

Neutrino APM

Руководство Пользователя

Содержание

Введение	3
Определения	4
Описание системы	5
Архитектура	6
Пользовательский интерфейс	8
Общие элементы навигации	9
Обзор	11
Сервисы	12
Инстансы	12
Пользователи	13
Анализ	13
Базы данных	14
Детализация по сервису	16
Детализация по эндпоинту	18
Детализация по инстансу	19
Детализация по пользователю	20
Детализация по базе данных	21
Детализация по SQL запросу	21
Детализация по трейсу	23
Дерево профилирования	24
Фильтрация трейсов	25
Дашборды	27
Настройка агента	29

Введение

Данное Руководство предназначено для конечных пользователей ПО и включает следующие разделы:

- Определения;
- Описание системы;
- Архитектура системы;
- Основные функции и возможности;
- Руководство по интерфейсу пользователя;
- Инструкции по использованию основных инструментов и функций.

Определения

Приложение – группа сервисов, отвечающая за работу определенной информационной системы (например, система документооборота или площадка электронной коммерции).

Сервис – функциональный модуль, отвечающий за выполнение работы в рамках информационной системы (например, сервис построения отчетов или сервис обработки очередей).

Инстанс – экземпляр сервиса, как правило, представленный процессом операционной системы.

Эндпоинт – функциональная часть сервиса, отвечающая за выполнение определенного неделимого объема работы (например, авторизация, выполнение поиска или обработка сообщений из очереди).

Трейс – действие в рамках приложения с учетом всех эндпоинтов в рамках всех сервисов, необходимых для корректного выполнения этого действия.

Описание системы

Программное обеспечение «Neutrino APM» — система мониторинга для выявления аномалий, контроля качества и поиска причин снижения производительности в работе приложений с точки зрения сервисов, прикладного кода, баз данных и конечных пользователей.

Программное обеспечение «Neutrino APM» — это комплексная система мониторинга на базе агентов, которые собирают информацию о работе конечных пользователей и приложений. Собранные агентами информация передается, сохраняется и индексируется на сервере анализа и используется как для анализа через пользовательский интерфейс системы, так и для работы алгоритмов выявления аномалий и для публикации через API для интеграции с другими системами мониторинга.

Архитектура

ПО «Neutrino APM» представляет комплексную систему мониторинга.

Ключевые компоненты ПО:

1. **Агенты мониторинга:**
 - Выполняют сбор диагностической информации о работе приложений: метрики, информация о работе потоков и пользователей.
2. **Модуль сбора и обработки данных от агентов мониторинга:**
 - Выполняет приём, разбор, трансформацию и сохранение полученной информации от агентов мониторинга.
3. **Хранилище данных:**
 - Хранит, индексирует и предоставляет доступ к собранной диагностической информации.
4. **Контроллер пользовательского интерфейса:**
 - Предоставляет доступ к данным для проведения анализа.
5. **Модуль выявления аномалий:**
 - Позволяет выявлять аномалии в работе приложения, сервисов и эндпоинтов.

Гибкая архитектура:

Программное обеспечение (Далее также – система, Neutrino APM) разработано с учетом гибкости установки, что позволяет развертывать компоненты на одном сервере, отдельных серверах или виртуальных машинах. Кроме того, ПО может быть интегрировано в кластер Kubernetes, обеспечивая масштабируемость и отказоустойчивость.

Поддержка Операционных Систем:

Все компоненты системы могут быть установлены на современные операционные системы Linux таких как Ubuntu, Debian, Rocky Linux, CentOS, а также Astra Linux и Red OS. Дополнительно, за исключением базы данных, компоненты могут быть развернуты на операционной системе Windows Server.

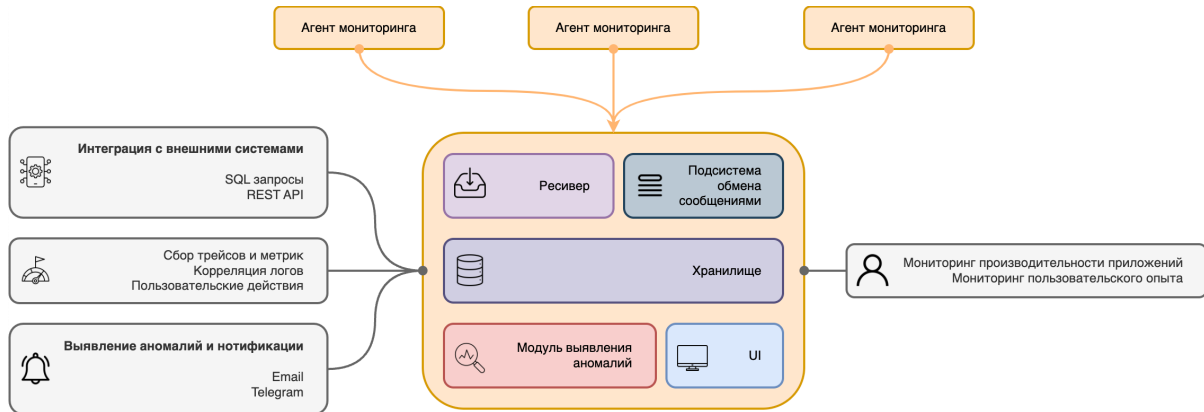
Верхнеуровневая Архитектура:

В схеме ниже представлена верхнеуровневая архитектура решения, демонстрирующая взаимодействие ключевых компонентов в системе.

