Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

# Разработка серверного приложения для автоматизации сборки и выпуска релизов

Отчёт по проектному практикуму

Команда «Purrpurr»

Участники:

Балакин Кирилл Александрович РИ-320910

Сысоева Елена Павловна РИ-320940

Тарасенко Елена Витальевна РИ-320944

Хренова Дарья Валерьевна РИ-320930

Куратор:

Пиксаева Анастасия Юрьевна

Екатеринбург 2025

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 2

ВВЕДЕНИЕ 3

ОПИСАНИЕ ПРОЕКТА 4

Описание проблемы 4

Описание целевой аудитории 4

Описание решения 4

ОСНОВНАЯ ЧАСТЬ 5

Область проекта 5

Обзор аналогов 5

Результаты проекта 5

ОТЧЁТЫ О РАБОТЕ КАЖДОГО УЧАСТНИКА 6

Отчёт Балакина Кирилла Александровича 6

Отчёт Сысоевой Елены Павловны 6

Отчёт Тарасенко Елены Витальевны 6

Отчёт Хреновой Дарьи Валерьевны 6

РЕФЛЕКСИЯ И ВЫВОДЫ КОМАНДЫ 7

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 8

ПРИЛОЖЕНИЕ А 9

ВВЕДЕНИЕ

В современную эпоху цифровых технологий всё больше людей используют различные программы и приложения, которые постоянно совершенствуются и обновляются. Каждое изменение в таких продуктах требует тщательной проверки и документации, что осуществляется через выпуск релизов. Однако этот процесс включает множество рутинных этапов: создание релизной ветки, проверка задач, контроль статусов изменений, интеграция нововведений в основную ветку и многое другое. Чтобы избавить команду от этих однообразных действий и ускорить процесс, нашему заказчику потребовалось веб-приложение для автоматизации выпуска релизов.

Для реализации проекта мы сформировали команду, в которую вошли тимлид-аналитик, два бэкенд-разработчика, фронтенд-разработчик, а также дизайнер из другого учебного заведения.

В ходе работы мы использовали обучающие видеоматериалы, доступные в сети, а при возникновении сложных вопросов консультировались с нашим куратором. В качестве технологического стека был выбран React.js для фронтенда и Java Spring для бэкенда. Дизайн разрабатывался в Figma, а для организации аналитической работы использовались GitHub Projects, Microsoft Word, PowerPoint и платформа draw.io.

Основной задачей было разработать веб-приложение для уменьшения ручного труда при сборке релизов сервисов за счёт автоматизации процессов проверки задач, создания релизных веток и управления Merge Request.

ОПИСАНИЕ ПРОЕКТА

Описание проблемы

Проблема проекта заключается в том, что процесс выпуска релизов требует множества рутинных действий, таких как создание релизной ветки, проверка задач, контроль статусов изменений и интеграция нововведений. Это отнимает много времени и сил, поэтому необходимо разработать веб-приложение, которое автоматизирует эти процессы и упростит работу команды.

Описание целевой аудитории

Целевая аудитория проекта — это компании и команды разработчиков, занимающиеся созданием и поддержкой программного обеспечения. В первую очередь, это тимлиды, аналитики, бэкенд- и фронтенд-разработчики, которые участвуют в процессе выпуска релизов. Для них важно минимизировать рутинные задачи, автоматизировать контроль за изменениями и ускорить интеграцию обновлений. Также продукт может быть полезен менеджерам проектов, которым необходима прозрачность и удобство управления релизами.

Описание решения

Решением данной проблемы станет разработка веб-приложения, которое автоматизирует ключевые этапы выпуска релизов. Оно будет включать:

1. Автоматическую проверку задач, снижая вероятность ошибок в процессе релиза;
2. Создание релизных веток без необходимости вручную выполнять рутинные действия;
3. Управление Merge Request, обеспечивая удобное согласование и объединение изменений;
4. Отображение информации о diff-коммитах и задачах, вошедших в релиз, для полного контроля над обновлениями;
5. Интеграцию с YouTrack и GitLab, что позволит работать с уже используемыми инструментами;
6. Надежную защиту пользователей и данных, гарантируя безопасность работы с системой.

Такой подход значительно упростит процесс выпуска релизов, сократит время на рутинные операции и повысит удобство работы команды.

ОСНОВНАЯ ЧАСТЬ

Область проекта

Область проекта лежит на пересечении разработки программного обеспечения, автоматизации процессов и управления релизами. Он охватывает такие направления, как разработка веб-приложений, DevOps-практики, интеграция с системами управления проектами (YouTrack) и системами контроля версий (GitLab).

