

## Курсовая работа

“Разработка анализатора проектов JIRA”

### Состав группы:

1. Костин Андрей Константинович - 5130904/20101
2. Мамонова Алина Сергеевна - 5130904/20101
3. Клеблеева Элина Маратовна - 5130904/20105

**Ссылка на репозиторий:** <https://github.com/IDevFrye/Jira-Analyzer>

**Тема проекта:** Анализатор проектов JIRA

### Описание технологического стека:

1. *Backend:* Golang
2. *Frontend:* TypeScript, React
3. *DB:* PostgreSQL
4. *Внешнее API:* Jira

(<https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro/#about>)

## Определение проблемы

Компании, использующие Jira, сталкиваются со сложностью **получения сводной статистики и аналитики** по своим проектам, особенно если речь идет о нескольких проектах одновременно. Отсутствует удобный инструмент, который позволял бы в едином интерфейсе сравнивать проекты, получать детальную аналитику по времени выполнения задач, динамике создания и закрытия тикетов, распределению задач по статусам, приоритетам и т.д. Для проектных офисов и аналитиков критично иметь такую информацию в удобном виде, чтобы принимать решения по управлению командами и процессами.

## Цель

Создать клиент-серверное приложение с микросервисной архитектурой для автоматизированного анализа задач из репозитория Jira компании Apache с сохранением данных в PostgreSQL и предоставлением удобного веб-интерфейса для получения статистики, аналитики и сравнения проектов.

# Требования

## Пользовательские истории

Пользовательские истории	
<b>User Story 1</b> (Администратор)	Как администратор, я хочу иметь возможность настроить список отслеживаемых Jira проектов, чтобы приложение собирало данные автоматически с заданной периодичностью.
<b>User Story 2</b> (Менеджер проекта)	Как менеджер проекта, я хочу получить сводную статистику по своему проекту, чтобы быстро оценить состояние задач, динамику их выполнения и общую загрузку команды.
<b>User Story 3</b> (Руководитель департамента)	Как руководитель департамента, я хочу иметь возможность сравнить несколько проектов по ключевым метрикам, чтобы выявить успешные практики и зоны для улучшений в управлении задачами.

## Функциональные требования

Функциональные требования к блоку “Аналитика”	
ФТ.1	Страница “Мои проекты” содержит сухую статистику задач по каждому из проектов
ФТ.1.1	Статистика должна содержать: <ul style="list-style-type: none"><li>● Общее кол-во задач</li><li>● Кол-во открытых задач</li><li>● Кол-во закрытых задач</li><li>● Среднее время выполнения задачи (часы)</li><li>● Среднее кол-во заведенных задач в день за последнюю неделю</li></ul>

<b>Функциональные требования к блоку “Аналитика”</b>	
ФТ.2	У каждого проекта должна быть возможность выбора аналитических задач
ФТ.2.1	Аналитические задачи: <ul style="list-style-type: none"> <li>• Гистограмма, отражающая время, которое задачи провели в открытом состоянии</li> <li>• Диаграммы, которые показывают распределение времени по состоянием задач</li> <li>• График активности по задачам</li> <li>• График сложности задач</li> <li>• График, отражающий приоритетность всех задач</li> <li>• График, отражающий приоритетность закрытых задач</li> </ul>

### Детализация ФТ

№	Требование / Функция	Описание / Комментарий
<b>1</b>	<b>Получение сухой статистики по проектам Jira</b>	<b>Основная статистическая информация по каждому проекту, обновляемая при каждом запуске ETL</b>
1.1	Общее количество задач	Количество всех задач в проекте, вне зависимости от статуса
1.2	Количество открытых задач	Все задачи, которые не имеют конечного статуса (Closed, Resolved и т.п.)
1.3	Количество закрытых задач	Все задачи со статусом Closed
1.4	Количество переоткрытых задач	Задачи, у которых есть переходы в статус Reopened
1.5	Количество разрешенных задач	Все задачи со статусом Resolved
1.6	Количество задач со статусом "In progress"	Текущие задачи, над которыми идет работа

1.7	Среднее время выполнения задачи (часы)	Разница между датой создания и закрытия задачи (только для закрытых задач)
1.8	Среднее количество заведенных задач в день за последнюю неделю	Количество новых задач за 7 дней, деленное на 7
<b>2</b>	<b>Возможность аналитики проектов</b>	<b>Построение диаграмм и графиков по каждому проекту</b>
2.1	Гистограмма времени в открытом состоянии	Сколько времени задачи находились в открытом статусе (по закрытым задачам)
2.2	Диаграммы по времени для каждого статуса	Время, которое задачи провели в каждом статусе (Open, In Progress, Resolved и т.д.)
2.3	Гистограмма по залогированному времени	Сколько времени пользователи логировали в задачи (по закрытым задачам)
2.4	График по приоритетам	Распределение задач по приоритетам (Blocker, Critical, Major, Minor и т.п.)
<b>3</b>	<b>Сравнение проектов</b>	<b>Инструмент для сравнения нескольких проектов между собой</b>
3.1	Сводная таблица сухой статистики	Таблица с основными метриками (пункты 1.1-1.7) по каждому проекту для сравнения
3.2	Сводные графики по аналитике	Графики и диаграммы (пункты 2.1-2.4) по каждому проекту, наложенные для сравнения

## Оценка характеристик

Оценка характеристик использования ресурсов Jira	
Количество пользователей	10k активных пользователей в сутки
Среднее количество Jira проектов на одного пользователя	3-5
Среднее количество задач на проект	500-1000
Глубина хранения	5 лет исторических данных
Среднее обновление данных	каждые 5 минут (конфигурируемо)

## Затраты и окупаемость

Оценка серверных затрат	Окупаемость
Сервер для разворота веб-приложения (хостинг):	36.000 руб. (за 1 год)
Домен веб-сайта + SSL-сертификат:	2.000 руб. (единожды)
Итоговая оценка инфраструктуры	~38.000 руб./год.

- Потенциально окупается за счет сокращения времени на ручную аналитику и отчетность в 5-10 раз.
- Оценка экономии трудозатрат: ~20 часов в месяц на каждого менеджера (при средней ставке 600 руб./час (100.000 руб./месяц) — экономия ~12.000 руб./месяц).

**Окупается, в среднем, за 4 месяца.**

## НФТ (Нефункциональные требования)

Показатель	Значение	Пояснение выбора
<b>Throughput (RW QPS)</b>	1000 RPS (чтение), 100 RPS (запись)	Основная нагрузка — аналитические дашборды, которые запрашиваются часто, особенно при мониторинге. Запись идет батчами при ETL.
<b>Latency (RW)</b>	чтение < 200 мс, запись < 500 мс	Пользователи ожидают быструю загрузку графиков и метрик. Обновления могут выполняться чуть медленнее.
<b>R/W Ratio</b>	90% чтение, 10% запись	Аналитика — это в первую очередь чтение данных. Обновление данных (ETL) происходит относительно редко.
<b>Traffic Volume (RW)</b>	до 10 ГБ в сутки	Основная нагрузка — запросы аналитики, трафик ETL минимален (все данные текстовые). Средний размер одной задачи (JSON от Jira API): <b>2-5 КБ</b> — зависит от количества полей, комментариев, истории изменений. JSON статистики по проекту: <b>~50-100 КБ</b> (графики, таблицы, мета)
<b>Storage (DB)</b>	100 ГБ за 5 лет	Оценка объема с запасом на историю задач, статусы, переходы, комментарии и мета-информацию.

		Средний размер задачи в БД (плоский вид + история) = <b>20-50 КБ</b> (JSON, связи, индексы)
<b>Cache</b>	Данные статистики можно кэшировать на уровне БД (MV), клиента (React.useMemo/useCallback) и на уровне аналитического сервиса (Redis).	Максимально сокращаем время отклика за счет предобработки и кэширования.
<b>CAP</b>	Strong consistency внутри одного проекта, Eventual consistency при межпроектных расчетах. Availability в Табл.1	Внутри проекта — строгая консистентность (статистика должна совпадать везде). При глобальных сводках по нескольким проектам — допускаем задержку.
<b>Performance vs Scalability</b>	приоритет на масштабируемость по числу проектов и задач. 1 микросервис = ~1 проект	Важно поддерживать возможность добавления новых проектов без деградации производительности.
<b>Costs</b>	~38.000 руб./год	Расчет выше



## Availability

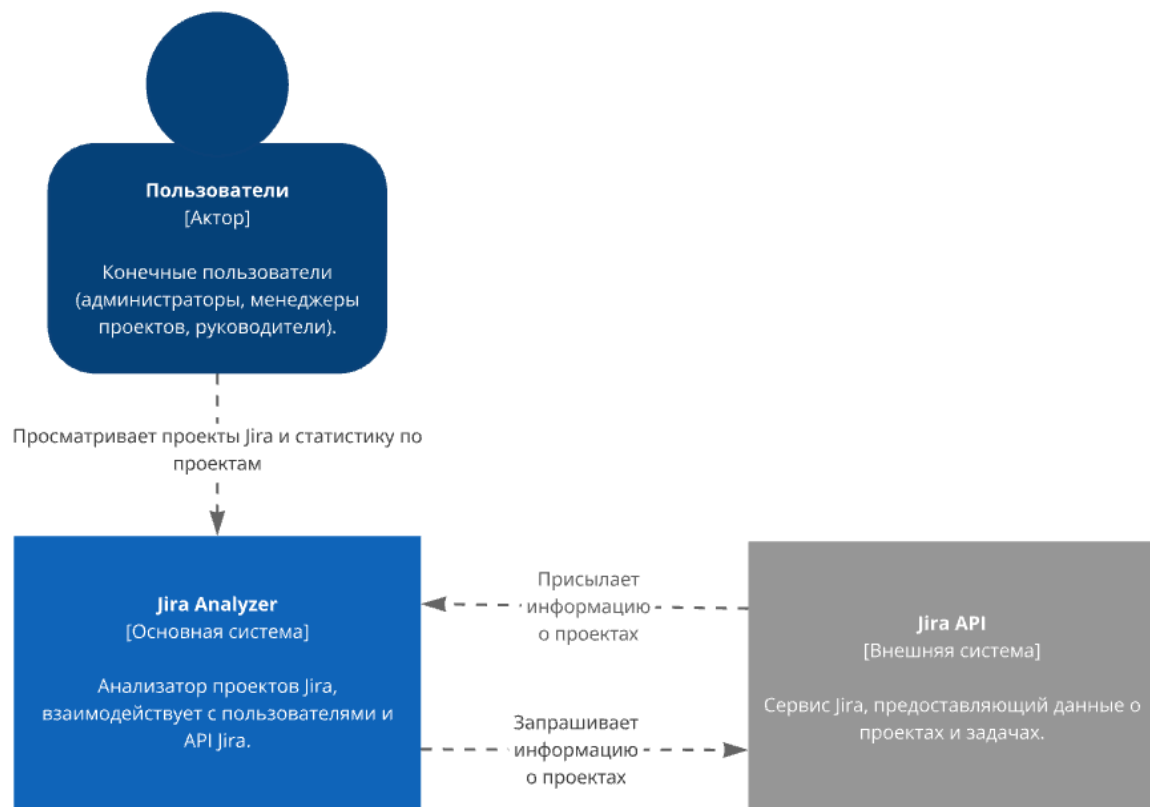
Подсистема	Доступность (SLA)	Комментарий
Backend (API)	99.5%	Высокая критичность для работы фронтенда. Простой приводит к полной недоступности аналитики для пользователей.
Frontend (UI)	99%	Относительно низкая критичность, допустимы краткосрочные простои, так как данные не теряются, а интерфейс может восстановиться при перезагрузке.
ETL (сбор данных из Jira)	95%	Низкая критичность к доступности. Простой влияет только на актуальность данных, но не на работу фронтенда или сохранность информации.
БД (Postgres)	99.9%	Критичный компонент с высокими требованиями по доступности. Долгий простой приводит к потере функциональности всей системы. Рекомендуется кластерная конфигурация с резервированием и автоматическим failover.

