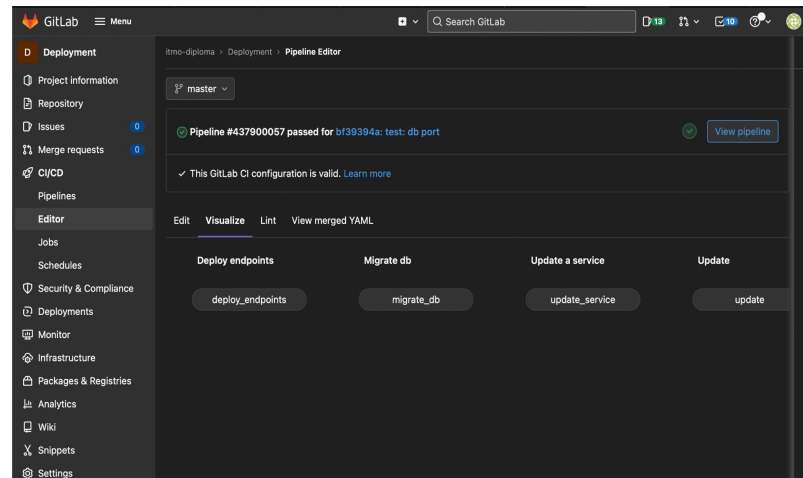


Рисунок 1. Скриншот GitLab

6. Диаграмма развёртывания веб-сервиса

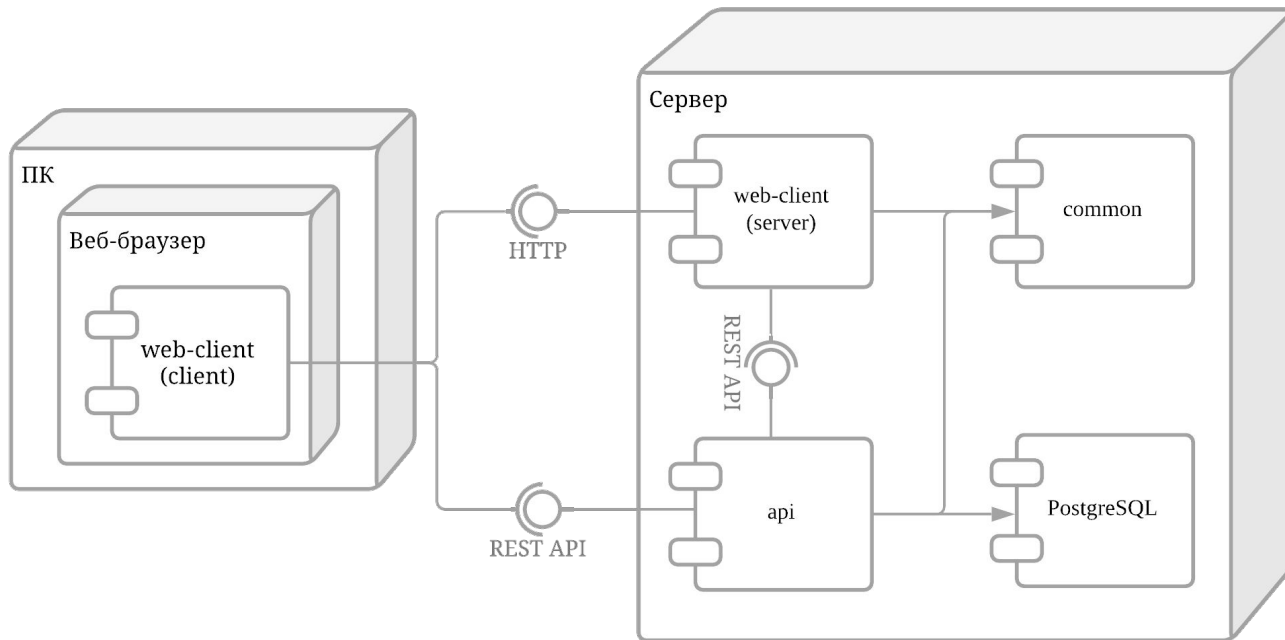


Рисунок 2. Диаграмма развёртывания веб-сервиса

7. Диаграмма случаев использования

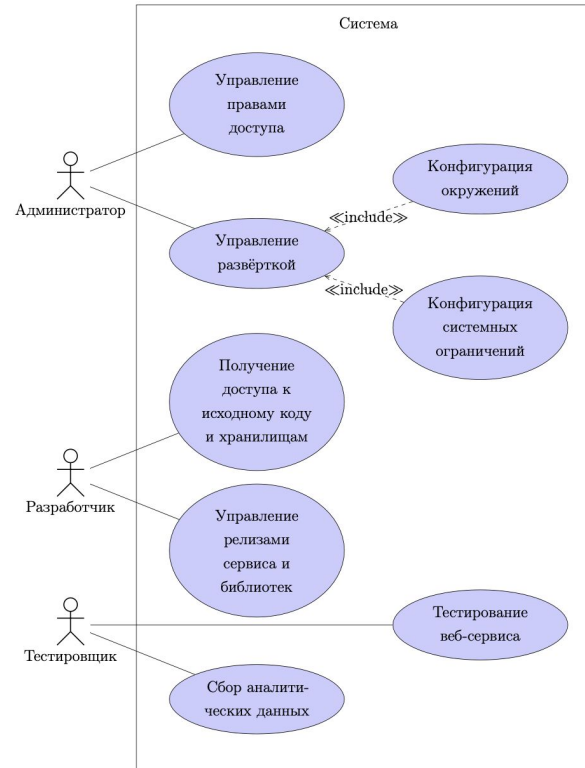


Рисунок 3. Диаграмма случаев использования

8. Диаграмма последовательностей линий и задач

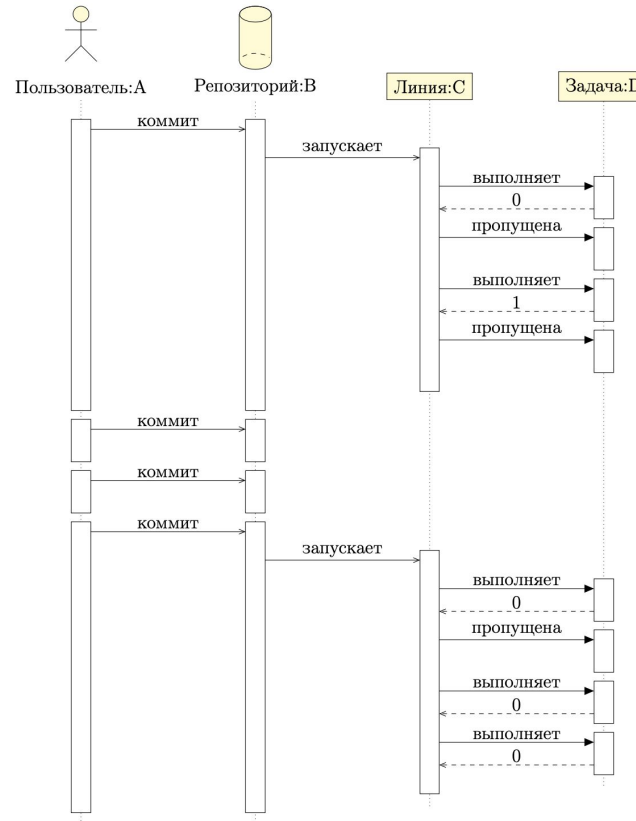


Рисунок 4. Диаграмма последовательностей линий и задач

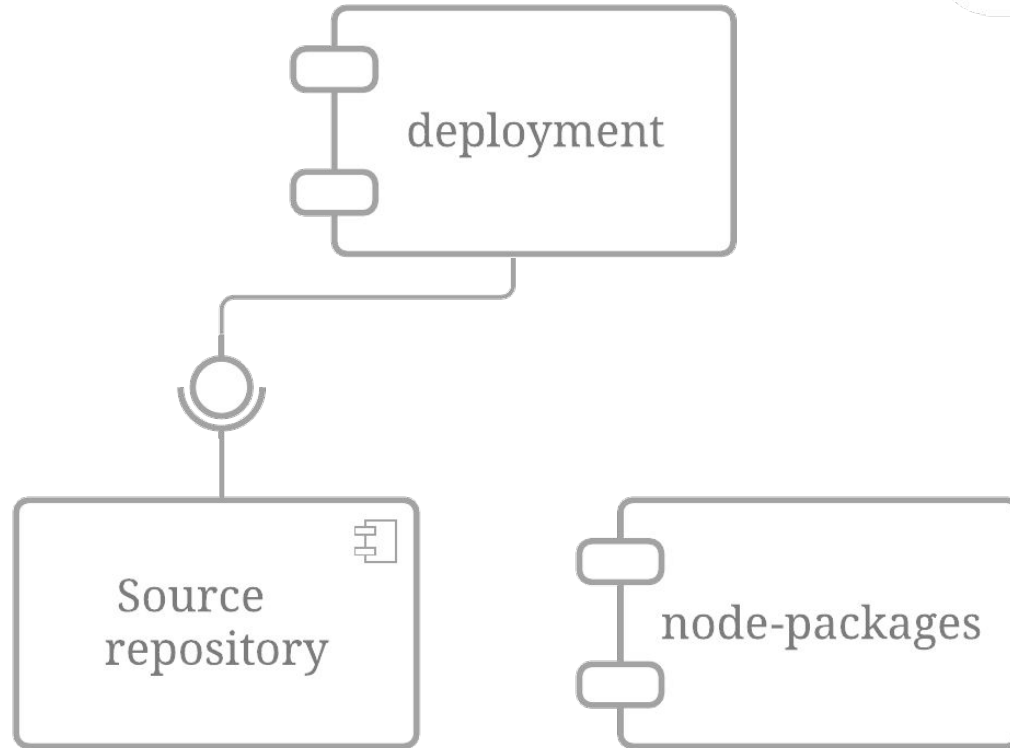


Рисунок 5. Диаграмма компонентов репозиториев



Название вида тестирования	Условие запуска	Расположение	Типы ошибок
Модульное	Каждый коммит	Репозитории с исходным кодом	Семантические, компиляции, логические на уровне модуля
Интеграционное	Каждый день в 10 утра	Репозитории с конфигурациями развёртывания	Логические на уровне веб-сервиса