Проект направлен на оптимизацию процессов разработки, устранение рутинных задач при выпуске обновлений и повышение эффективности работы команд разработчиков. Важную роль играет также безопасность данных и пользователей, что делает его актуальным для ИТ-компаний, занимающихся разработкой и сопровождением ПО.

Обзор аналогов

В качестве основных альтернатив мы рассмотрели инструменты управления релизами, которые помогают командам планировать и контролировать передачу кода, изменений и настроек от команды разработчиков приложений, а также упаковку и развёртывание в тестовых средах со стороны операционной системы, вплоть до анализа рисков, оценки влияния и успешного развёртывания в производственных средах.

Многие из них пользуются технологией автоматизации тестирования доставки новых модулей разрабатываемого проекта заинтересованным сторонам - CI/CD.

1. Jenkins

Jenkins — один из самых популярных open-source инструментов для CI/CD, который позволяет автоматизировать сборку, тестирование и развертывание программного обеспечения.

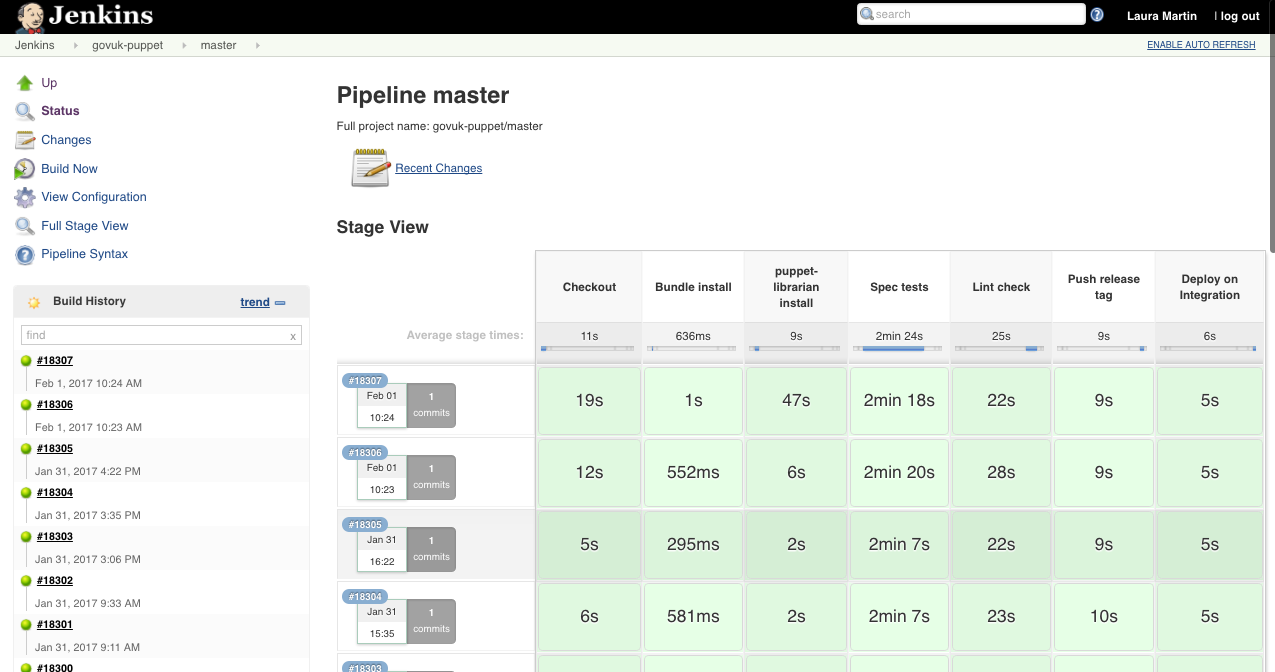


Рисунок 1 - Пример интерфейса пайплайна в Jenkins

Преимущества:

* Кросс-платформенность. Есть версии для Linux, macOS и Windows;
* Гибкость и расширяемость. Jenkins поддерживает более 1800 плагинов, что позволяет интегрировать его с практически любым инструментом DevOps;
* Система бесплатна и имеет открытый код;
* Поддержка множества языков программирования;
* Удовлетворяет высоким требованиям безопасности поддержкой self-hosted развертывания;
* Гибкость в настройке пайплайнов;
* Масштабируемость. Поддерживает распределенные сборки на разных машинах;
* Нет ограничений на количество пользователей и сборок.

Недостатки:

* Сложность в настройке и поддержке;
* Требует больших серверных ресурсов;
* Неудобный и устаревший веб-интерфейс.

1. Atlassian Bitbucket Pipelines

Это встроенный CI/CD-инструмент в Bitbucket, предназначенный для автоматизации сборки, тестирования и развертывания кода.