# Верхнеуровневая архитектура

## Характер нагрузки на сервис

1. Соотношение R/W нагрузки:
  - Основной упор идет на чтение (Read) данных, так как аналитические задачи требуют агрегации и отчетности. Ожидаемое соотношение R/W = 90/10.
  - Записи (Write) происходят при создании, обновлении задач, смене их статусов.
2. Объемы трафика:
  - до 10 ГБ в сутки
  - Основная нагрузка — запросы аналитики, трафик ETL минимален (все данные текстовые).
  - Средний размер одной задачи (JSON от Jira API): **2-5 КБ** — зависит от количества полей, комментариев, истории изменений.  
JSON статистики по проекту: **~50-100 КБ** (графики, таблицы, мета)
3. Объемы дисковой системы:
  - Ожидаемый объем хранения в БД: 100 ГБ за 5 лет (Средний размер задачи в БД (плоский вид + история) = **20-50 КБ** (JSON, связи, индексы)
  - Логирование событий: до 20 ГБ в год.

## High Level Design (HLD + C4)



**Анализатор проектов Jira**  
Контекстная диаграмма C4  
Последнее изменение: 11.03.2025

Рис.1 Контекстная диаграмма C4

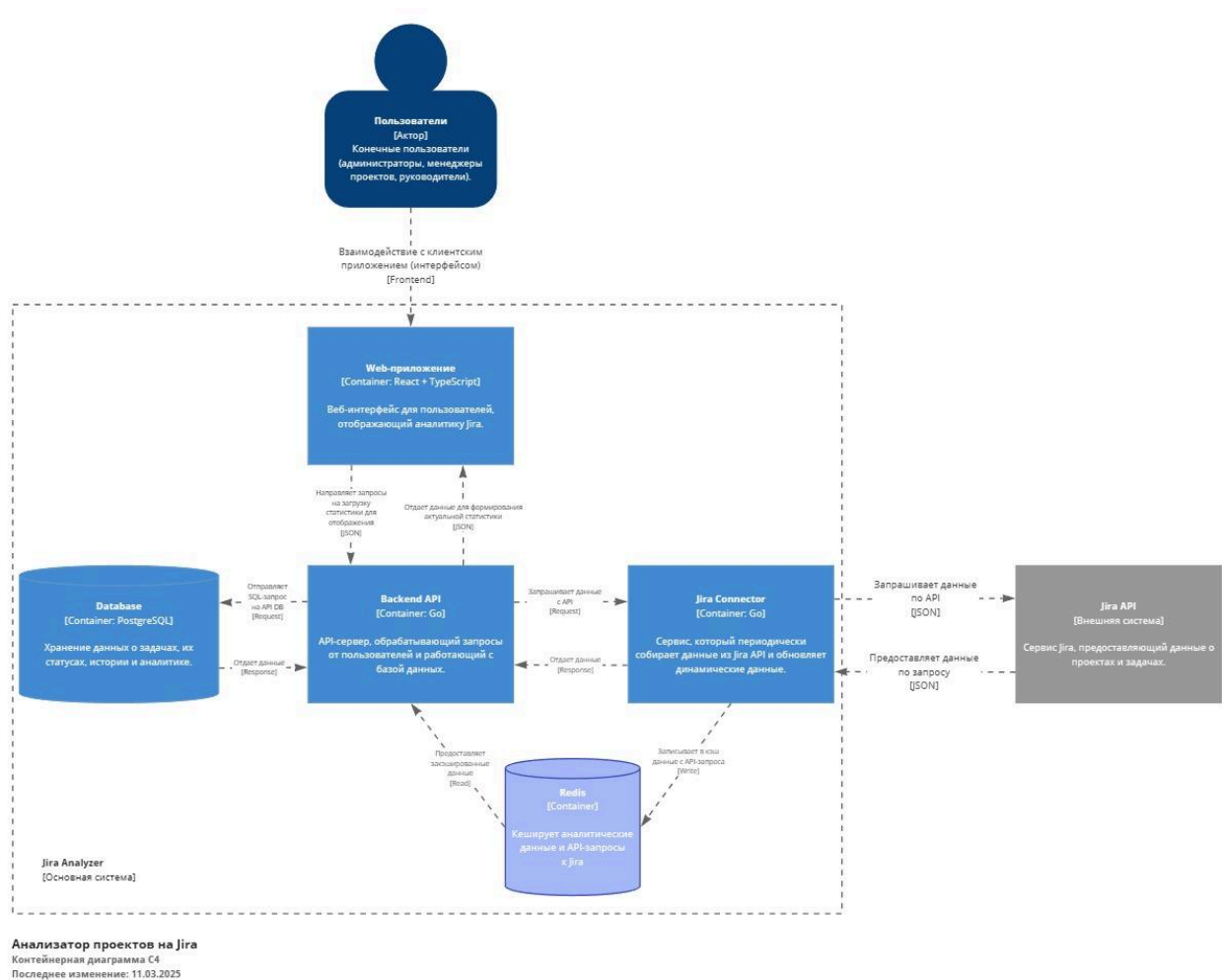


Рис.2 Контейнерная диаграмма C4

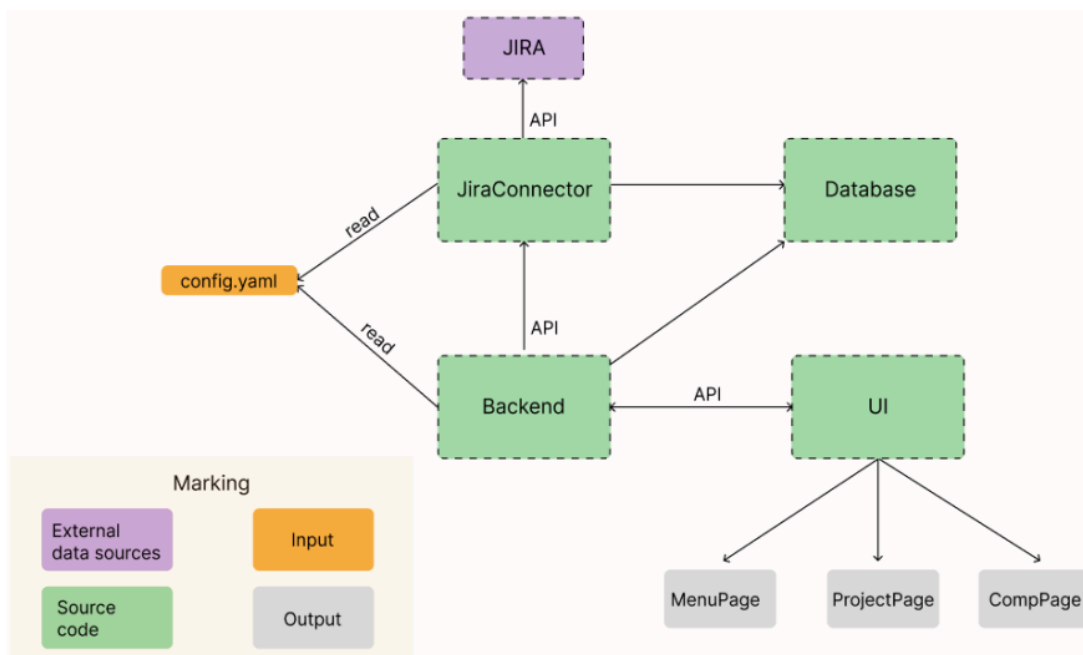


Рис 3. Высокоуровневый дизайн приложения "Jira analyzer"

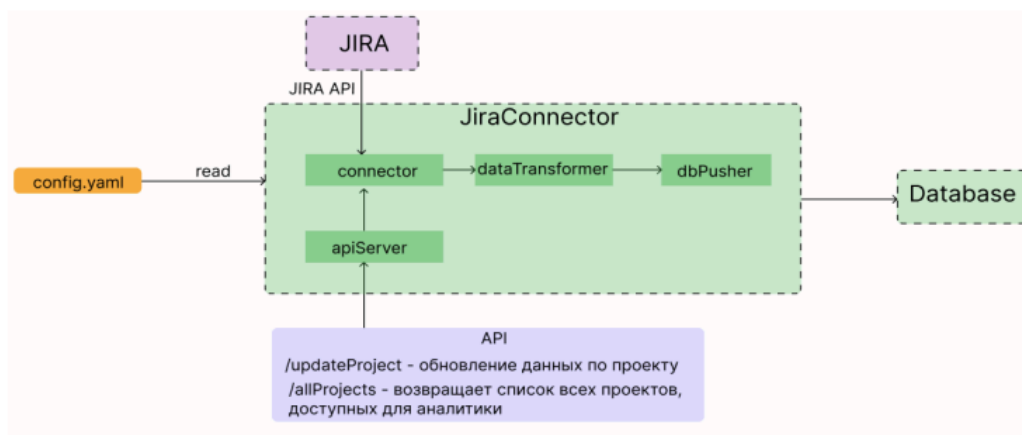


Рис 4. Высокоуровневый дизайн сервиса “Jira connector”

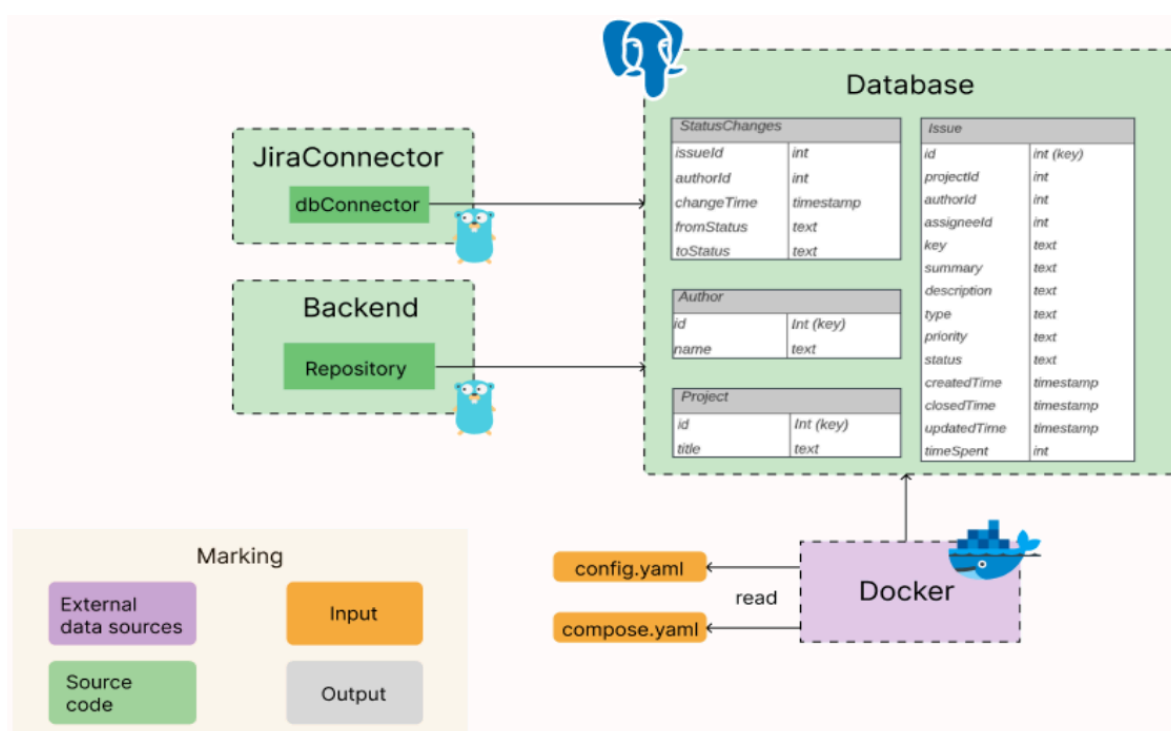


Рис. 5. Высокоуровневый дизайн базы данных

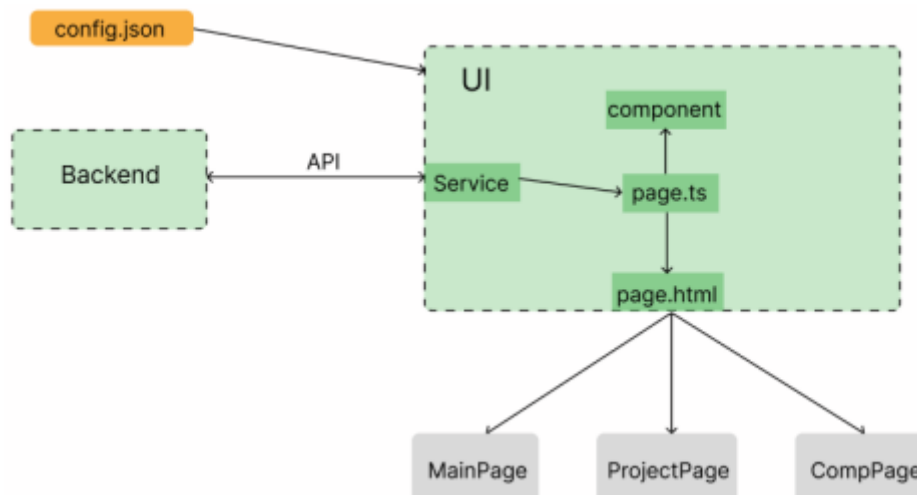


Рис. 6. Высокоуровневая архитектура модуля, реализующего пользовательский Web-интерфейс

## Контракты API (Swagger)

Подробное описание эндпоинтов можно рассмотреть в [Приложении 1](#).