Таблица 2. Линии тестирования

11. Задача модульного тестирования на примере web-client

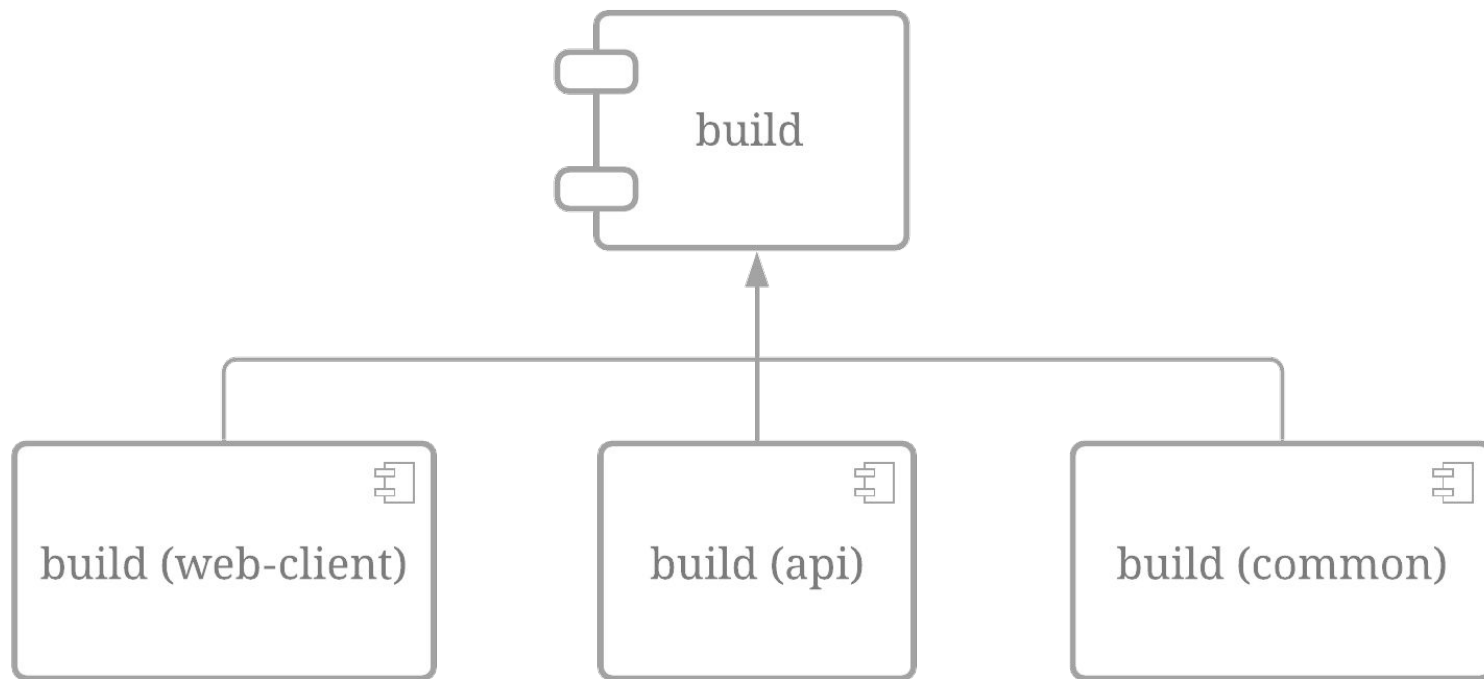
The screenshot displays the GitLab CI/CD interface for a project named 'Web-client'. The left sidebar contains navigation links: Project information, Repository, Issues (8), Merge requests (0), CI/CD, Pipelines, Editor, Jobs (selected), Schedules, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, and Analytics. The main panel shows the details of a job named 'unit'. The job status is 'finished' with a duration of 1 minute 12 seconds, completed 1 month ago. The runner is #10521693 (1nThPPGJ) using the 'builder' image. The job artifacts section indicates that the latest artifacts will not be deleted until newer ones are available. The commit is 569779ef refactor. The pipeline is #14632745 for dev. The job log shows the following steps: 52 ✓ should parse locale with any, 53 make class name, 54 ✓ should make empty class (1 ms), 55 ✓ should make class by strings, 56 ✓ should make class by objects, 57 ✓ should make class by objects and strings, 58 Test Suites: 1 passed, 1 total, 59 Tests: 13 passed, 13 total, 60 Snapshots: 0 total, 61 Time: 3.899 s, 62 Ran all test suites, 64 Saving cache for successful job (00:01), 65 Creating cache dev..., 66 WARNING: /root/.npm: no matching files, 67 Archive is up to date!, 68 Created cache, 70 Uploading artifacts for successful job (00:02), 71 Uploading artifacts..., 72 junit.xml: found 1 matching files and directories, 73 Uploading artifacts as "junit" to coordinator... ok id=1810964955 responseStatus=201 Created token=voQitHq-, 75 Cleaning up file based variables (00:01), and 77 Job succeeded.