Для обеспечения работы Neutrino APM задействованы следующие модули:

- Агент мониторинга;
- Модуль сбора и обработки данных;
- Хранилище;
- Контроллер;
- Модуль выявления аномалий.

Архитектура системы представлена на схеме:



Используемые порты и интеграции

Для корректной работы системы необходимо предусмотреть доступность следующих портов:

- TCP/8087 – для приема информации от агентов мониторинга;
- TCP/80,443 – для доступа в пользовательский интерфейс.

Помимо этого, система в процессе своей работы обращается к следующим элементам системы:

- По LDAP обращается к AD для аутентификации пользователей.

Требования к вычислительным ресурсам

Спецификация аппаратной платформы, на которой будет устанавливаться система, зависит от количества и детализации собираемой агентами диагностической информации и от длительности хранения информации. Для помощи с определением спецификации аппаратной платформы обратитесь к технической поддержке.

Пользовательский интерфейс

Структура интерфейса пользователя делится на следующие представления:

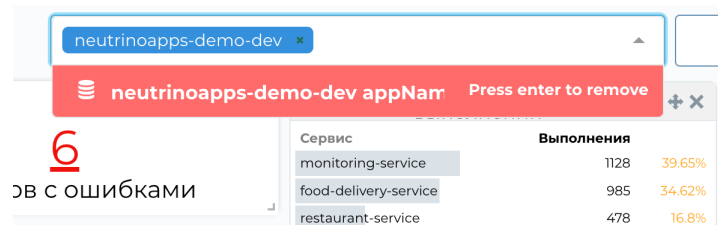
1. Общие элементы навигации:
 - Меню
 - Выбор приложения
 - Выбор временного интервала времени
 - Механизм Zoom-In
 - Механизм Drill-Down
2. Разделы для анализа приложения:
 - Обзор
 - Сервисы
 - Инстансы
 - Пользователи
 - Анализ
 - SQL
3. Разделы для детального анализа таких сущностей:
 - Сервис
 - Эндпоинт
 - Инстанс
 - Пользователь
 - База данных
 - SQL запрос
 - Трейс
 - Дерево профилирования
4. Кастомные дашборды

Общие элементы навигации

В системе предусмотрены общие для всех представлений элементы управления:

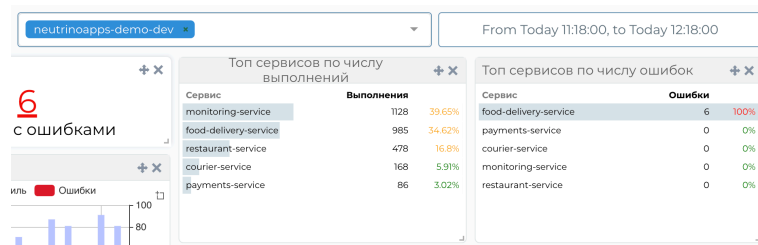
- **Выбор приложения:**

Система позволяет выбрать приложение для анализа из выпадающего списка. После выбора приложения это значение будет применено ко всем открытым вкладкам и дашбордам.



- **Выбор временного интервала:**

Для удобства пользователей доступен выбор различных временных интервалов для отчетов.



Быстрый выбор времени: последние 15 минут, 1 час, 6 часов, 24 часа, 7 дней, 30 дней, сегодня и вчера.

Выбор относительного времени.

Выбор произвольного временного интервала для создания индивидуальных отчетов с учетом специфических задач, параметров и временных рамок.

- Механизм Zoom-In:**

В отчетах системы внедрен механизм zoom-in, который позволяет выделять область на графике и автоматически пересчитывать статистику за выделенный временной интервал. Для использования этого механизма необходимо нажать на соответствующую иконку в правом нижнем углу графика, затем, удерживая левую кнопку мыши, выделить желаемую область на графике и отпустить кнопку мыши. Эта функция доступна на всех графиках.