### JiraAnalyzer API 1.0.0 OAS 3.0

API для взаимодействия с Jira и получения аналитики проектов

Servers	
https://api.jiraanalyzer.ru - Продакшн сервер	
<b>Connector API</b>	
POST	/updateProject Обновить или загрузить проект из Jira
GET	/projects Получить список проектов
<b>Backend API</b>	
GET	/api/v1/projects Получить список загруженных проектов
GET	/api/v1/projects/{id} Получить статистику проекта
DELETE	/api/v1/projects/{id} Удалить проект из БД
GET	/api/v1/connector/projects Получить список доступных проектов из Jira
POST	/api/v1/connector/updateProject Обновить или загрузить проект по ключу
GET	/api/v1/graph/get/{taskNumber} Получить данные по аналитической задаче
POST	/api/v1/graph/make/{taskNumber} Проведение аналитической задачи
DELETE	/api/v1/graph/delete Удаление всех аналитических задач для проекта
GET	/api/v1/isAnalyzed Проверка наличия аналитики для проекта
GET	/api/v1/compare/{taskNumber} Получение данных по аналитической задаче для нескольких проектов

Рис. 7. Контракты API (Swagger)

## Ожидаемые нефункциональные требования на время отклика

### 1. API:

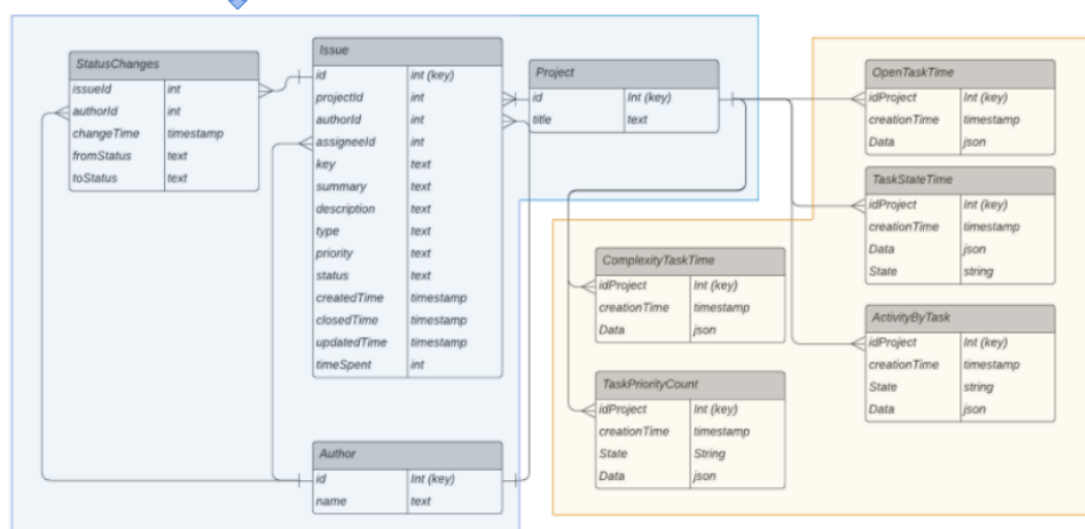
- Чтение данных (GET-запросы) — до 100 мс.
- Запись данных (POST/PUT) — до 250 мс.
- Массовые запросы аналитики — до 500 мс.

### 2. База данных:

- SELECT по индексированным полям — <50 мс.
- INSERT/UPDATE — <150 мс.
- JOIN с агрегацией — <400 мс.

## Схема базы данных

Основные данные



Хранение аналитических задач

Рис. 8. Схема базы данных для приложения "Jira analyzer"

## Почему схема БД выдержит нефункциональные требования?

### 1. Индексация:

- Использование индексов по `id`, `projectId`, `status`, `createdTime` для быстрого поиска и агрегации.
- JSON-поля (для аналитических данных) не индексируются, что ускоряет вставку.

### 2. Партиционирование:

- Таблица **Issue** разделяется по `projectId`, чтобы уменьшить нагрузку на запросы.
- Таблица **StatusChanges** партиционируется по `changeTime` (месячные сегменты).

### 3. Кэширование:

- Использование Redis для хранения часто запрашиваемых данных (Issue, Project).
- Возможность реплицирования READ-запросов на реплики базы данных.

## Схема масштабирования сервиса

### При росте нагрузки в 10 раз:

#### 1. Горизонтальное масштабирование базы данных:

- Разделение нагрузки с помощью шардинга по `projectId`.
- Использование `read-replicas` для масштабирования чтения.
- Ожидаемая конфигурация:
  - Было: 1 мастер + 2 реплики (по 1000 RPS)
  - Стало: 1 мастер + 9 реплик (по 10 000 RPS)

#### 2. Балансировка нагрузки на API:

- Nginx + Load Balancer для распределения запросов.
- Увеличение количества инстансов сервисов через Kubernetes.
- Ожидаемая конфигурация:
  - Было: 3 API-инстанса (по 1000 RPS каждый)
  - Стало: 30 API-инстансов (по 1000 RPS каждый)

#### 3. Обработка фоновых задач:

- Очереди сообщений (Kafka) для обработки долгих задач (аналитика, отчеты).
- Batch-обработка данных для вычислений аналитических отчетов.
  - Было: 1 воркер-пул (5 потоков)
  - Стало: 10 воркер-пулов (по 5 потоков каждый)

#### 4. Кэширование и CDN:

- Расширение слоя кэша (Redis) для уменьшения нагрузки на базу.
- Использование CDN (Cloudflare) для уменьшения задержек API-запросов.
  - Было: 5 ГБ кэша (Redis)
  - Стало: 50 ГБ кэша (Redis)



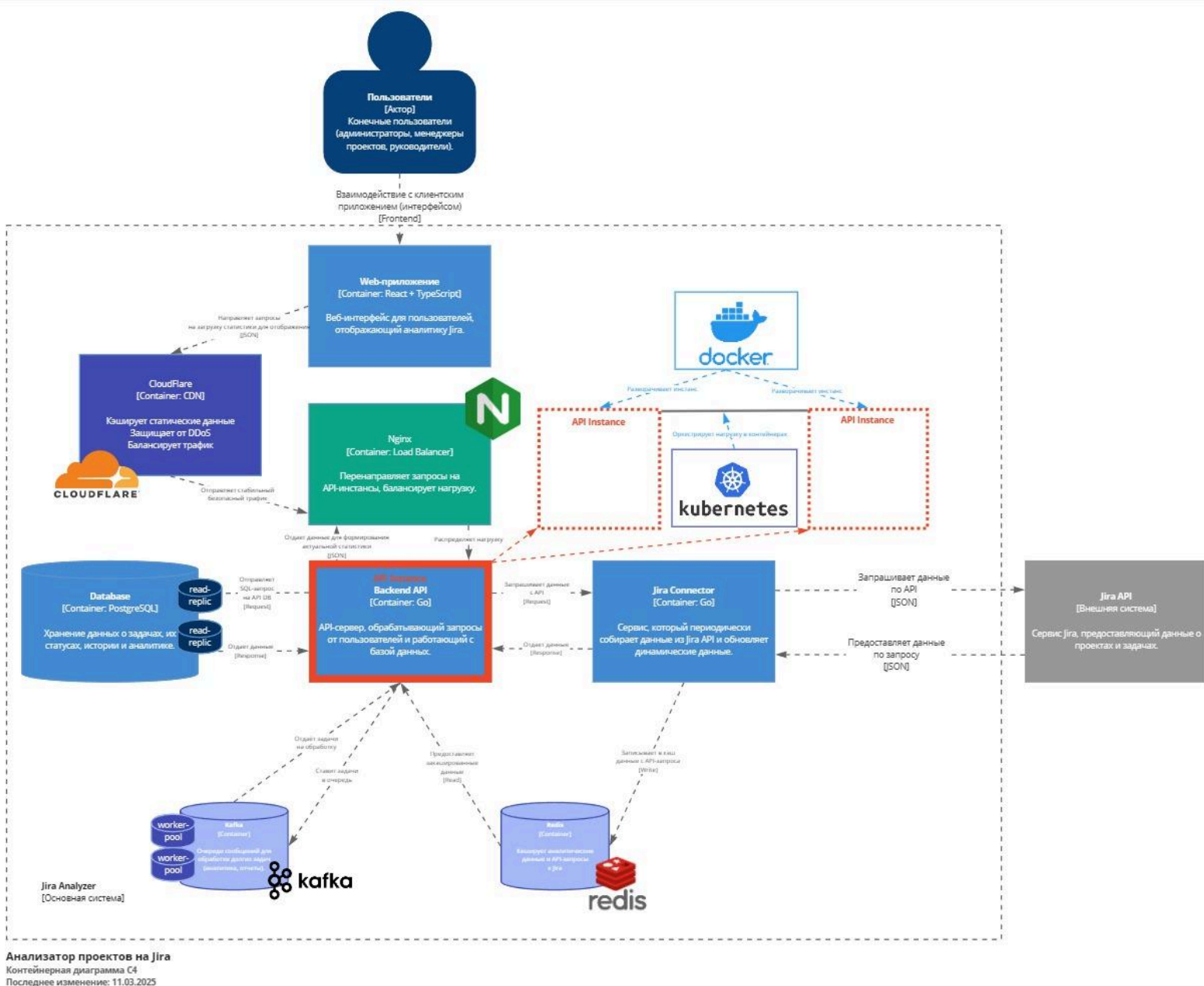


Рис. 9. C4 диаграмма масштабирования сервиса

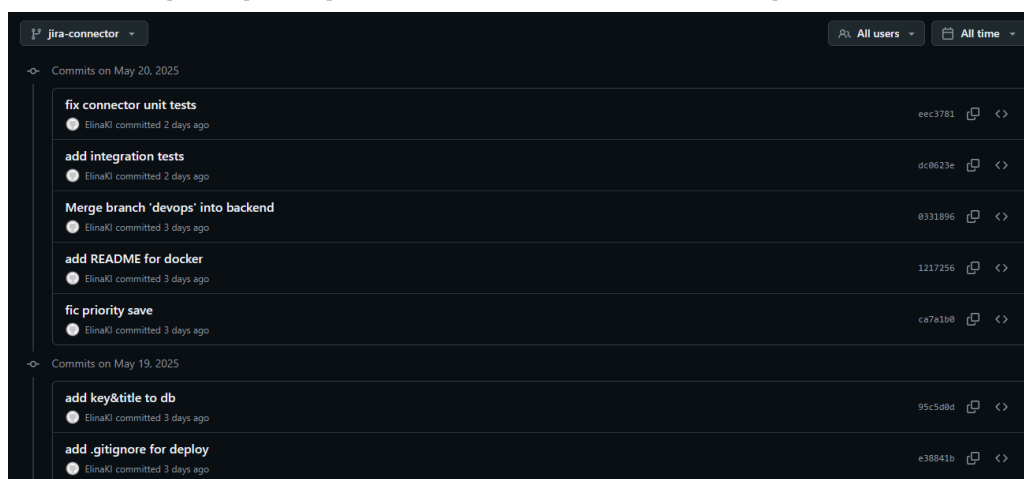
# Кодирование и отладка

## Разработка велась в ветках:

- [main](#) - основная ветка с релизом
- [develop-front](#) - ветка разработки клиентской части
- [backend](#) - ветка разработки серверной части (микросервис бэкенда, предоставляющий API фронтенду и обращающийся к БД и микросервису-коннектору)
- [jira-connector](#) - ветка разработки серверной части (микросервис бэкенда, предоставляющий API бэкенду и обращающийся к БД и внешнему API Jira)
- [devops](#) - ветка разработки конфигов и файлов для сборки, тестирования и развертывания проекта

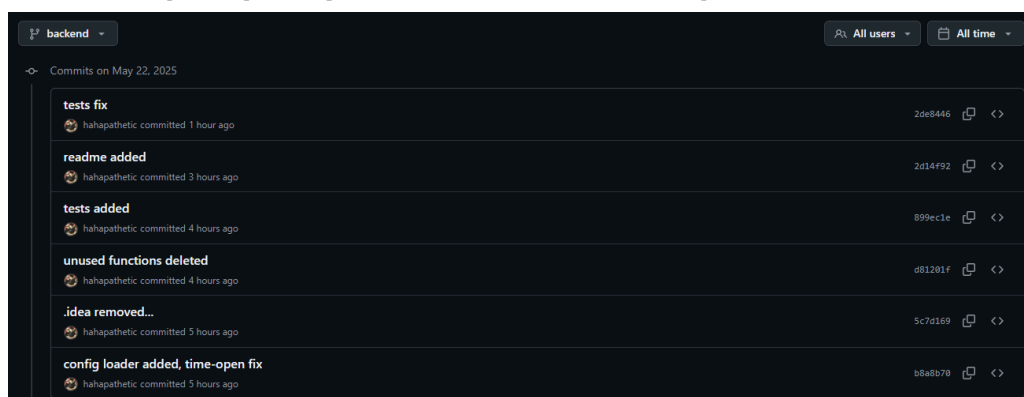
Каждый участник сделал коммиты в данный проект. Подробнее - см. историю коммитов данного репозитория (по всем веткам).