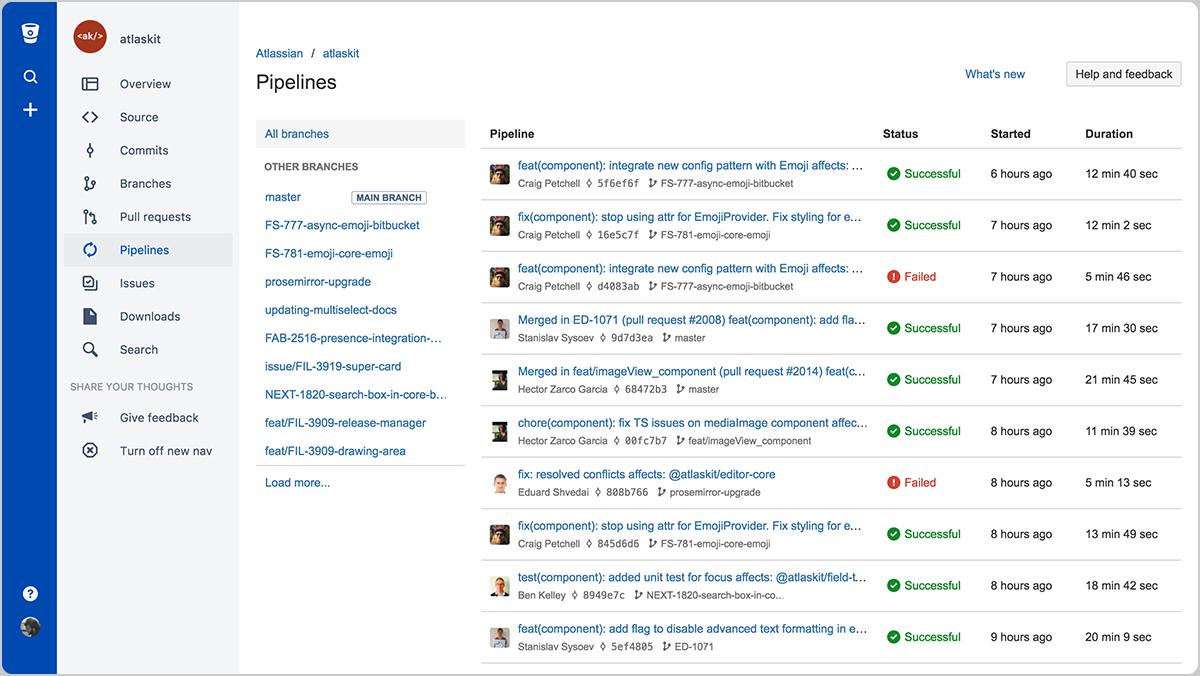


Рисунок 2 - Пример интерфейса пайплайна в Atlassian Bitbucket Pipelines

Преимущества:

* Простая конфигурация. YAML-файл с удобной настройкой этапов сборки;
* Поддержка Docker. Позволяет использовать контейнеры для изолированной сборки и тестирования;
* Автоматическое кеширование зависимостей. Ускоряет повторные сборки, снижая время выполнения CI/CD-процессов;

Недостатки:

* Недоступен на территории РФ;
* Ограниченные возможности кастомизации пайплайнов;
* Не удовлетворяет высоким требованиям безопасности, так как нет поддержки self-hosted развертывания;
* Ограничения по использованию в бесплатной версии;
* Ограниченный выбор инструментов.

1. GitLab CI/CD

Это встроенная в GitLab система CI/CD, поддерживающая автоматическую сборку, тестирование и развертывание приложений.