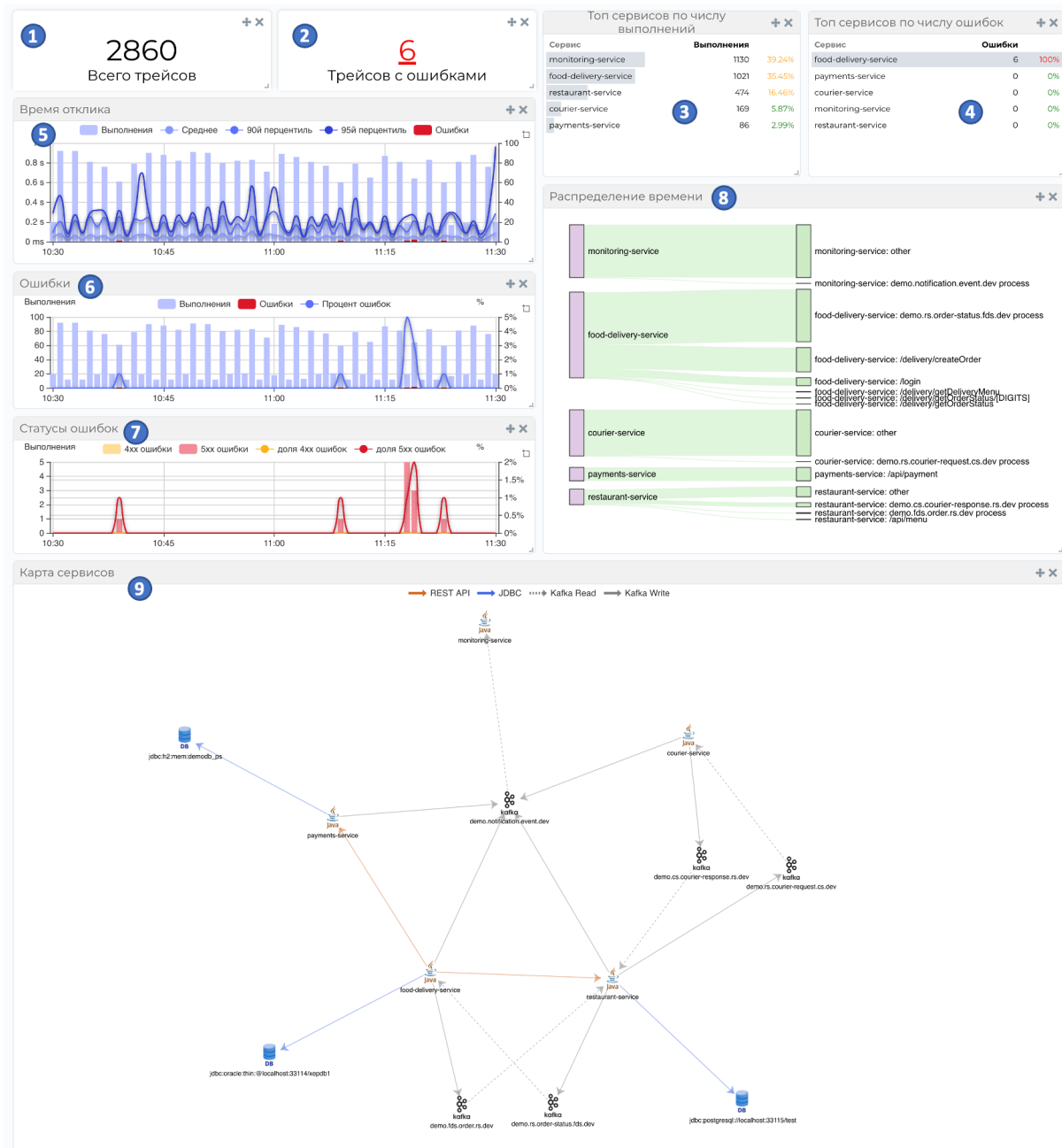
- Механизм Drill-Down**

В системе реализованы сценарии переходов Drill Down между визуальными представлениями, которые позволяют пользователям более детально проводить анализ. Механизм реализован как клик на значения в табличных и других представлениях. При переходах сохраняется текущий выбранный диапазон времени.

Этот гибкий механизм Drill Down предоставляет пользователям возможность глубже исследовать данные в различных контекстах, что обеспечивает более точное и всестороннее понимание работы приложения.

Обзор

Раздел "Обзор" – это первоначальная страница пользовательского интерфейса, предоставляет комплексную информацию об актуальном состоянии приложения.



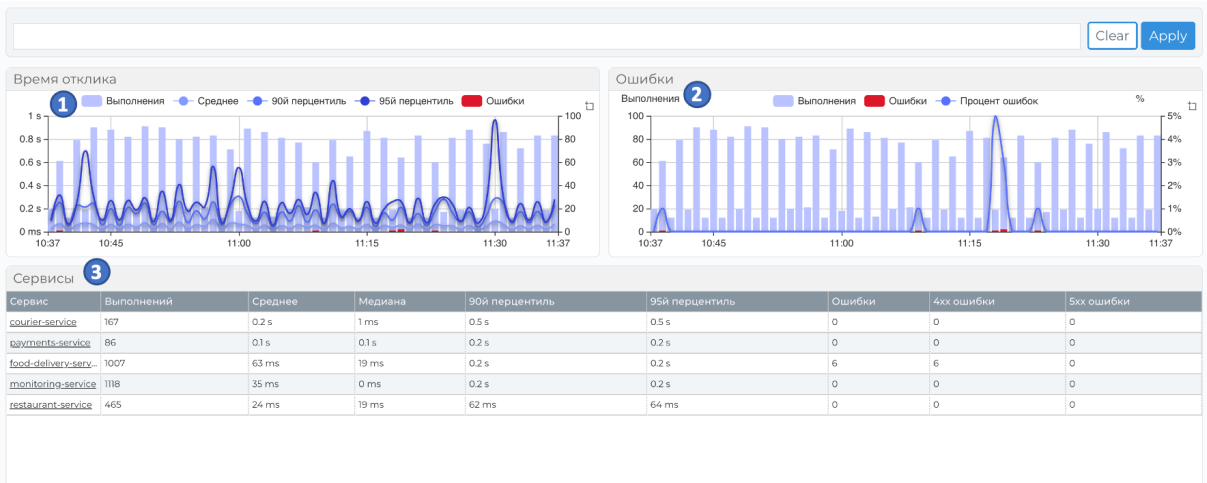
В данном разделе представлены следующие виджеты:

1. Кол-во трейсов;
2. Кол-во трейсов, завершившихся с ошибкой;
3. Перечень сервисов с указанием числа трейсов (поддерживается переход на вкладку с детализацией по сервису);
4. Перечень сервисов с указанием числа трейсов, завершившихся с ошибкой (поддерживается переход на вкладку с детализацией по сервису);