## Backend (микросервис от внешнего API Jira):



jira-connector		All users	All time
Commits on May 20, 2025			
fix connector unit tests	ElinaKi committed 2 days ago	eec3781	
add integration tests	ElinaKi committed 2 days ago	dc0623e	
Merge branch 'devops' into backend	ElinaKi committed 3 days ago	0331896	
add README for docker	ElinaKi committed 3 days ago	1217256	
fix priority save	ElinaKi committed 3 days ago	ca7a1b0	
Commits on May 19, 2025			
add key&title to db	ElinaKi committed 3 days ago	95c5dbd	
add .gitignore for deploy	ElinaKi committed 3 days ago	e38841b	

## Backend (микросервис API для клиента):



backend		All users	All time
Commits on May 22, 2025			
tests fix	hahapathetic committed 1 hour ago	2de8446	
readme added	hahapathetic committed 3 hours ago	2d14f92	
tests added	hahapathetic committed 4 hours ago	899ec1e	
unused functions deleted	hahapathetic committed 4 hours ago	d81201f	
.idea removed...	hahapathetic committed 5 hours ago	5c7d169	
config loader added, time-open fix	hahapathetic committed 5 hours ago	b8a8b70	

## Frontend:

develop-front	All users	All time
Commits on May 23, 2025		
[Frontend][Technical] Update README	Verified 2a9d7e6	<>
[Frontend][Bugfix] Update key param for backend API + Charts view	e9eeF27	<>
Commits on May 20, 2025		
[Frontend][Bugfix] Delete prj from db + Charts colors + backend data model change	80f7579	<>
Commits on May 17, 2025		
[Frontend][Bugfix] Fix delete issues + pagination + wrap	a08e1b4	<>
[Frontend][Integr] Integration with Proxy Server	a3fe48b	<>
Commits on May 15, 2025		
[Frontend][Feat] Charts added. All pages redesigned for API	71ab008	<>
Commits on May 14, 2025		
[Frontend][Feat] Basic components & pages added	82795d7	<>
Commits on May 13, 2025		
[Frontend] Init React+TS JA App	a375000	<>

## Devops (Docker, Nginx):

devops	All users	All time
Commits on May 20, 2025		
add README for docker	1217256	<>
Commits on May 19, 2025		
add .gitignore for deploy	e38841b	<>
Commits on May 17, 2025		
devops v1	ce0a162	<>
Merge branch 'develop-front' into devops	0a31e63	<>
[Frontend][Bugfix] Fix delete issues + pagination + wrap	a08e1b4	<>
Merge branch 'develop-front' into devops	cc55490	<>
[Frontend][Integr] Integration with Proxy Server	a3fe48b	<>
Merge branch 'develop-front' into devops	d5ba215	<>
rebase .gitignore	66190dc	<>

# Unit тестирование

## Покрытие UNIT тестами по функциональным модулям

Модуль	Покрытие
JiraConnector/jiraHandlers	100.0% of statements
JiraConnector/jiraService	100.0% of statements
JiraConnector/connector	90.8% of statements
JiraConnector/dataTransformer	100.0% of statements
JiraConnector/dbPusher	90.2% of statements
endpointHandler/compare	81.3% of statements
endpointHandler/handler	62.5% of statements
endpointHandler/repository	72.7% of statements
endpointHandler/service	81.5% of statements
endpointHandler/analytics	80.0% of statements

**Итого:** 80-100% покрытие ключевых модулей, что является достаточным для определения функциональных модулей корректными.

# Интеграционное тестирование

## Тесты интеграций отдельных микросервисов:

Интеграция	Сценарий	Время
<b>Jira Connector Server</b>		
<b>[JiraService - DB]</b>	updateProject (new)	-- PASS: TestLoadNewProject (0.04s)
<b>[JiraService - DB]</b>	updateProject (Exists)	--- PASS: TestUpdateExistingProject (0.05s)
<b>[JiraService - DB]</b>	updateProject (Dup)	--- PASS: TestProjectEdgeCases (0.02s)
<b>[JiraConnector - Jira API]</b>	GET /project (All)	--- PASS: TestGetAllProjects (0.004s)
<b>[JiraConnector - Jira API]</b>	GET /project?page=N	--- PASS: TestGetProjectsPage/First_page (0.001s)  --- PASS: TestGetProjectsPage/Search_filtered (0.001s)  --- PASS: TestGetProjectsPage/Empty_page (0.001s)  --- PASS: TestGetProjectsPage (0.001s)
<b>[JiraConnector - Jira API]</b>	GET /project (Invalid)	--- PASS: TestErrorHandling/Jira_unavailable (0.03s)  --- PASS: TestErrorHandling/Rate_limiting (0.00s)

		<p>--- PASS: TestErrorHandling/Invalid_JSON_res ponse (0.00s)</p> <p>--- PASS: TestErrorHandling (0.03s)</p>
<b>[Full Jira Connector ServerWorkflow ]</b>	<p>GET /projects -&gt;</p> <p>POST /updateProject? project=NAME -&gt;</p> <p>POST /updateProject? project=UNKN OWN</p>	<p>--- PASS: TestFullIntegration/Get_projects_list (0.00s)</p> <p>--- PASS: TestFullIntegration/Load_project_iss ues (0.12s)</p> <p>--- PASS: TestFullIntegration/Error_cases (0.00s)</p> <p>--- PASS: TestFullIntegration (0.12s)</p>
<b>Backend Server</b>		
<b>[Backend - Jira Connector]</b>	GET connector/proje cts	--- PASS: TestJiraConnectorProjects (0.21s)
<b>[Backend - Jira Connector]</b>	POST connector/upda teProject	--- PASS: TestJiraConnectorUpdate (0.45s)
<b>[Backend - DB]</b>	GetAllProjects	--- PASS: TestDBGetAllProjects (4.81s)
<b>[Backend - DB]</b>	GetStats	--- PASS: TestDBGetStats (0.63s)
<b>[Backend - DB]</b>	DeleteProject	--- PASS: TestDBDeleteProject (0.63s)

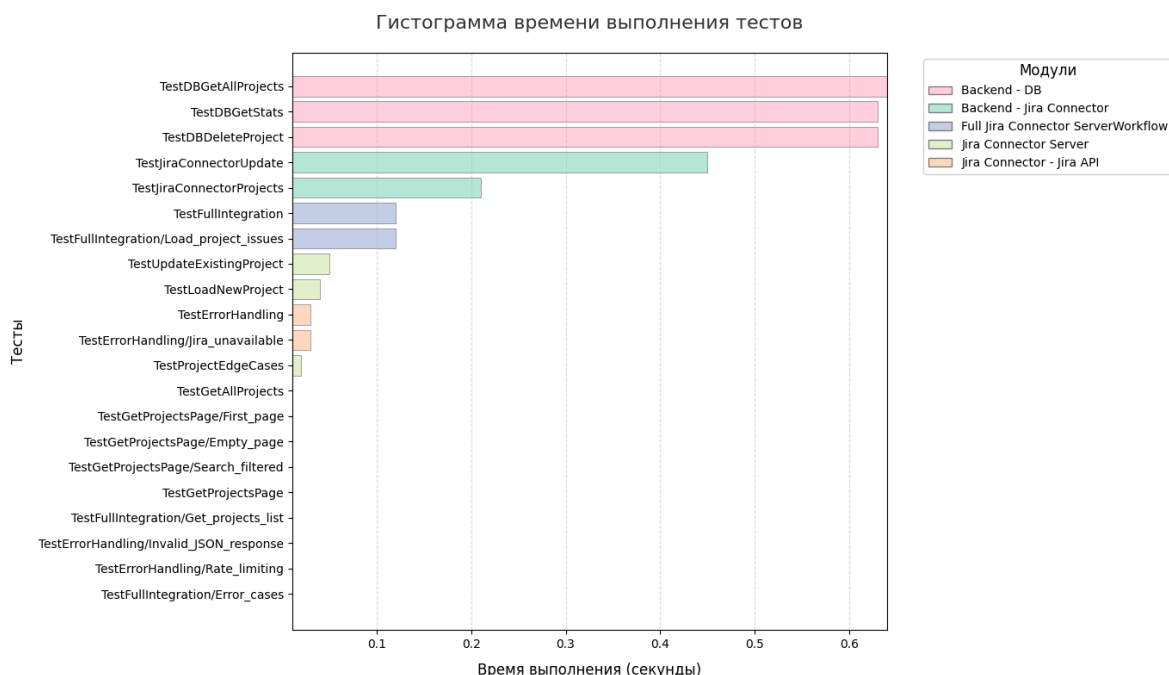


Рис. 10. Гистограмма времени выполнения интеграционных тестов

### Полный пользовательский сценарий:

#### Сценарий 1:

Просмотр всех доступных проектов -> Добавление себе проекта -> просмотр аналитики по добавленному проекту -> Удаление проекта

<b>[Backend - Jira Connector - DB]</b>	<b>--- PASS:</b> <b>TestFullScenarioFirst</b> <b>(0.62s)</b>
[Backend -> Jira Connector] GET /connector/projects?page=1&limit=9 -> получены 9 проектов из Jira	--- PASS: TestFullScenarioFirst/ GET_/connector/projects (0.23s)
[Backend -> JiraConnector -> DB] POST /connector/updateProject?project= KEY -> загружен один выбранный проект в базу данных	--- PASS: TestFullScenarioFirst/ POST_/connector/updateProject (0.38s)
[Backend -> DB] GET/projects	--- PASS: TestFullScenarioFirst/ GET_/projects

-> загружен 1 проект с тем же именем и ключом	(0.00s)
[Backend -> DB] GET /projects/ID -> получение всей информации о проекте из базы данных	--- PASS: TestFullScenarioFirst/ GET_/projects/{:id} (0.00s)
[Backend -> DB] GET /analytics/status-distribution?key=KEY -> информация о соотношении открытых\закрытых задач не пуста	--- PASS: TestFullScenarioFirst/ GET_/analytics/status-distribution (0.00s)
[Backend -> DB] DELETE /projects/ID -> удаление выбранного проекта по id из базы данных	--- PASS: TestFullScenarioFirst/ DELETE_/deleteProject (0.00s)
<b>Сценарий 2:</b> Просмотр всех доступных проектов -> Добавление себе двух проектов -> просмотр сравнительного анализа по проектам	
<b>[Backend - Jira Connector - DB]</b>	<b>--- PASS:</b> <b>TestFullscenarioSecond</b> <b>(1.95s)</b>
[Backend -> Jira Connector] GET /connector/projects -> получены проекты из Jira	--- PASS: TestFullscenarioSecond/ GET_/connector/projects (0.31s)
[Backend -> JiraConnector -> DB] POST /connector/updateProject?project=KEY -> загружен один выбранный проект в базу данных	--- PASS: TestFullscenarioSecond/ POST_/connector/updateProject_[two] (1.64s)
[Backend -> JiraConnector -> DB] POST	--- PASS: TestFullscenarioSecond/



/connector/updateProject?project=KEY -> загружен второй выбранный проект в базу данных	GET_/projects (0.00s)
[Backend -> DB] GET /projects -> загружено 2 проекта с теми же именами	--- PASS: TestFullscenarioSecond/ GET_/compare/priority (0.00s)
[Backend -> DB] GET /compare/priority?key=KEY1,KEY2 -> получение сравнительных данных по двум ранее загруженным задачам	--- PASS: TestFullscenarioSecond/ DELETE_/deleteProject (0.00s)