12. Конфигурация планирования интеграционного тестирования

The screenshot shows the GitLab web interface for configuring a new pipeline schedule. The left sidebar contains navigation links: Deployment, Project information, Repository, Issues (0), Merge requests (0), CI/CD, Pipelines, Editor, Jobs, Schedules (selected), Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, Snippets, and Settings. The main content area is titled 'Schedule a new pipeline' and includes the following fields:

- Description:** A text input field containing 'Integration tests'.
- Interval Pattern:** Radio buttons for scheduling options: 'Every day (at 9:00am)', 'Every week (Thursday at 9:00am)', 'Every month (Day 15 at 9:00am)', and 'Custom (Cron syntax)' (selected). Below the radio buttons is a text input field containing '0 10 * * *'.
- Cron Timezone:** A dropdown menu set to 'UTC'.
- Target Branch:** A dropdown menu set to 'master'.
- Variables:** Two rows of variable configuration. The first row has 'Variable' (dropdown), 'BRANCH', 'develop', and a close button 'X'. The second row has 'Variable' (dropdown), 'MODE', 'integration', and a close button 'X'.

Рисунок 7. Конфигурация планирования интеграционного тестирования



Цель тестирования:

Снижение количества ошибок в работе линий и задач, а так-же конфигурационных файлов

Тестирование проводилось разработчиком

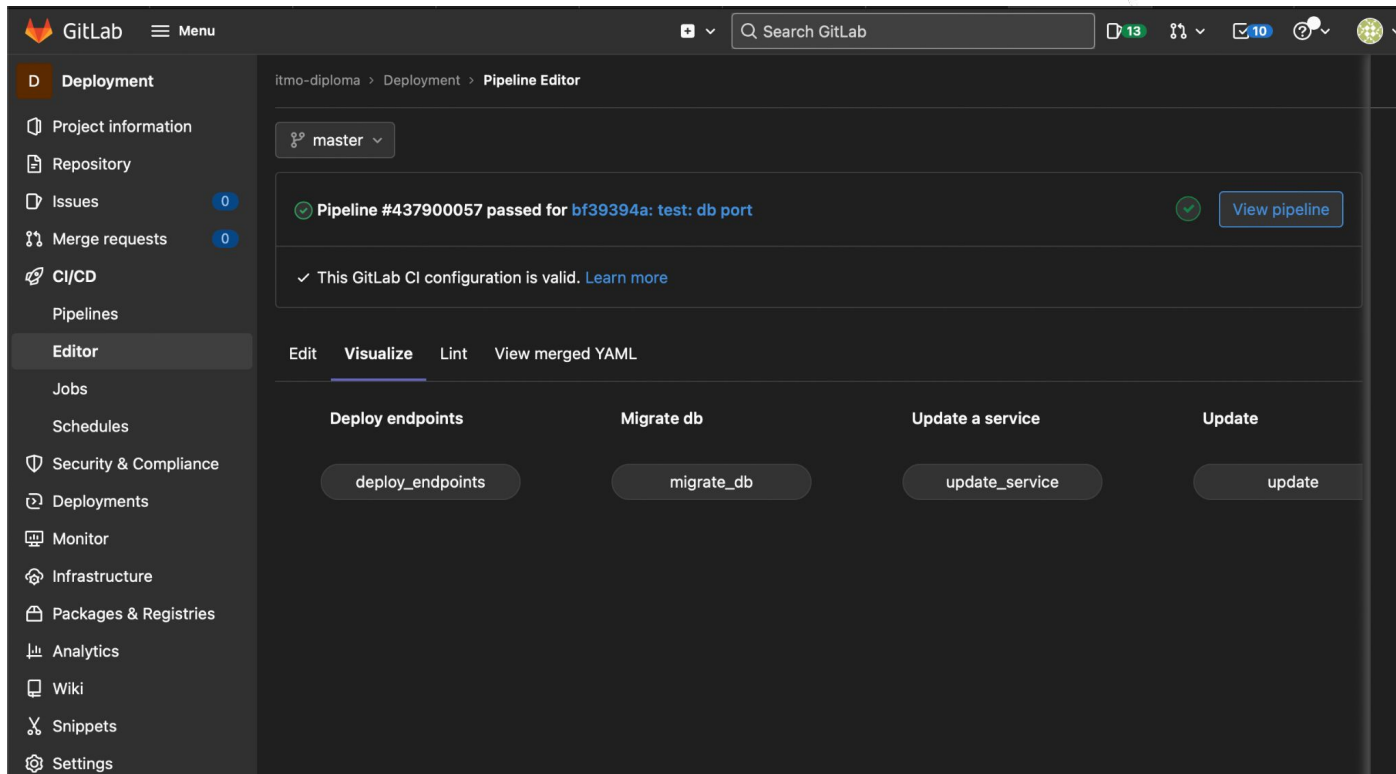
Методология:

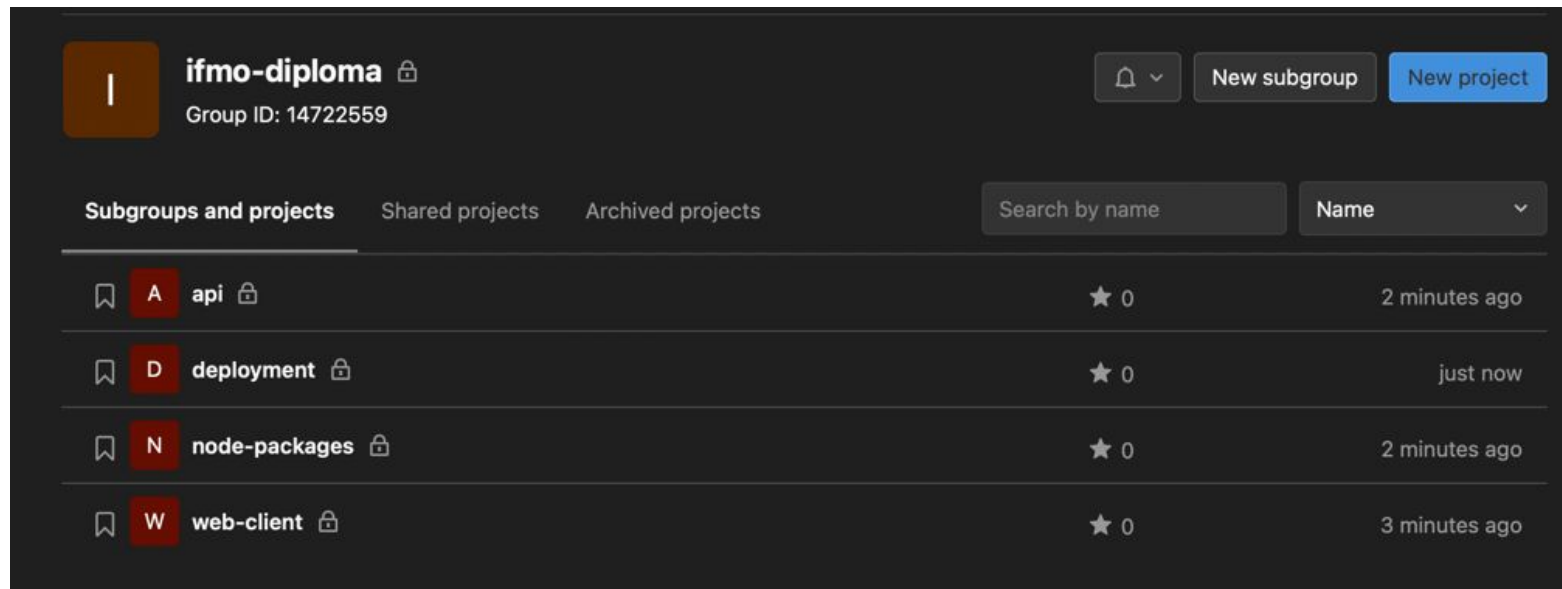
Ручное тестирование белого ящика

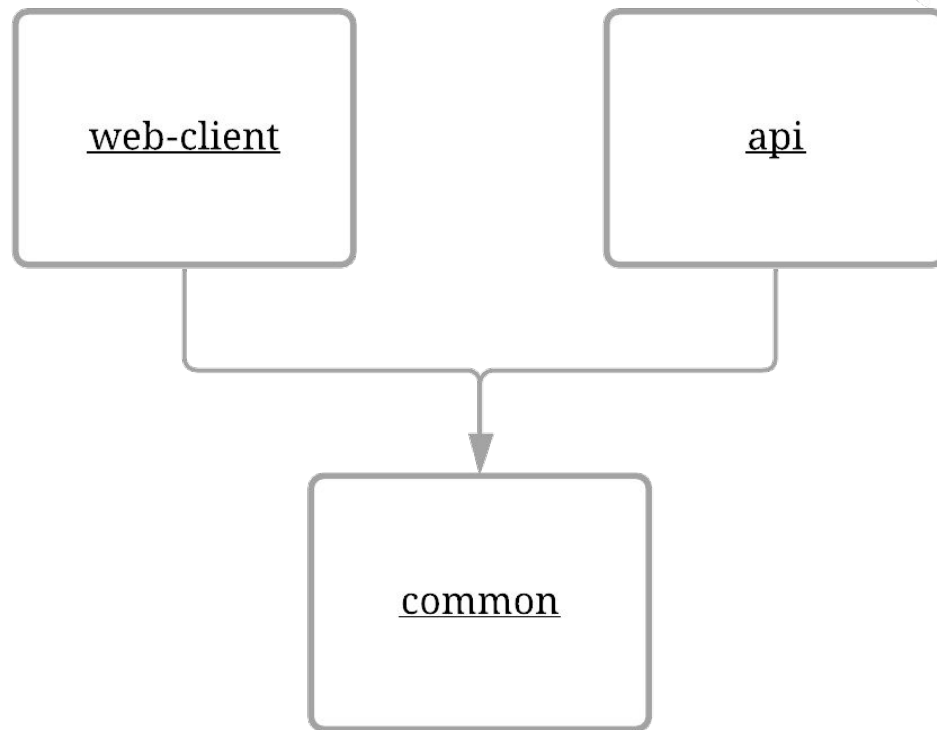
Тестовые сценарии:

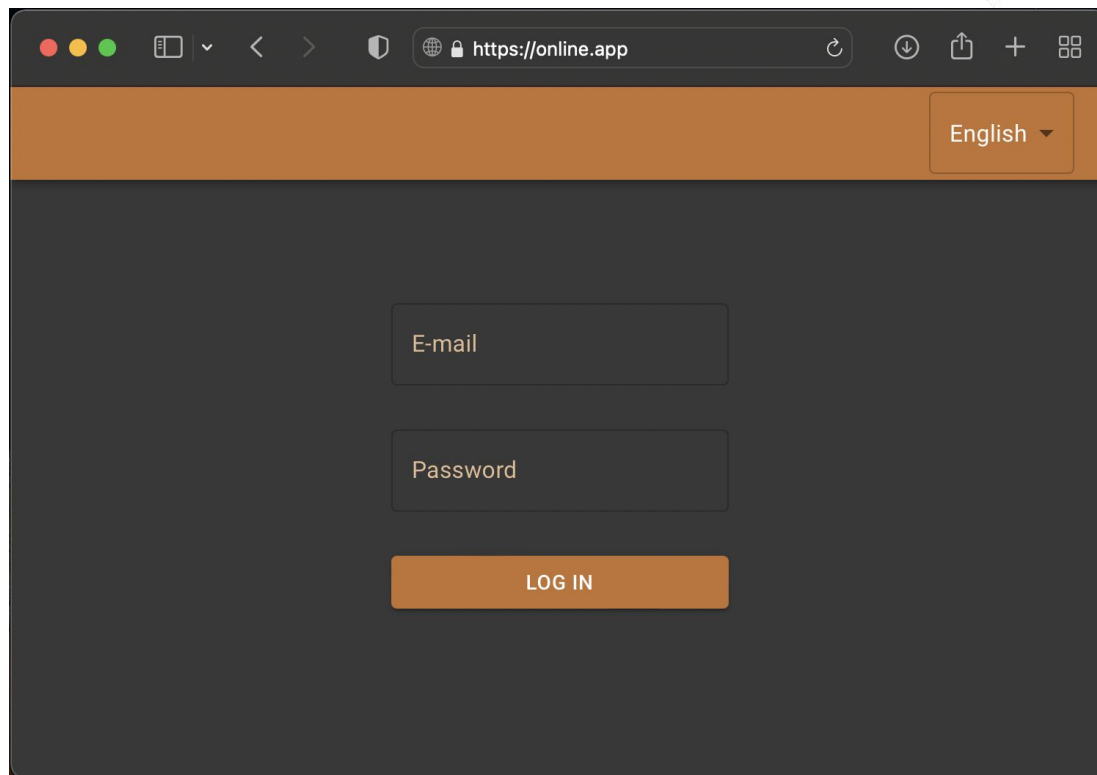
1. Линии задач автоматизации тестирования
2. Линии задач релиза создания релиза
3. Линии задач управления развёртыванием