- 5. График с указанием числа трейсов, трейсов с ошибками, средним временем выполнения, максимальными временем выполнения 90% трейсов, максимальным временем выполнения 95% трейсов;
- 6. График с указанием числа трейсов, трейсов с ошибками, процентным соотношением числа трейсов с ошибками к общему числу трейсов;
- 7. График с указанием числа трейсов с 400-ыми и 500-ыми ошибками и процентным соотношением числа таких трейсов к общему числу трейсов;
- 8. Распределение времени работы сервисов и эндпоинтов внутри этих сервисов;
- 9. Карта сервисов, показывающая взаимосвязи между сервисами, очередями сообщений и базами данных с указанием количества запросов между сервисами в минуту и средним временем выполнения вызовов.

Сервисы

Раздел "Сервисы" предназначен для анализа работы сервисов, представленных в приложении.



ClearApply

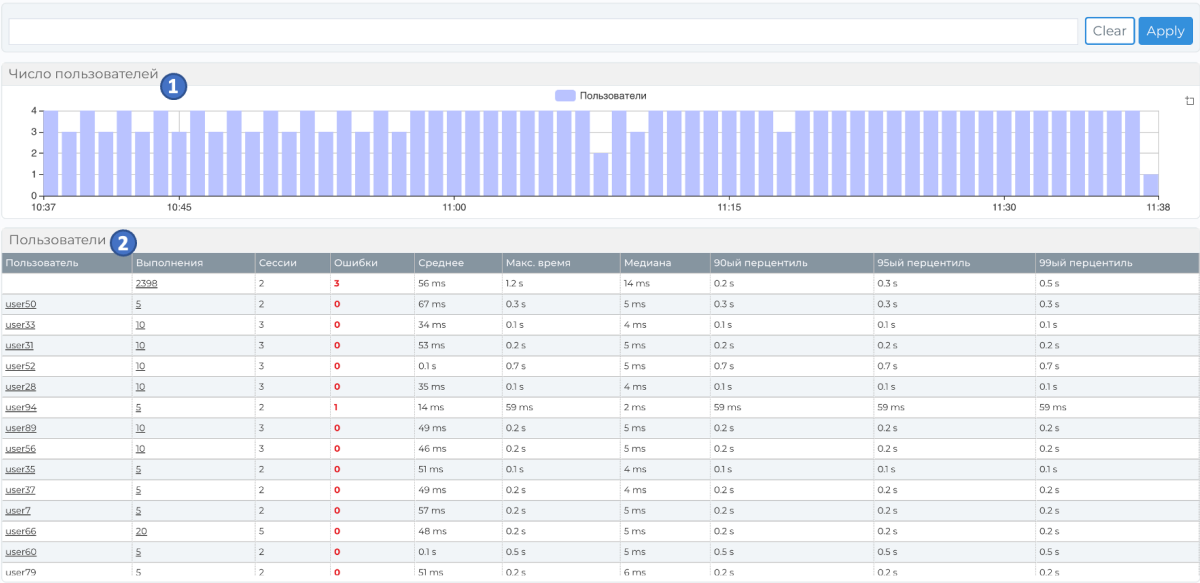
Инстансы1

Узел	Сервис	Инстанс	Версия агента	Выполнения	Ошибки	Среднее	Медиана	90ый перцентиль	95ый перцентиль	99ый перцентиль
host1	payments-ser-	465	20240701	86	0	0.1 s	0.1 s	0.2 s	0.2 s	0.2 s
host1	restaurant-se-	762	20240701	445	0	24 ms	19 ms	62 ms	64 ms	70 ms
host1	courier-service	195	20240701	167	0	0.2 s	1 ms	0.5 s	0.5 s	0.6 s
host1	monitoring-s-	323	20240701	118	0	36 ms	0 ms	0.2 s	0.2 s	0.3 s
host1	food-deliver-	732	20240701	1007	6	63 ms	19 ms	0.2 s	0.2 s	0.5 s

В данном разделе представлены следующие виджеты:

1.
- Таблица с указанием названия узла, где работает инстанс, названия сервиса (поддерживается переход на вкладку с детализацией по сервису), идентификатора инстанса (поддерживается переход на вкладку с детализацией по инстансу), версии агента мониторинга, кол-ва трейсов, среднего времени работы сервиса, медианы времени выполнения, максимального времени выполнения 90% трейсов, максимального времени выполнения 95% трейсов, кол-ва ошибок, кол-ва 400-ых ошибок и кол-ва 500-ых.