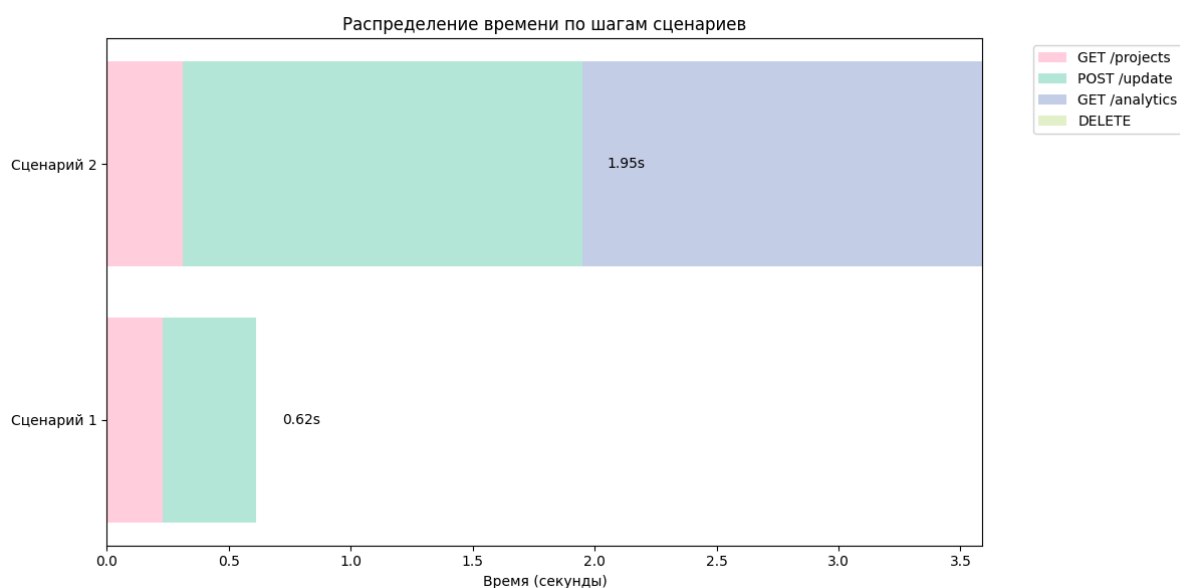


Рис. 11. График распределения времени по шагам пользовательских сценариев

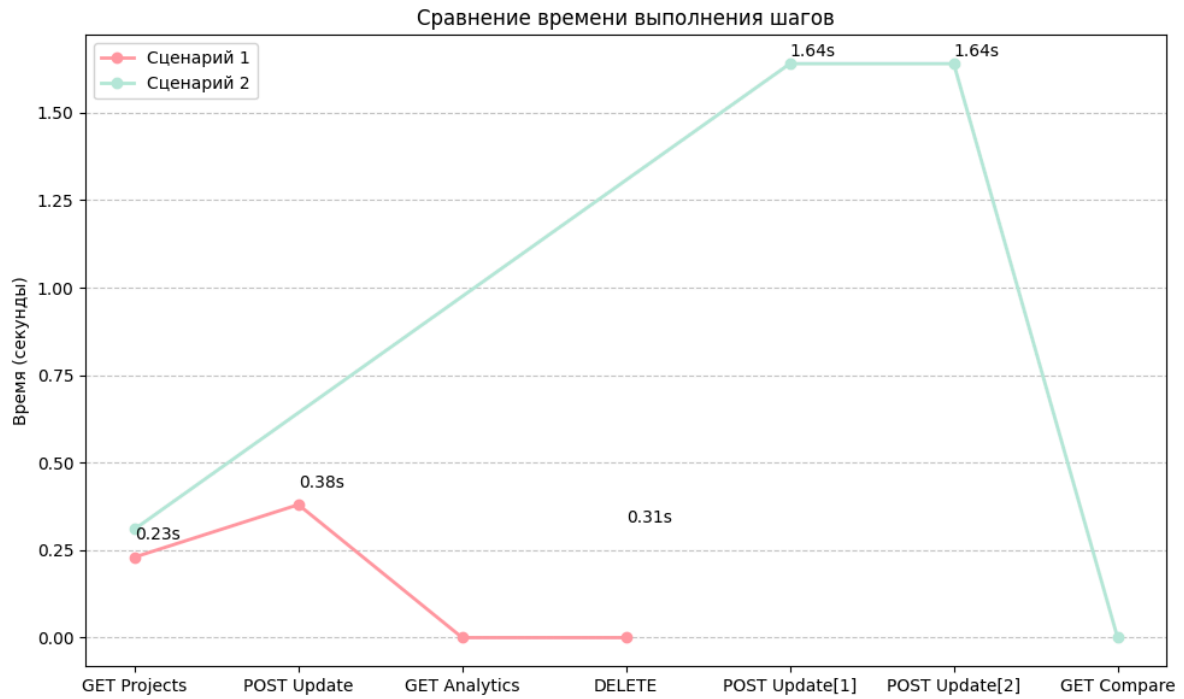


Рис. 12. График сравнения времени выполнения разных пользовательских сценариев

# Нагрузочное тестирование

## Цели нагрузочного тестирования

- Как сервис справляется с типичной и пиковой нагрузкой.
- Какова задержка отклика API при различных объемах R/W запросов.
- Как система масштабируется с ростом проектов и задач.
- Устойчивость при долгой нагрузке (Soak Test).
- Поведение при резких пиках (Stress Test)

## Основные метрики:

Метрика	Цель	Инструмент
<b>Throughput (RPS/QPS)</b>	Проверка соответствия 1000 RPS (чтение) и 100 RPS (запись)	xk6 + InfluxDB
<b>Latency (p95, p99)</b>	Проверка соответствия: чтение < 200 мс, запись < 500 мс	xk6 + Grafana
<b>Error rate</b>	Обнаружение сбоев или деградации под нагрузкой	xk6 + Grafana
<b>R/W соотношение</b>	Проверка соответствия 90/10	xk6 логика

## Сценарии нагрузочного тестирования

### Сценарий 1: Основная нагрузка (Read-heavy)

- 90% — GET запросы (графики, таблицы, сводки).
- 10% — POST/PUT запросы (обновление данных).
- Постепенное увеличение нагрузки до 1000 RPS (чтение) и 100 RPS (запись).
- Продолжительность: 15–30 мин.



Рис. 13. График динамики полученных данных и пиков виртуальных пользователей для read-heavy тестов



Рис. 14. График динамики отправленных данных и активности виртуальных пользователей для read-heavy тестов

Параметр	Значение	Оценка
<b>Длительность</b>	15м30с	задано
<b>Запросов/сек</b>	~428 итераций/сек	Цель — 1000 RPS чтение
<b>Max VUs</b>	400 (достигнут потолок)	Лимит VUs не дал выйти на нужную нагрузку
<b>Ошибки</b>	присутствуют, timeouts	Недостаток ресурсов
<b>Dropped iters</b>	да (см. warning)	Нагрузка не выдержана

### Сценарий 2: Пиковая нагрузка (Stress)

- Быстрый рост до 1500 RPS.
- Проверка устойчивости системы и времени восстановления.
- Имитирует пик в мониторинге.

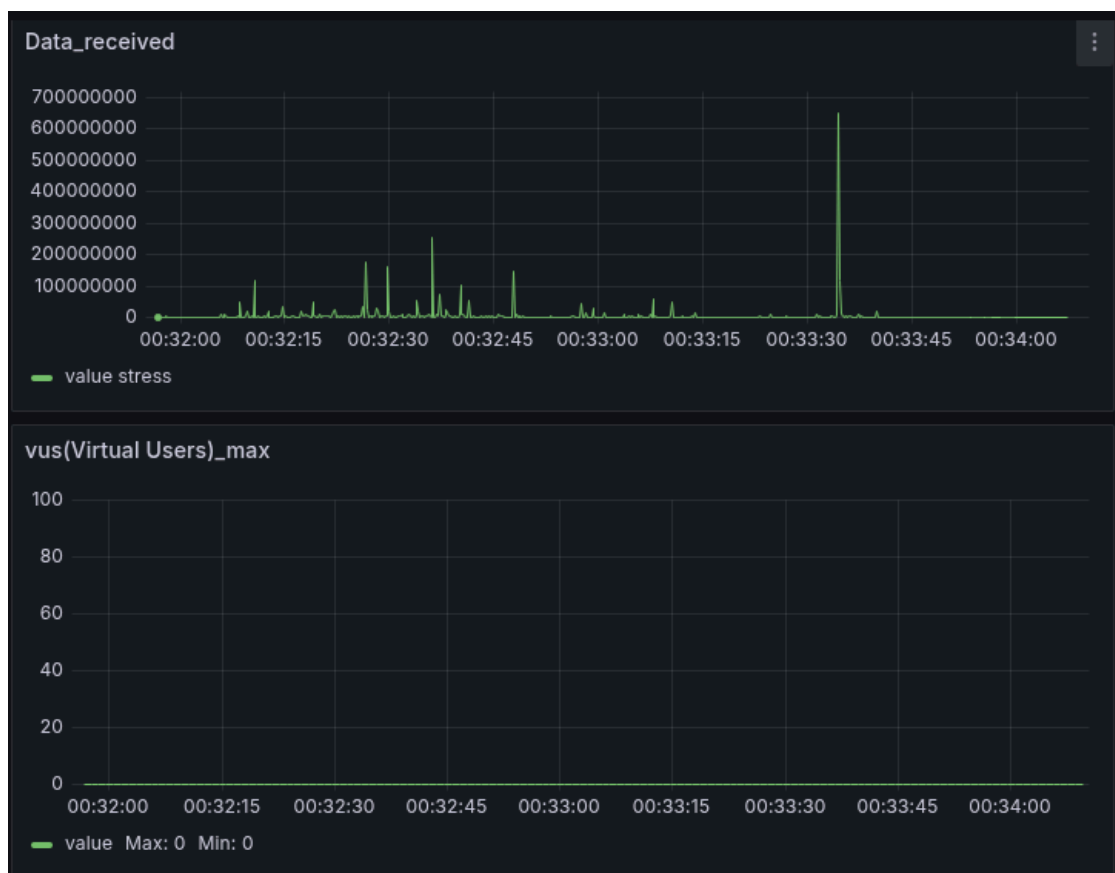


Рис. 15. График динамики полученных данных и пиков виртуальных пользователей для stress тестов



Рис. 16. График динамики отправленных данных и активности виртуальных пользователей для stress тестов

Параметр	Значение	Оценка
<b>Длительность</b>	2 минуты	короткий пик
<b>Цель нагрузки</b>	1500 iters/sec	достигнуто
<b>Max VUs</b>	1500	достигнуто
<b>Ошибки</b>	есть (404, timeout)	допустимо для стресса
<b>Dropped iters</b>	да	не выдержал всю нагрузку

Вывод:

- Тест максимально нагрузил систему, и она показала реальные пределы.
- Стресс-тест максимум выдержал, но с ошибками и просадкой по ответам.
- jira\_projects дал p95 = 16.8 сек, max = 60 сек.
- Сильно выросло количество dropped итераций — система не тянет пик, что и нужно было выяснить.

### Сценарий 3: Долговременное тестирование (Soak)

- Постоянная нагрузка на уровне 60–70% от предельной.
- Продолжительность: 4–6 часов.
- Цель — выявить утечки памяти, деградацию производительности.