15. Использование GitLab Runner эмулятора









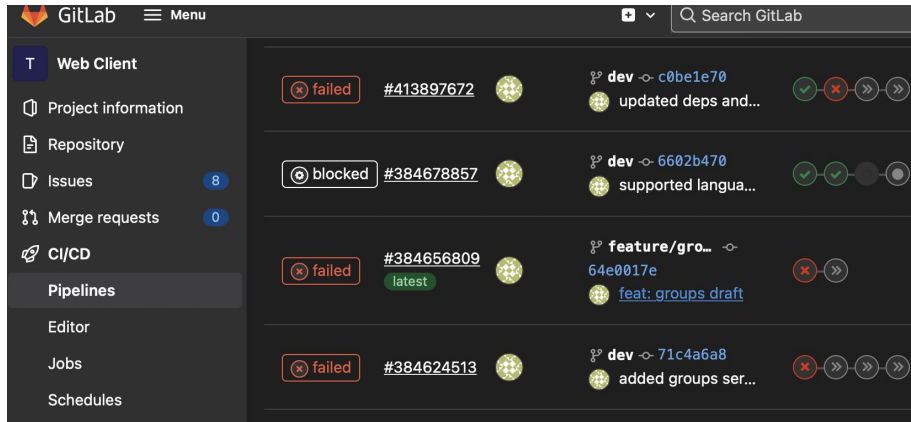


Рисунок 13. Линии задач web-client

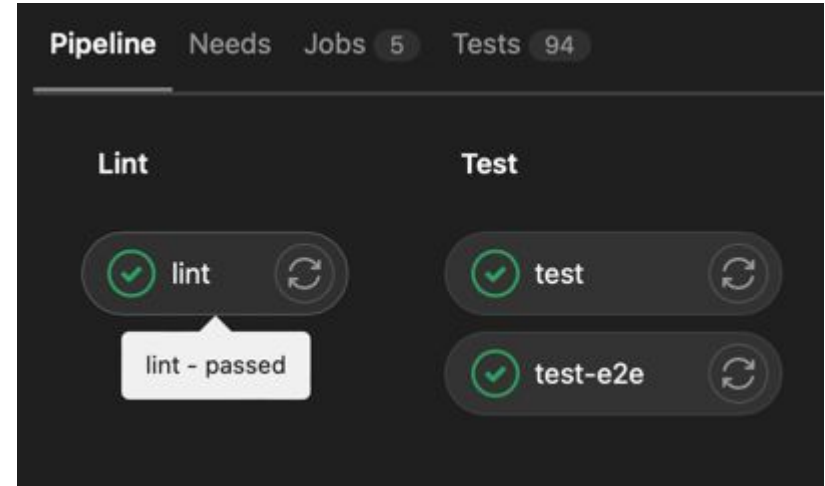


Рисунок 14. Линии задач в эмуляторе

20. Линии задач создания релиза

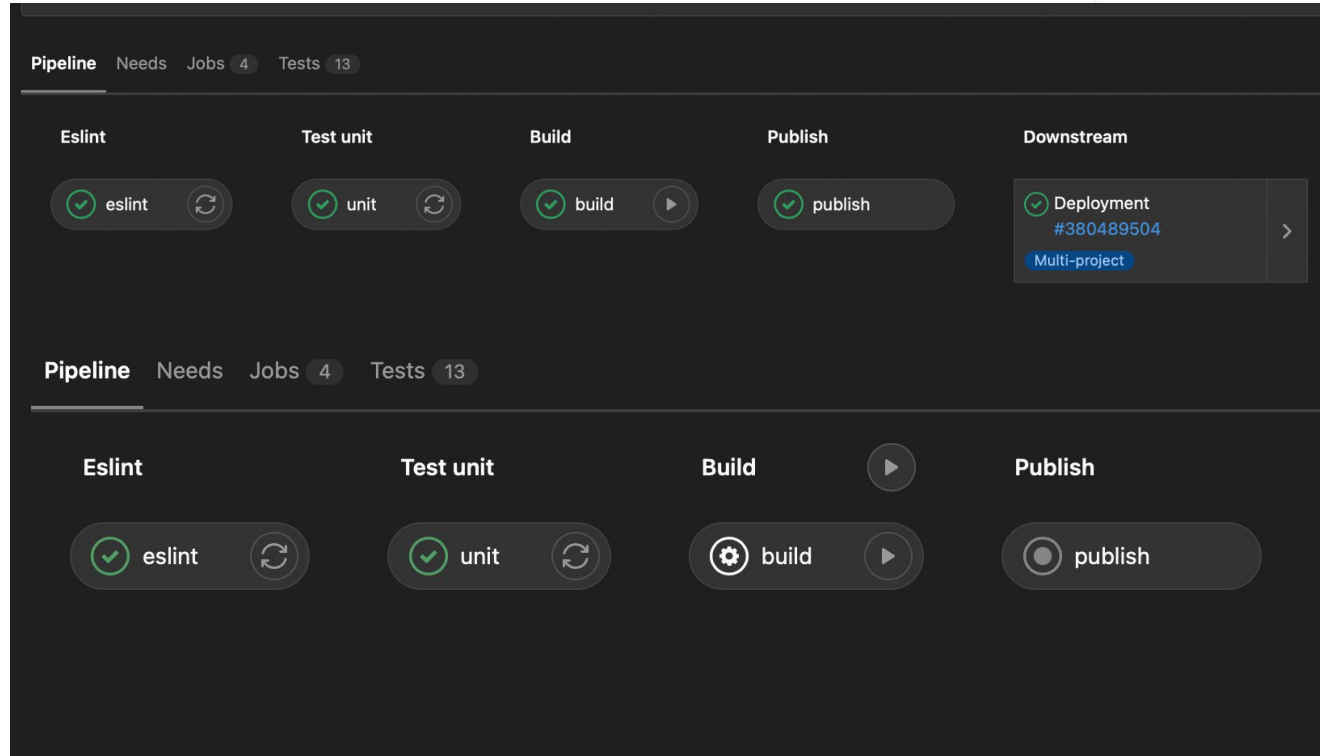


Рисунок 15. Линии задач создания релиза