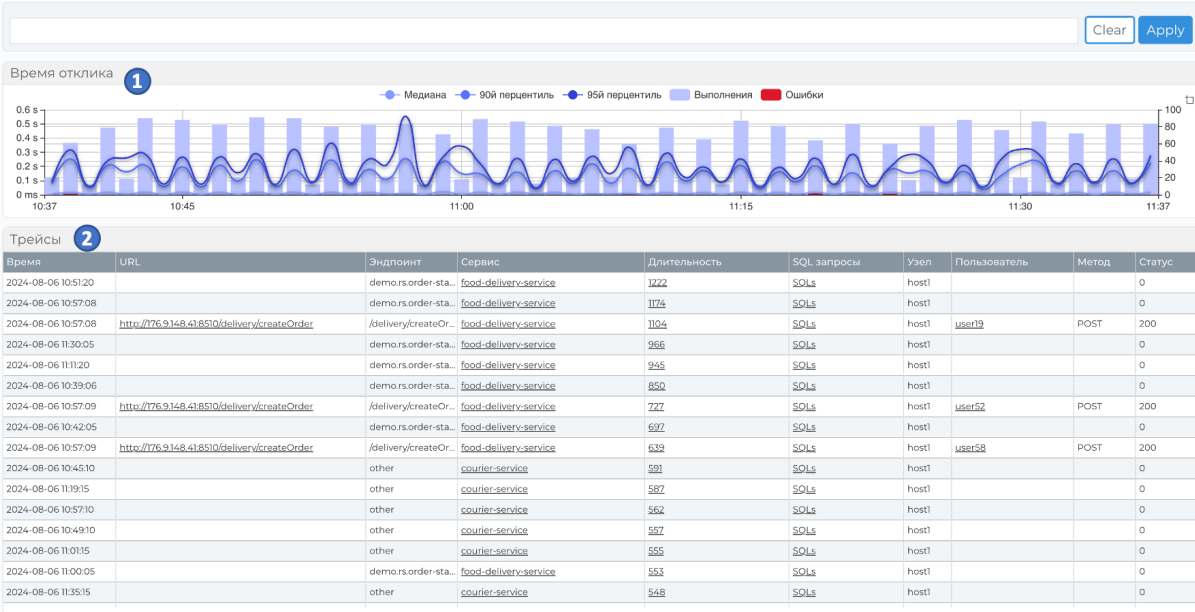
Пользователи



В данном разделе представлены следующие виджеты:

1.
- График числа одновременно работающих пользователей;
2.
- Таблица с указанием имени пользователя (поддерживается переход на вкладку с детализацией по пользователю), кол-ва трейсов у каждого пользователя, кол-ва сессий пользователя, кол-ва трейсов с ошибками, среднего времени работы сервиса, медианы времени выполнения, максимального времени выполнения, максимального времени выполнения 90% трейсов, максимального времени выполнения 95% трейсов;

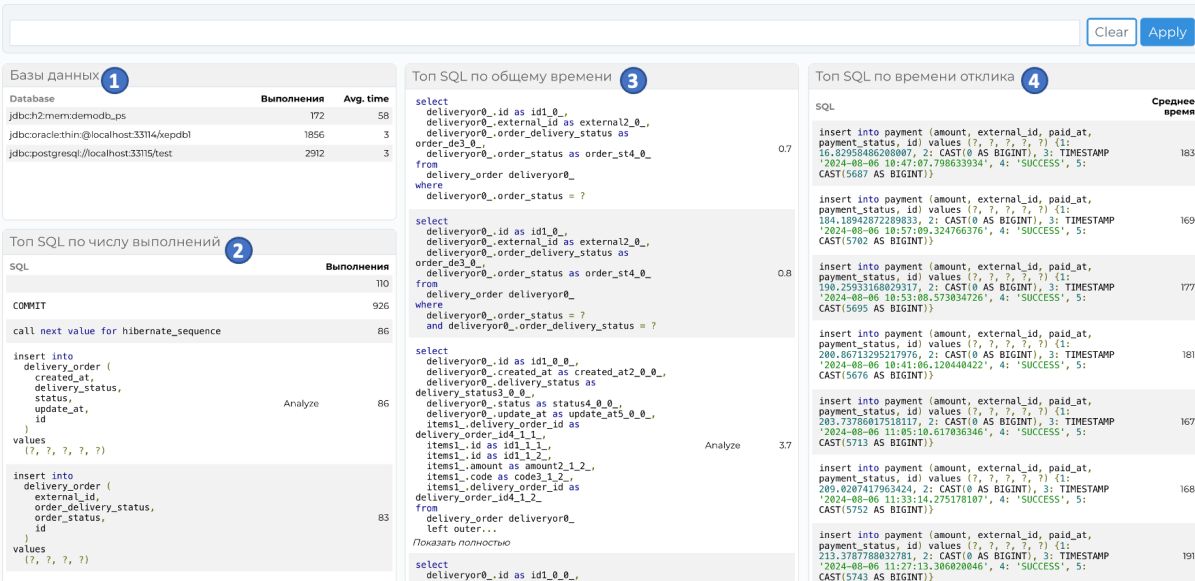
Анализ



В данном разделе представлены следующие виджеты:

- График с указанием числа трейсов, трейсов с ошибками, средним временем выполнения, максимальными временем выполнения 90% трейсов, максимальным временем выполнения 95% трейсов;
- Таблица с трейсами, где указано время начала трейса, url трейса (поддерживается переход на вкладку с детализацией по трейсу), наименование эндпоинта (поддерживается переход на вкладку с детализацией по эндпоинту), наименование сервиса (поддерживается переход на вкладку с детализацией по сервису), длительность трейса (поддерживается переход на вкладку с деревом профилирования трейса), время выполнения SQL запросов (поддерживается переход на вкладку с детализацией по SQL запросам), имя узла, пользователь, метод http, статус http.