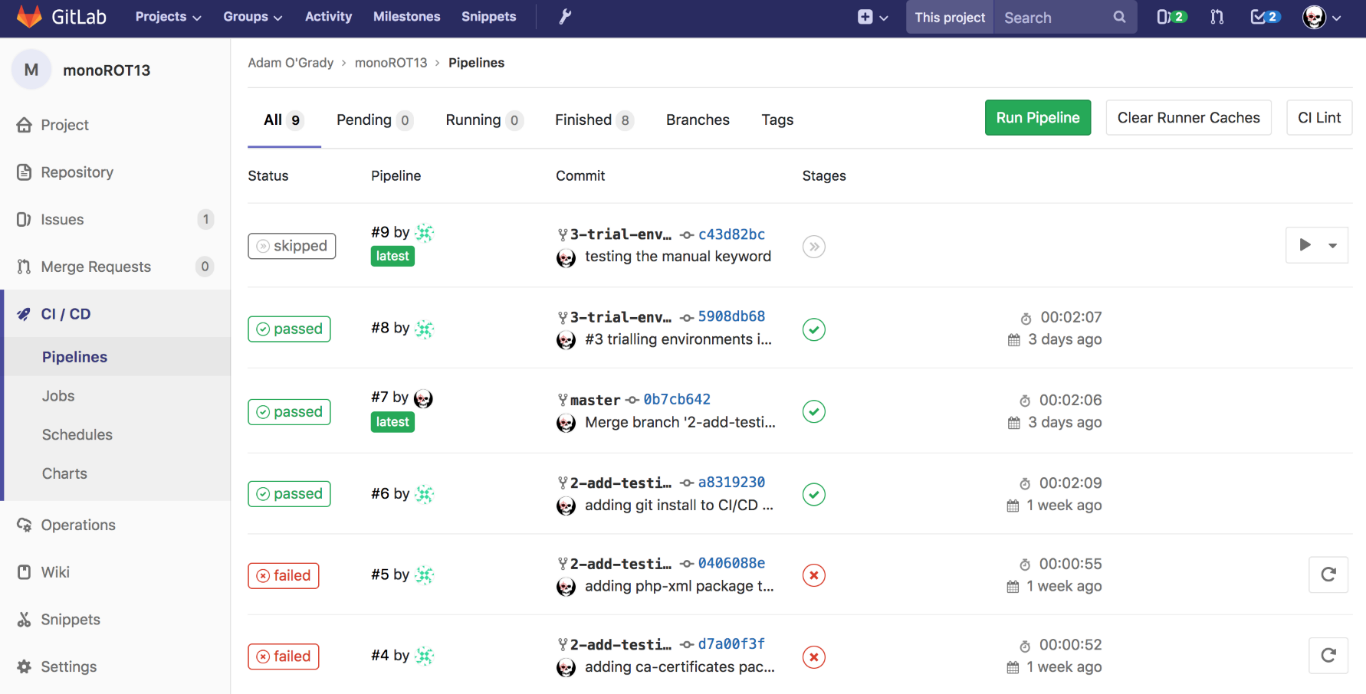


Рисунок 3 - Пример интерфейса пайплайна в GitLab CI/CD

Преимущества:

* Глубокая интеграция с GitLab;
* Гибкая настройка пайплайнов;
* Поддержка Docker и Kubernetes;
* Поддержка параллельных билдов и стадий;
* Удовлетворяет высоким требованиям безопасности поддержкой self-hosted развертывания;
* Гибкое управление правами пользователей;
* Автоматическое кеширование зависимостей. Снижает время выполнения повторных билдов;
* Развитая система мониторинга и отчетности.

Недостатки:

* Сложность в настройке;
* Ограничения бесплатной версии.

Подводя итоги, можно отметить, что технология CI/CD лежит в основе большинства современных сервисов для автоматизации сборки и выпуска релизов, и наш продукт должен соответствовать этим стандартам. Важно обеспечить высокий уровень безопасности, сравнимый с GitLab и Jenkins, а удобство и простоту работы позаимствовать у решений Atlassian. Автоматизация должна быть не только эффективной, но и быстрой, минимизируя нагрузку на серверные ресурсы. Кроме того, необходимо расширить возможности интеграции, добавив поддержку не только GitLab, но и YouTrack, что сделает наш сервис более универсальным и удобным для пользователей.

Результаты проекта

ОТЧЁТЫ О РАБОТЕ КАЖДОГО УЧАСТНИКА

Отчёт Балакина Кирилла Александровича

Отчёт Сысоевой Елены Павловны

Отчёт Тарасенко Елены Витальевны

Отчёт Хреновой Дарьи Валерьевны

РЕФЛЕКСИЯ И ВЫВОДЫ КОМАНДЫ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Автоматизация процесса релиза / [Электронный ресурс] // habr.com : [сайт]. — URL: https://habr.com/ru/companies/hh/articles/645421/ (дата обращения: 30.03.2025).

Что такое CI/CD простыми словами: CI/CD в GitLab для новичка / [Электронный ресурс] // www.reg.ru : [сайт]. — URL: https://www.reg.ru/blog/rukovodstvo-po-ci-cd-v-gitlab-dlya-novichka/?utm\_source=yandex.ru&utm\_medium=organic&utm\_campaign=yandex.ru&utm\_referrer=yandex.ru (дата обращения: 30.03.2025).

Top 5 Open Source Release Management Software / [Электронный ресурс] // www.devopsschool.com : [сайт]. — URL: https://www.devopsschool.com/blog/top-5-open-source-release-management-tool/ (дата обращения: 30.03.2025).

ПРИЛОЖЕНИЕ А