21. Линии задач управления развёртыванием

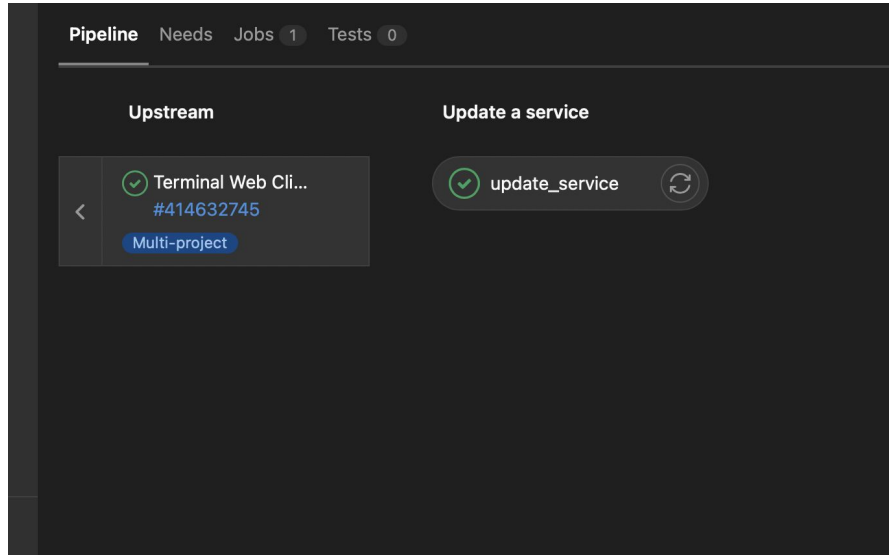


Рисунок 16. Управление развёртыванием сервиса компонента

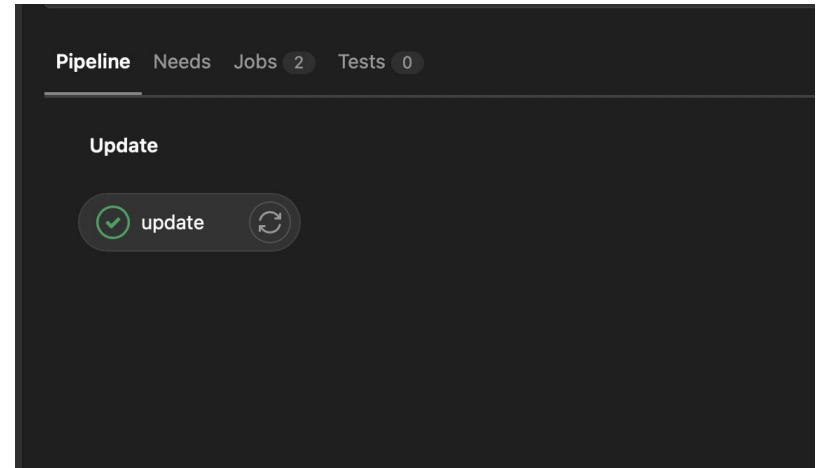


Рисунок 17. Управлением развёртыванием веб-сервиса

Спасибо за внимание!

IT'sMO^{re} than a
UNIVERSITY