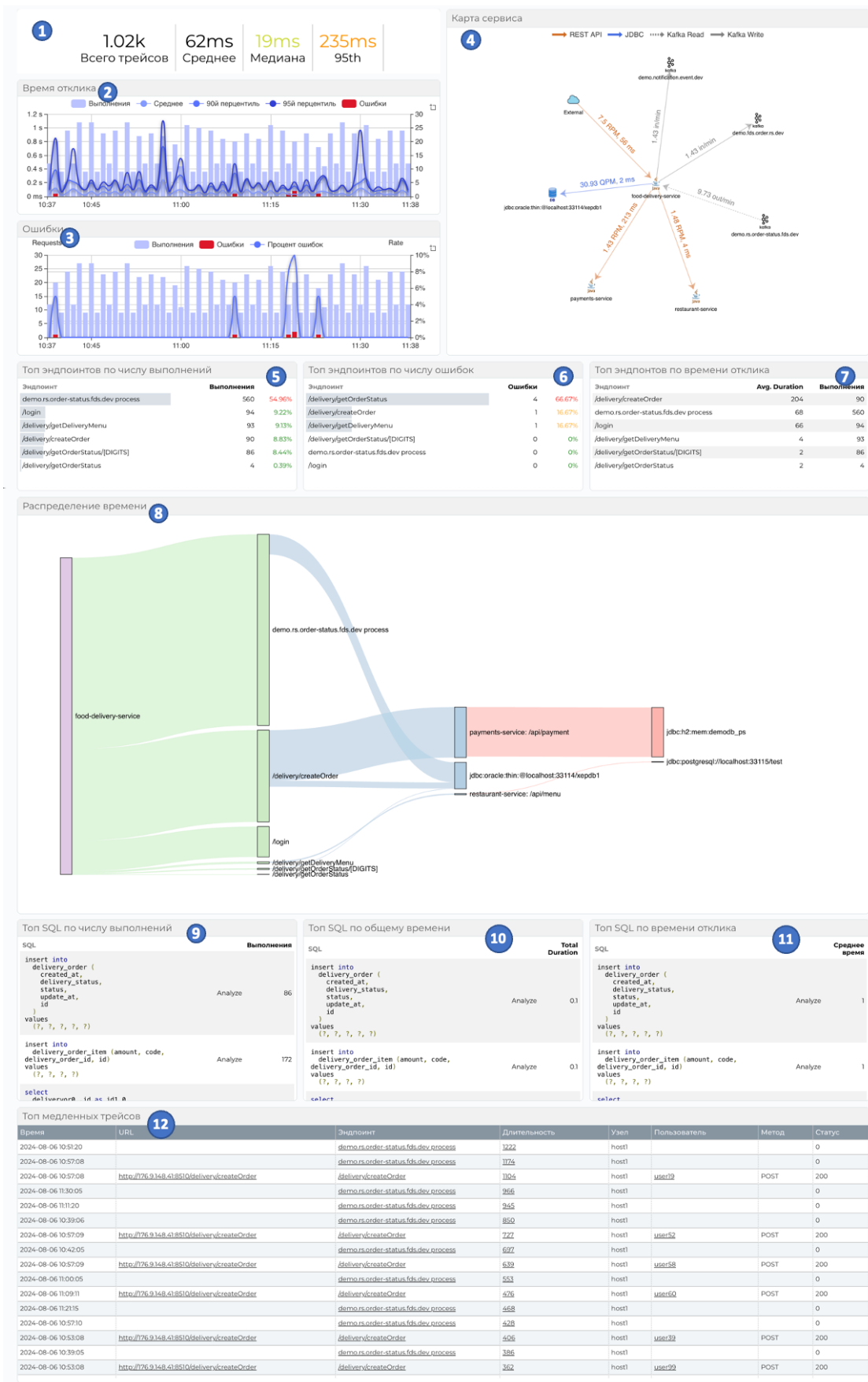
Базы данных



В данном разделе представлены следующие виджеты:

1. Список баз данных с указанием числа выполненных запросов и средним временем выполнения (поддерживается переход на вкладку с детализацией по базе данных);
2. Список запросов с указанием числа их выполнения (поддерживается переход на вкладку с детализацией по запросу);
3. Список запросов с указанием суммарного времени их выполнения (поддерживается переход на вкладку с детализацией по запросу);
4. Список запросов с указанием среднего времени их выполнения (поддерживается переход на вкладку с детализацией по запросу).