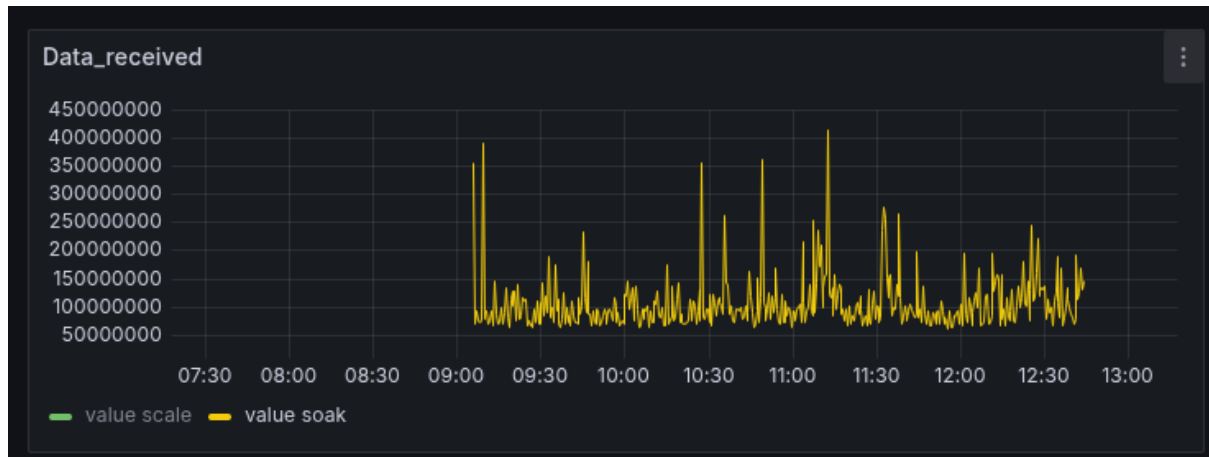


Рис. 17. График динамики полученных данных для Soak тестов

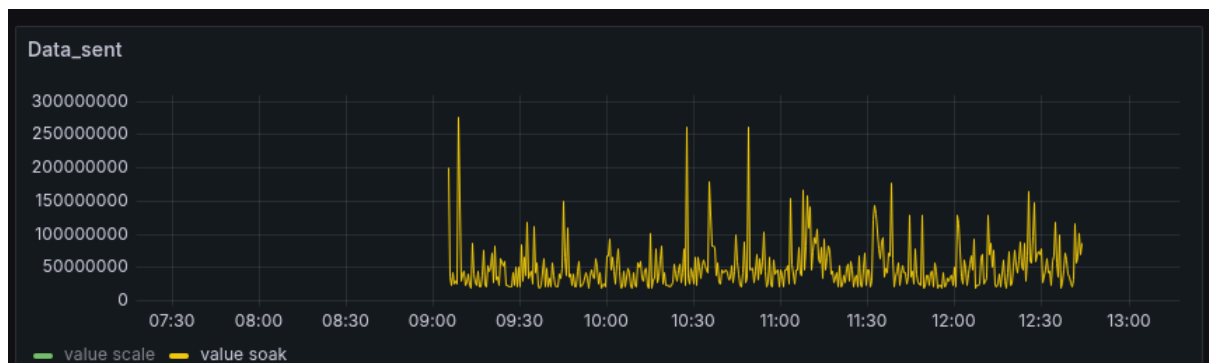


Рис. 18. График динамики отправленных данных для Soak тестов



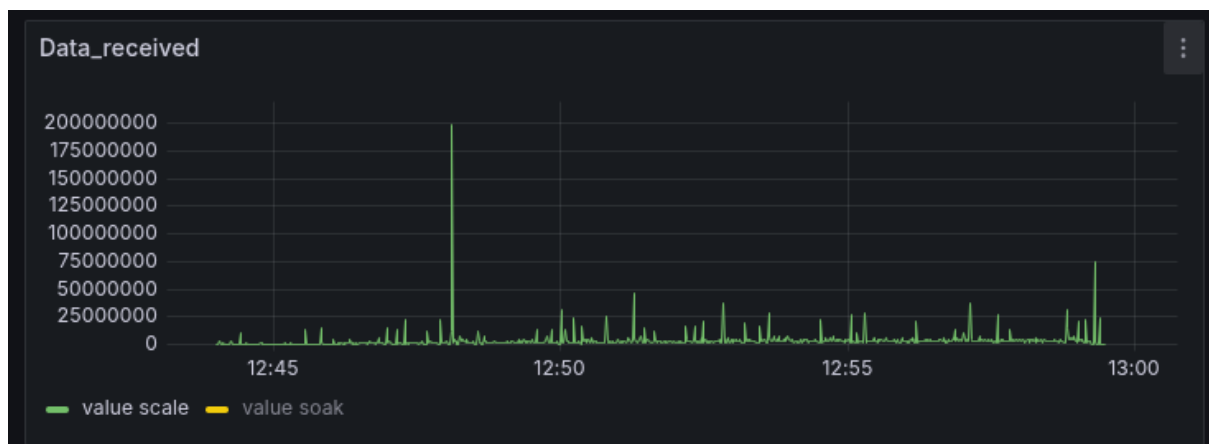
Рис. 19. Общая информация о прохождении Soak тестов

Метрика	Значение / Комментарий
Достигнутый RPS	~700 it/s соответствует цели
Пиковое время	http_req_duration p95: 20.18ms — стабильно
Проблемы	816,603 dropped_iterations (~8%) 23.09% ошибок (HTTP 500 / timeout) - много
Среднее время ответа по endpoint'ам	Большинство GET запросов — приемлемо update_project, jira_projects и project_stats — среднее > 1 сек
Утечек или роста latency со временем	до 21311214ms (~6 часов) — возможны зависания запросов или ресурсы не очищаются

*Вывод: Система не выдерживает длительную нагрузку без деградации.*

#### **Сценарий 4: Масштабируемость (Scale)**

- Эмуляция добавления новых проектов (больше данных в БД).
- Проверка производительности API при росте нагрузки.



*Рис. 20. График динамики полученных данных для scale тестов*



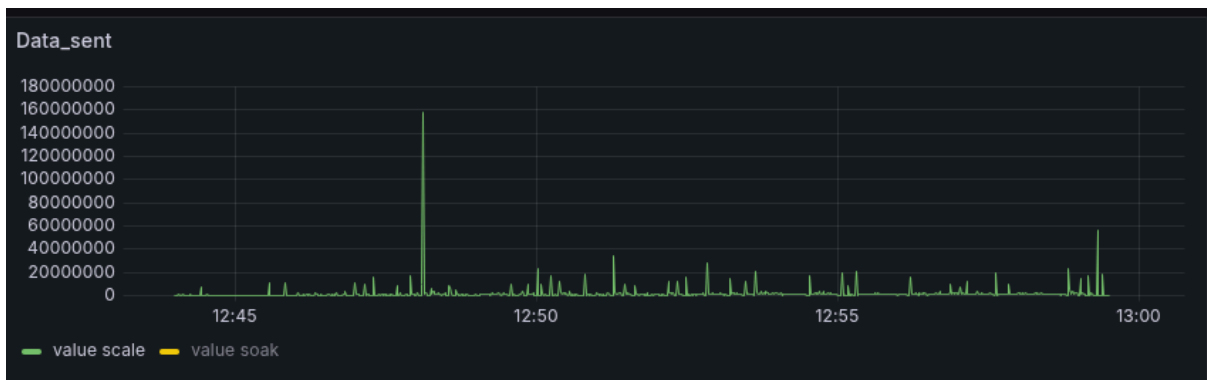


Рис. 21. График динамики отправленных данных для scale тестов



Рис. 22. Общая информация о прохождении scale тестов

Метрика	Значение / Комментарий
Достигнутый RPS	~408 it/s - не достигается цель
Ошибки	23%
Лимиты по VUs	В логах: Insufficient VUs, reached 1000 active VUs — масштабирование ограничено
Latency пост	http_req_duration p95 поднялся до 20.18ms, update_project и jira_projects — до 2463ms и выше

Вывод:

Система не масштабируется пропорционально росту нагрузки:

- RPS растёт медленнее, чем увеличиваются VUs;
- ресурсы или архитектура не выдерживают.

### Индивидуальный анализ endpoint'ов:

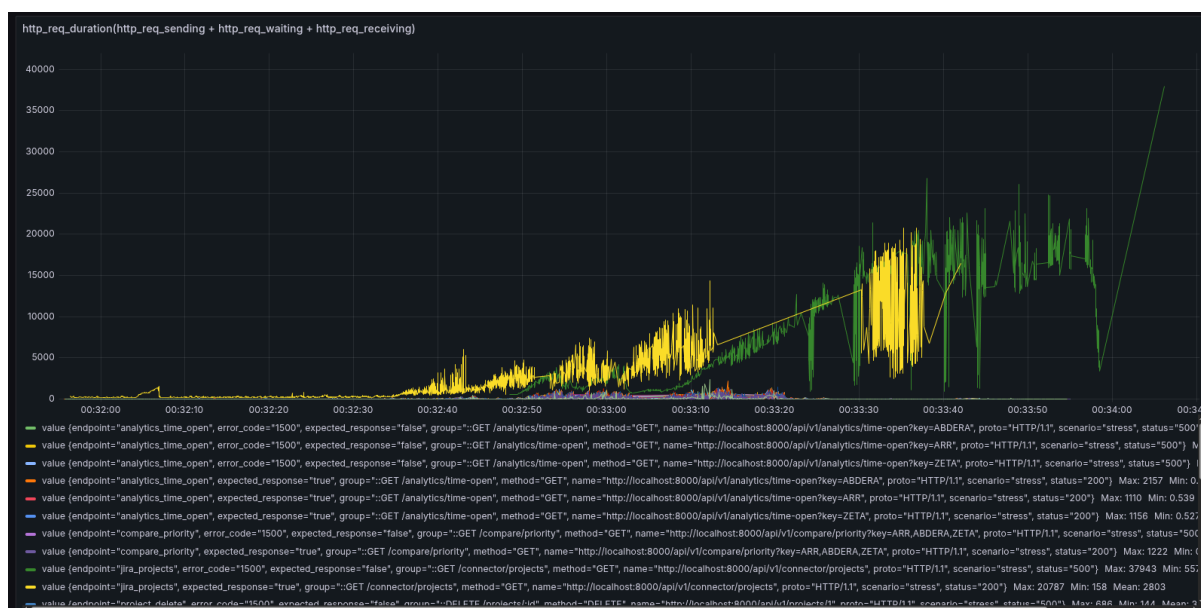


Рис. 23. График динамики активности получения *http*-запросов по разным эндпоинтам

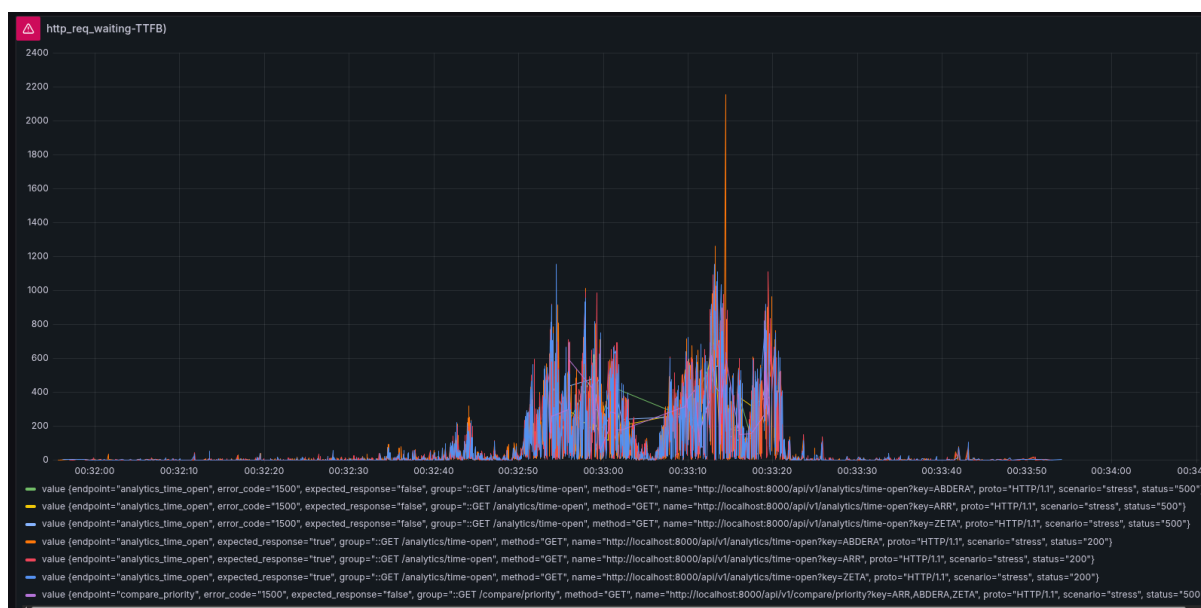


Рис. 24. График динамики ожидания http-запросов по разным эндпоинтам

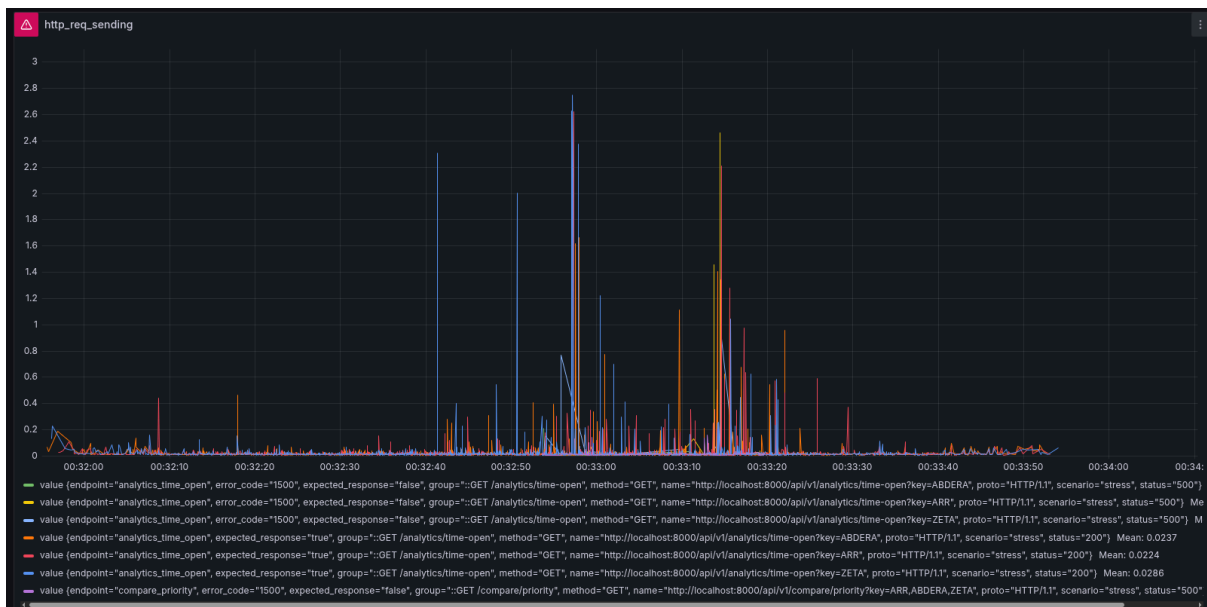


Рис. 25. График динамики активности отправления http-запросов по разным эндпоинтам

Endpoint	p95 (мс)	Max (мс)	Вывод
jira_projects	16868	60008	Узкое место
project_stats	83.2	2519	Допустимо
projects_list	27.4	1936	Хорошо
compare_priority	36.3	1896	Хорошо
update_project	24.4	5068	Хорошо
project_delete	32.9	1931	Хорошо
analytics_time_open	38.1	2019	Хорошо

Вывод: *jira\_projects* нуждается в оптимизации.

## Итоги проверки

### Общие результаты

Показатель	Значение	Доп. информация
Всего проверок (checks_total)	372 869 (362.525062/s)	
Успешные проверки (checks_succeeded)	367 573	98.57% от общего числа
Неудачные проверки (checks_failed)	5 296	1.42% от общего числа
Статус 200	✓ 367 573 / ✗ 5 296	98% успешных

### Пользовательские метрики (CUSTOM)

Метрика	Средне е (avg)	Миниму м (min)	Медиан а (med)	Максиму м (max)	p(90)	p(95)
analytics_time_open	15.406361	0.478504	0.895182	2019.273036	7.20976	38.114469
compare_priority	15.091051	0.360329	0.704941	1896.311283	6.462513	36.381096
jira_projects	2100.621025	168.246138	296.90803	60008.809401	3403.620295	16868.05574
project_delete	14.221508	0.261225	0.530432	1931.569733	5.466846	32.943741
project_stats	23.08175	1.187742	1.880561	2519.682665	15.412256	83.190146
projects_list	13.314306	0.20354	0.437003	1936.767415	3.640192	27.395261
update_project	38.309831	0.287863	0.707344	5068.988813	4.3034	24.438531

### HTTP-метрики

Метрика	Значение	Доп. информация
http_req_duration	avg=392.87ms	min=203.54µs, max=1m0s
	med=1.04ms	p(90)=299.56ms, p(95)=376.78ms
{expected_response:true}	avg=191.06ms	min=203.54µs, max=59.97s
	med=1.01ms	p(90)=289.72ms, p(95)=345.37ms
http_req_failed	1.42% (5 296 из 372 869)	
http_reqs	372 869 (362.525062/s)	

### Исполнение (EXECUTION)

Метрика	Значение	Доп. информация
dropped_iterations	620 129 (602.925704/s)	
iteration_duration	avg=1.39s	min=1s, max=1m1s
	med=1s	p(90)=1.3s, p(95)=1.37s
iterations	372 869 (362.525062/s)	
vus	1	min=1, max=1900
vus_max	1900	min=500, max=1900

### Сеть (NETWORK)

Метрика	Значение
data_received	219 MB (212 kB/s)
data_sent	38 MB (37 kB/s)

### Общий анализ

Метрика	Значение	Комментарий
Всего запросов	372,869	Хороший объём
Успешные проверки	98.57%	Чуть ниже идеала (>99%), но приемлемо
Ошибки (http_req_failed)	1.42%	Стремиться к <1%, что приемлемо
Dropped iterations	620,129 (!)	Очень много — нехватка VU
Максимальный VU	1900	Использованы все
Пиковая задержка (http_req_duration)	95% ≤ 376ms	приемлемо

### Сравнение с заданной архитектурой:

Вопрос	Цель	Итог тестирования
Проходит ли система порог по RPS?	Соответствие throughput.	Ограничения по числу VUs, CPU или базе
Укладывается ли в latency (p95/p99)?	Важно для UX.	Под нагрузкой latency критична
Есть ли рост latency при росте нагрузки?	Поиск точки деградации.	Сильно растёт → деградация

Как быстро восстанавливается система после пикирования?	Надежность и отказоустойчивость.	Остаточные ошибки и таймауты
Какие компоненты дают наибольшую нагрузку?	Узкие места в архитектуре.	/connector/update Project, jira_projects

**Вывод: предложенная схема масштабирования актуальная для улучшения ключевых показателей работы системы**

## Сборка

Описание также лежит в Jira-analyzer/deployment/[README.md](#)

### Запуск приложения

Описание запуска приложения через task, make, docker compose. Перейдите в папку Jira-Analyzer/deployment и все дальнейшие шаги выполняйте оттуда.

Сайт открывает по адресу: `http://localhost:80`

### Taskfile

Удобно, кроссплатформенно, простой синтаксис (требует предварительную установку).

```
task up                # Запустить всё в Docker
task unit-test         # Юнит тесты
task integration-test   # Интеграционные тесты
task down              # Остановить контейнеры
task clean             # Полная очистка
task logs              # Просмотр логов
task db                # Подключение к консоли БД
task all               # Полный пайплайн
```

### Makefile

Для Unix-систем. На Windows работает через WSL или Git Bash

```
make up                # Запустить всё
make unit-test         # Юнит тесты
make integration-test   # Интеграционные тесты
make down              # Остановить
make clean             # Удалить контейнеры и тома
```

```
make logs          # Логи всех сервисов
make db            # Подключение к консоли БД
make all           # Полный пайплайн
```

## Команды docker compose

*Прямой и универсальный способ — просто команды Docker.*

```
# Сборка и запуск
docker compose up --build -d

# Запуск юнит-тестов эндпоинтов
docker compose run --rm backend go test ./... -v
# Запуск юнит-тестов коннектора
docker compose run --rm jiraconnector go test ./... -v

# Запуск интеграционных тестов для коннектора
docker compose run --rm jiraconnector go test
./tests/integration/... -v -tags=integration
# Запуск интеграционных тестов для всей системы
docker compose run --rm backend-test

# Остановка и удаление
docker compose down

# Полная очистка
docker compose down -v --remove-orphans

# Логи
docker compose logs -f

# Состояние базы данных
docker exec -it deployment-postgres-1 psql -U postgres -d
testdb
```

---



# Приложение 1. Swagger-документация API

POST

/updateProject

Обновить или загрузить проект из Jira

^

Получает (или обновляет) все issues из проекта с указанным ключом

Parameters

Try it out

Name	Description
project * required	Ключ проекта в Jira
string (query)	<input type="text" value="project"/>

Responses

Code	Description	Links
200	Проект успешно обновлен <div>Media type application/json Control: Accept header</div> <div>Example Value   <pre>{   "message": "Project updated successfully" }</pre></div>	No links
400	Ошибка запроса (например, не передан параметр project) <div>Media type application/json</div> <div>Example Value   <pre>{   "error": "Project key is required" }</pre></div>	No links
404	Проект не найден в Jira <div>Media type application/json</div> <div>Example Value   <pre>{   "error": "Project not found" }</pre></div>	No links
500	Внутренняя ошибка сервера <div>Media type application/json</div> <div>Example Value   <pre>{   "error": "Internal server error" }</pre></div>	No links

GET

/api/v1/isAnalyzed

Проверка наличия аналитики для проекта

⌵

Parameters

Try it out

Name	Description
<b>project</b> * <small>required</small>	Ключ проекта
string <small>(query)</small>	<input type="text" value="project"/>

Responses

Code	Description	Links
200	Проверка завершена <div>Media type <div>application/json</div><div>⌵</div><div>Content-accept header</div></div> <div>Example Value  </div> <div><pre>{   "project": "XYZ",   "isAnalyzed": true }</pre></div>	No links
400	Ошибка запроса <div>Media type <div>application/json</div><div>⌵</div></div> <div>Example Value  </div> <div><pre>{   "error": "Project key is missing" }</pre></div>	No links
500	Внутренняя ошибка сервера <div>Media type <div>application/json</div><div>⌵</div></div> <div>Example Value  </div> <div><pre>{   "error": "Analysis service failure" }</pre></div>	No links

GET

/projects

Получить список проектов

⌵

Возвращает список доступных проектов с пагинацией и поиском

Parameters

Try it out

Name	Description
limit	Количество проектов на странице (по умолчанию 20)
integer	
(query)	<input type="text" value="20"/>
page	Номер страницы (по умолчанию 1)
integer	
(query)	<input type="text" value="1"/>
search	Фильтр по имени или ключу проекта
string	
(query)	<input type="text" value="search"/>

Responses

Code	Description	Links
200	Успешный ответ со списком проектов	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <pre>{  "Projects": [    {      "Id": 1,      "Key": "ABC",      "Name": "Project ABC",      "Url": "https://jira.example.com/projects/ABC",      "Existence": true    }  ],  "pageInfo": {    "currentPage": 1,    "pageCount": 10,    "projectsCount": 100  }}</pre>		
500	Внутренняя ошибка сервера	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <pre>{  "error": "Database connection failed"}</pre>		

GET

/api/v1/projects

Получить список загруженных проектов

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Список проектов в БД	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <pre>[  {    "Id": 1,    "Key": "XYZ",    "Name": "Project XYZ",    "Url": "https://jira.example.com/projects/XYZ"  }]</pre>		
500	Внутренняя ошибка сервера	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <pre>{  "error": "Database error"}</pre>		

GET

/api/v1/projects/{id} Получить статистику проекта

⌵

Parameters

Try it out

Name	Description
id <small>required</small>	ID проекта в БД
integer (path)	<input type="text" value="id"/>

Responses

Code	Description	Links
200	<div>Данные о статистике проекта</div> <div>Media type</div> <div>application/json</div> <div>Content-accept header</div> <div>Example Value</div> <div><pre>{  "id": 1,  "key": "XYZ",  "name": "Project XYZ",  "allIssuesCount": 100,  "openIssuesCount": 20,  "closeIssuesCount": 70,  "resolvedIssuesCount": 10,  "reopenedIssuesCount": 5,  "progressIssuesCount": 10,  "averageTime": 12,  "averageIssuesCount": 3}</pre></div>	No links
404	<div>Проект не найден</div> <div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{  "error": "Project not found"}</pre></div>	No links
500	<div>Внутренняя ошибка сервера</div> <div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{  "error": "Unexpected error occurred"}</pre></div>	No links

No links

GET

/api/v1/compare/{taskNumber} Получение данных по аналитической задаче для нескольких проектов

⌵

Parameters

Try it out

Name	Description
<b>taskNumber</b> <small>* required</small> integer <small>(path)</small>	Индекс аналитической задачи <input type="text" value="taskNumber"/>
<b>project</b> <small>* required</small> string <small>(query)</small>	Ключи проектов, разделенные запятой <input type="text" value="project"/>

Responses

Code	Description	Links
200	Данные аналитики по проектам <div>Media type <div>application/json</div><div>Control: Accept header</div><div>Example Value   <pre>{   "taskNumber": 2,   "projects": [     "ABC",     "XYZ"   ],   "comparisonData": {     "...": null   } }</pre></div></div>	No links
400	Ошибка запроса <div>Media type <div>application/json</div><div>Example Value   <pre>{   "error": "Invalid project keys" }</pre></div></div>	No links
404	Задача не найдена <div>Media type <div>application/json</div><div>Example Value   <pre>{   "error": "Analysis task not found" }</pre></div></div>	No links
500	Внутренняя ошибка сервера <div>Media type <div>application/json</div><div>Example Value   <pre>{   "error": "Comparison service failure" }</pre></div></div>	No links

DELETE

/api/v1/projects/{id} Удалить проект из БД

^

Parameters

Try it out

Name	Description
id * required	
integer	
(path)	

Responses

Code	Description	Links
200	Проект успешно удален	No links
	<div>Media type</div> <div>application/json</div> <div>Content Accept header:</div> <div>Example Value</div> <div><pre>{  "message": "Project deleted successfully"}</pre></div>	
404	Проект не найден	No links
	<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{  "error": "Project not found"}</pre></div>	
500	Внутренняя ошибка сервера	No links
	<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{  "error": "Deletion failed due to server issue"}</pre></div>	

GET

/api/v1/conconnector/projects

Получить список доступных проектов из Jira

^

Parameters

Try it out

Name	Description
limit	Количество проектов на странице (по умолчанию 20)
integer	Default value : 20
(query)	<input type="text" value="20"/>
page	Номер страницы (по умолчанию 1)
integer	Default value : 1
(query)	<input type="text" value="1"/>
search	Фильтр по имени или ключу проекта
string	<input type="text" value="search"/>
(query)	

Responses

Code	Description	Links
200	Успешный ответ со списком проектов	No links
	<div>Media type</div> <div>application/json</div> <div>Control accept header:</div> <div>Example Value</div> <div><pre>{  "Projects": [    {      "Id": 1,      "Key": "DEF",      "Name": "Project DEF",      "Url": "https://jira.example.com/projects/DEF",      "Existence": false    }  ],  "PageInfo": {    "currentPage": 1,    "pageCount": 5,    "projectsCount": 50  }}</pre></div>	

	400	Ошибка запроса	No links
	Media type  application/json  Example Value  ``` {  "error": "Invalid pagination parameters"} ```		
500	Внутренняя ошибка сервера	No links	
	Media type  application/json  Example Value  ``` {  "error": "Jira API connection failed"} ```		

POST

/api/v1/conector/updateProject

Обновить или загрузить проект по ключу

^

Parameters

Try it out

Name	Description
<b>project *</b> <small>required</small>	Ключ проекта в Jira
string (query)	<input type="text" value="project"/>

Responses

Code	Description	Links
200	Проект успешно обновлен <div>Media type <div>application/json</div><div>Controls: Accept header</div><div>Example Value  </div><div><pre>{   "message": "Project updated successfully" }</pre></div></div>	No links
400	Некорректный запрос <div>Media type <div>application/json</div><div>Example Value  </div><div><pre>{   "error": "Missing project key" }</pre></div></div>	No links
404	Проект не найден <div>Media type <div>application/json</div><div>Example Value  </div><div><pre>{   "error": "Project not found in Jira" }</pre></div></div>	No links
500	Внутренняя ошибка сервера <div>Media type <div>application/json</div><div>Example Value  </div><div><pre>{   "error": "Jira API error" }</pre></div></div>	No links



GET

/api/v1/graph/get/{taskNumber}

Получить данные по аналитической задаче

^

Parameters

Try it out

Name	Description
<b>taskNumber</b> * required	Номер аналитической задачи
integer (path)	<input type="text" value="taskNumber"/>
<b>project</b> * required	Ключ проекта
string (query)	<input type="text" value="project"/>

Responses

Code	Description	Links
200	Данные аналитики получены <div>Media type application/json</div> <div>Content Accept Header</div> <div>Example Value  </div> <div><pre>{  "taskNumber": 1,  "project": "XYZ",  "analysisData": {    "...": null  }}</pre></div>	No links
400	Ошибка запроса <div>Media type application/json</div> <div>Example Value  </div> <div><pre>{  "error": "Missing project key"}</pre></div>	No links
404	Задача или проект не найдены <div>Media type application/json</div> <div>Example Value  </div> <div><pre>{  "error": "Analysis task not found"}</pre></div>	No links
500	Внутренняя ошибка сервера <div>Media type application/json</div> <div>Example Value  </div> <div><pre>{  "error": "Analysis service failure"}</pre></div>	No links

POST

/api/v1/graph/make/{taskNumber}Проведение аналитической задачи

^

Parameters

Try it out

Name	Description
<b>taskNumber</b> * required	Индекс аналитической задачи
integer (path)	<input type="text" value="taskNumber"/>
<b>project</b> * required	Ключ проекта
string (query)	<input type="text" value="project"/>

Responses

Code	Description	Links
200	Аналитическая задача выполнена	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <div><pre>{   "message": "Analysis completed successfully" }</pre></div>		
400	Ошибка запроса	No links
<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{   "error": "Invalid project key" }</pre></div>		
500	Внутренняя ошибка сервера	No links
<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{   "error": "Analysis service error" }</pre></div>		

DELETE

/api/v1/graph/deleteУдаление всех аналитических задач для проекта

^

Parameters

Try it out

Name	Description
<b>project</b> * required	Ключ проекта
string (query)	<input type="text" value="project"/>

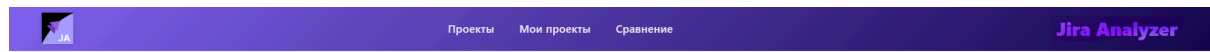
Responses

Code	Description	Links
200	Аналитические задачи удалены	No links
<div>Media type</div> <div>application/json</div> <div>Content Accept Header</div> <div>Example Value</div> <div><pre>{   "message": "All analysis tasks deleted" }</pre></div>		
400	Ошибка запроса	No links
<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{   "error": "Project key is required" }</pre></div>		
500	Внутренняя ошибка сервера	No links
<div>Media type</div> <div>application/json</div> <div>Example Value</div> <div><pre>{   "error": "Failed to delete analysis tasks" }</pre></div>		

## Приложение 2. Примеры работы приложения

### Страница со всеми проектами:

- Карточки проектов с возможностью добавить/удалить в локальную БД
- Поиск, пагинация



#### Все проекты

Поиск проектов...

aardvark  
AAR  
Перейти → Удалить

Abdera  
ABDERA  
Перейти → Добавить

Accumulo  
ACCUMULO  
Перейти → Добавить

ACE  
ACE  
Перейти → Добавить

ActiveCluster  
ACL  
Перейти → Удалить

ActiveMQ .Net  
AMQNET  
Перейти → Добавить

ActiveMQ Apollo (Retired)  
APLO  
Перейти → Удалить

ActiveMQ Artemis  
ARTEMIS  
Перейти → Удалить

ActiveMQ C++ Client  
AMQCPP  
Перейти → Удалить

← Назад 1 / 75 Вперед →



#### Все проекты

myproject x



Не найдено проектов по запросу "myproject"

Очистить поиск

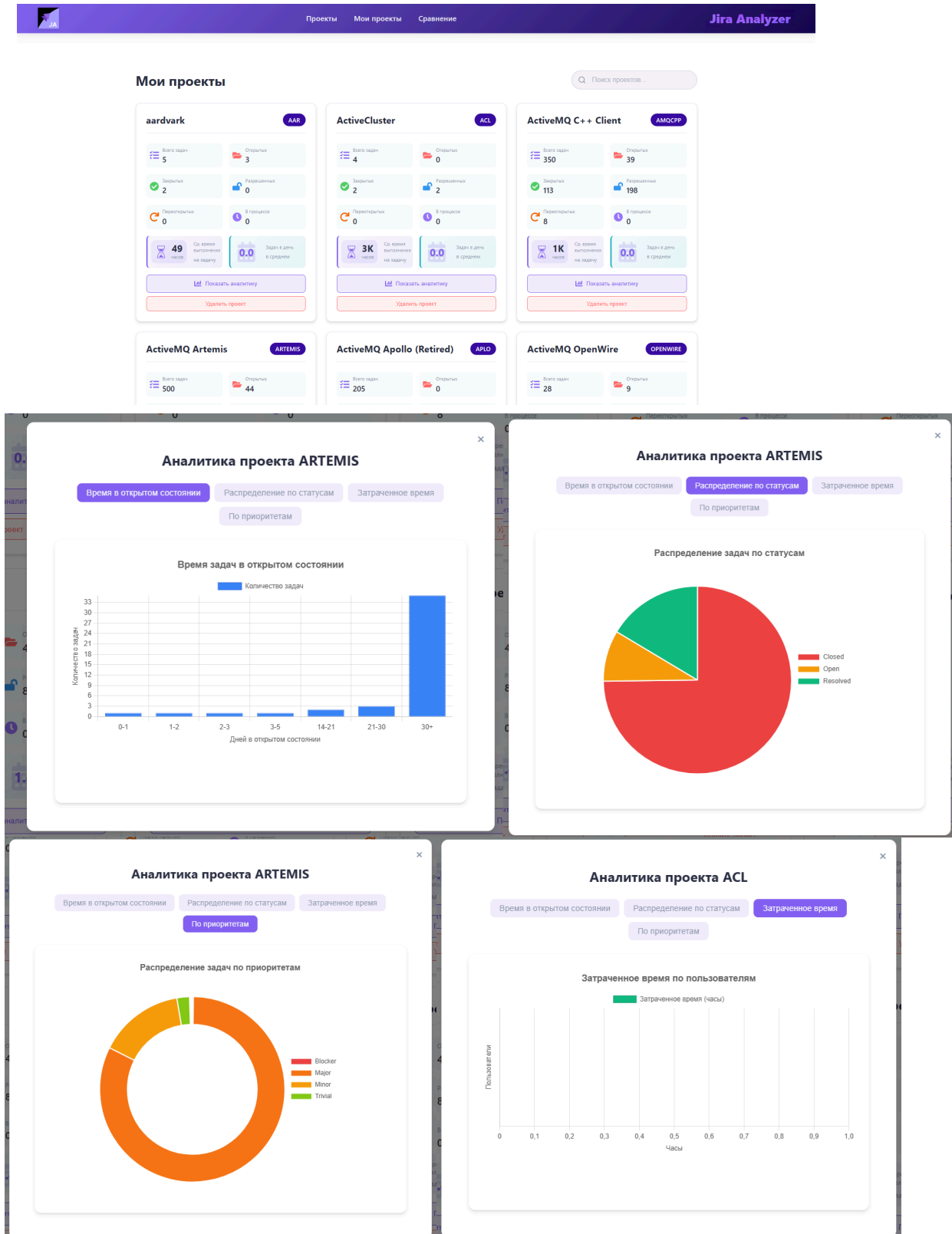
← Назад

3 / 75

Вперед →

## Страница с выгруженными в локальную БД проектами:

- Карточки проектов с просмотром сухой статистики, возможностью удалить проект, а также просмотреть аналитику по проекту
- Поиск



Страница для сопоставления нескольких проектов:

- Селектор проектов (2-3 проекта) с возможностью добавить/удалить из списка выбранные проекты
- Просмотр сопоставления сухой статистики и диаграмм

JA

ПроектыМои проектыСравнение

Jira Analyzer

Сравнение проектов

Доступные проекты

aardvark (AAR)

ActiveCluster (ACL)

ActiveMQ C++ Client (AMQCPP)

ActiveMQ Artemis (ARTEMIS)

ActiveMQ Apollo (Retired) (APLO)

ActiveMQ OpenWire (OPENWIRE)

Выбранные для сравнения

ActiveCluster (ACL)

ActiveMQ Artemis (ARTEMIS)

ActiveMQ OpenWire (OPENWIRE)

Сравнить выбранные проекты (3)

Сравнение проектов

Сухая статистика

Графики

Сухая статистика	ActiveCluster	ActiveMQ Artemis	ActiveMQ OpenWire
Общее количество задач	4	500	28
Количество открытых задач	0	44	9
Количество закрытых задач	2	374	8
Количество переоткрытых задач	0	4	0
Количество разрешенных задач	2	82	11
Количество задач "In progress"	0	0	0
Среднее время выполнения (часы)	3261.38	426.84	1179.36
Среднее количество задач в день	0	1.43	0