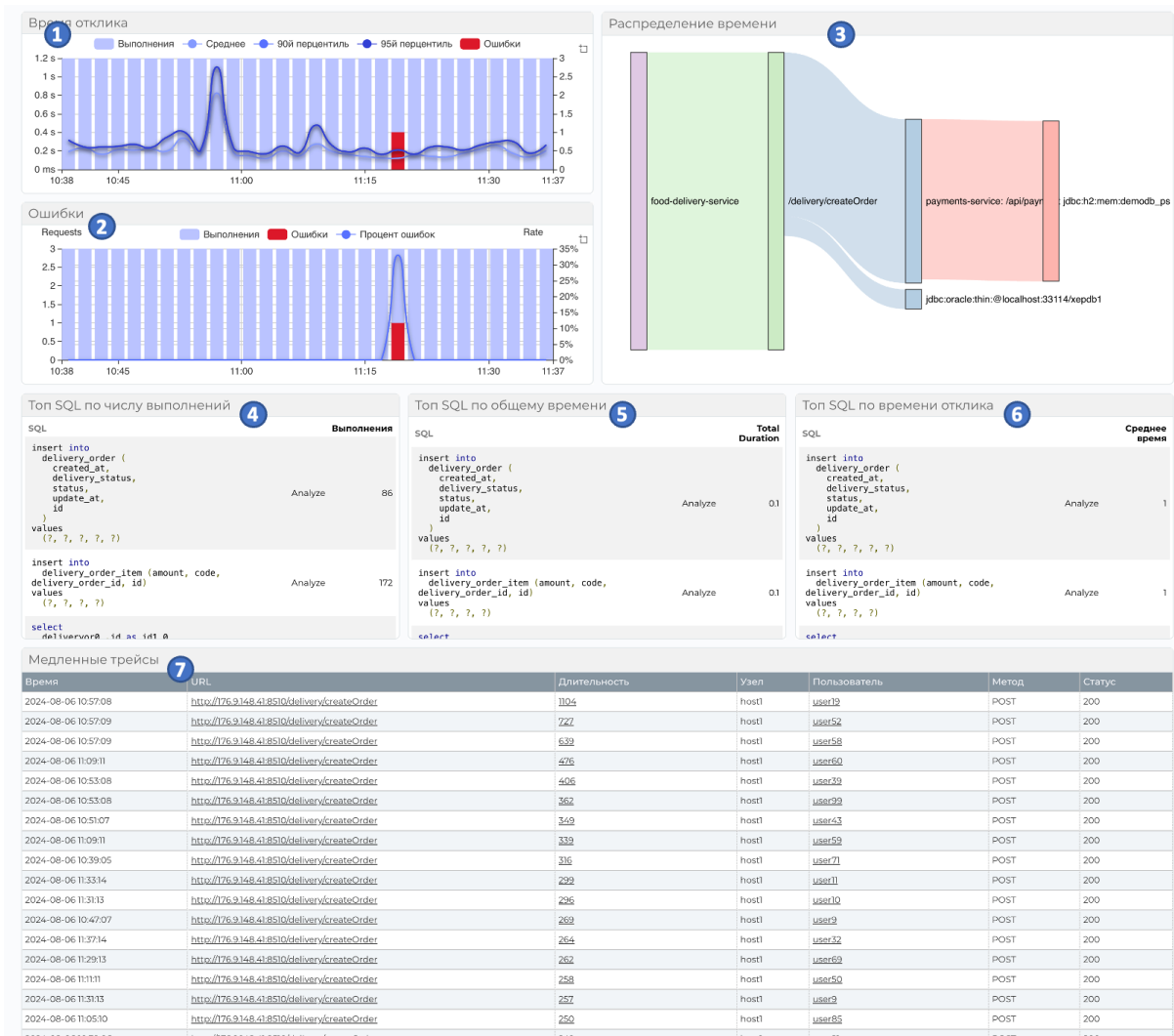
Детализация по сервису



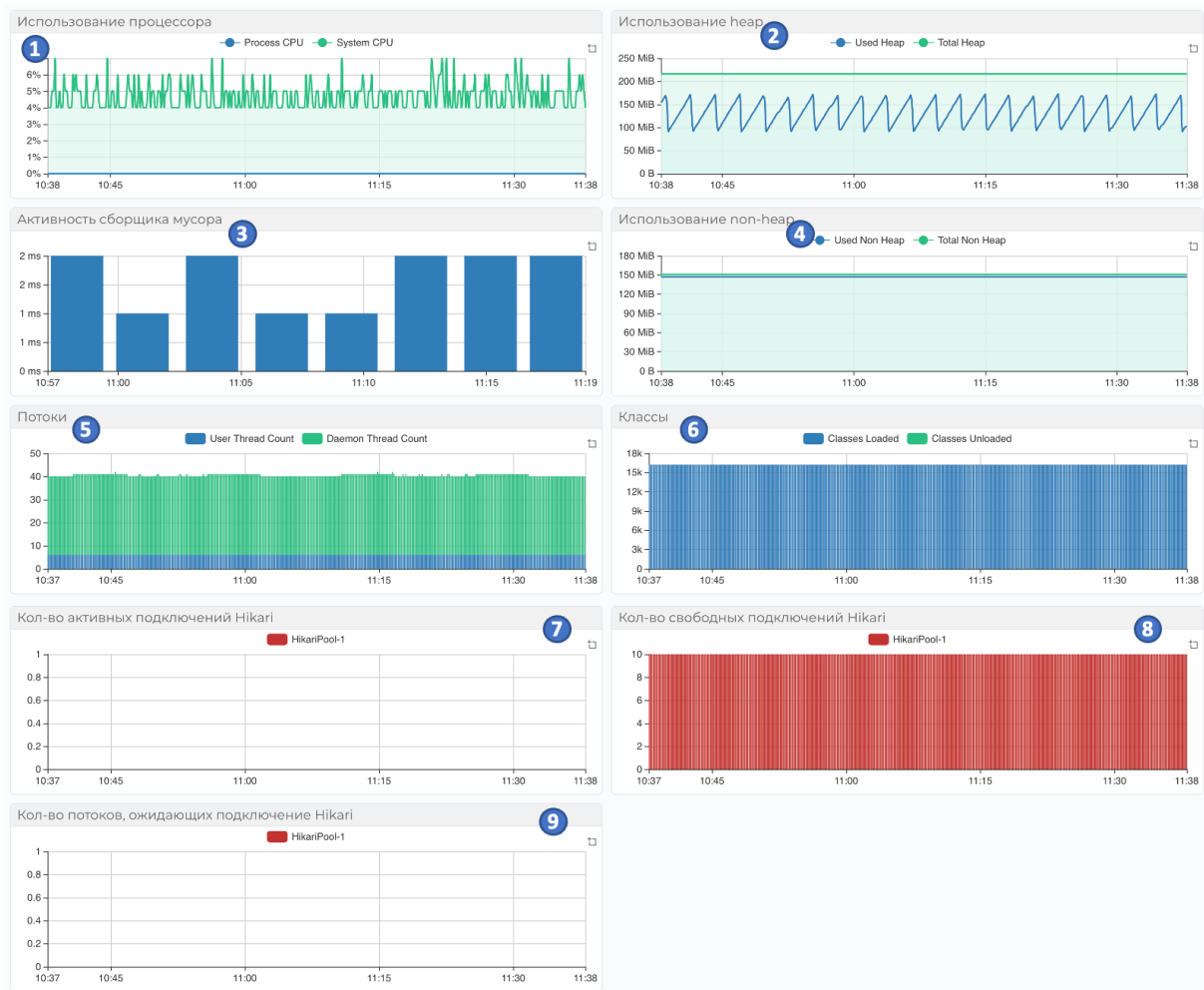
В данном разделе представлены следующие виджеты:

1. Информация о сервисе: кол-во трейсов, среднее время выполнения, медианное время выполнения, максимальное время выполнения 95% трейсов;
2. График с указанием числа трейсов, трейсов с ошибками, процентным соотношением числа трейсов с ошибками к общему числу трейсов;
3. График с указанием числа трейсов с 400-ыми и 500-ыми ошибками и процентным соотношением числа таких трейсов к общему числу трейсов;
4. Карта сервисов, показывающая взаимосвязи между текущим сервисом и сервисами, очередями сообщений и базами данных с указанием количества запросов между сервисами в минуту и средним временем выполнения вызовов;
5. Список эндпоинтов сервиса с указанием числа их выполнения (поддерживается переход на вкладку с детализацией по эндпоинту);
6. Список эндпоинтов сервиса с указанием числа их ошибочных выполнений (поддерживается переход на вкладку с детализацией по эндпоинту);
7. Список эндпоинтов сервиса с указанием среднего времени выполнения (поддерживается переход на вкладку с детализацией по эндпоинту);
8. Распределение времени по эндпоинтам внутри сервиса, эндпоинтам других сервисов и баз данных, с которыми взаимодействуют эндпоинты текущего сервиса;
9. Список SQL запросов, выполненных текущим сервисом, с указанием числа их выполнения (поддерживается переход на вкладку с детализацией по запросу);
10. Список SQL запросов, выполненных текущим сервисом, с указанием суммарного времени их выполнения (поддерживается переход на вкладку с детализацией по запросу);
11. Список SQL запросов, выполненных текущим сервисом, с указанием среднего времени их выполнения (поддерживается переход на вкладку с детализацией по запросу);
12. Таблица с самыми медленными трейсами данного сервиса, где указано время начала трейса, url трейса (поддерживается переход на вкладку с детализацией по трейсу), наименование эндпоинта (поддерживается переход на вкладку с детализацией по эндпоинту), длительность трейса (поддерживается переход на вкладку с деревом профилирования трейса), имя узла, пользователь, метод http, статус http.

Детализация по эндпоинту



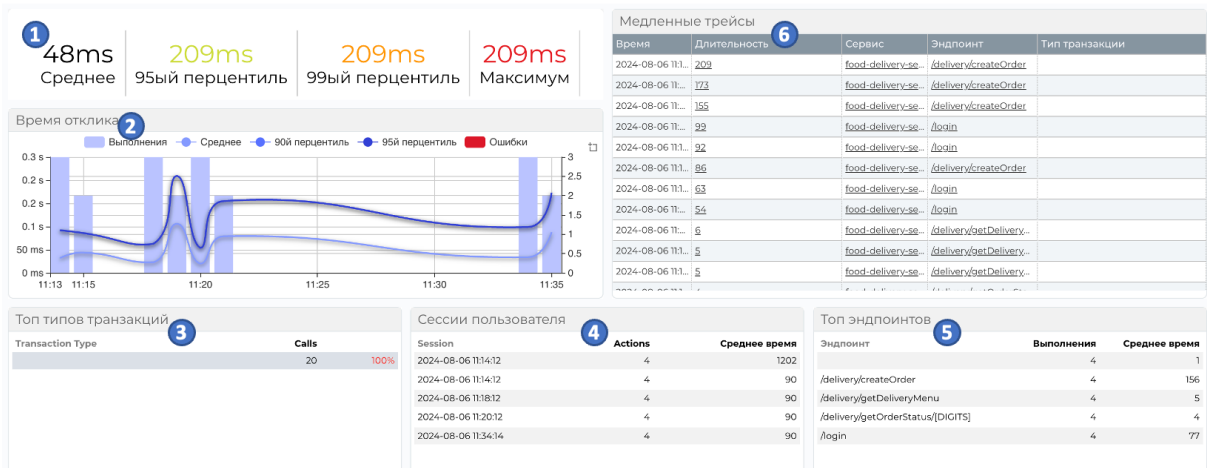
Детализация по инстансу



В данном разделе представлены следующие виджеты:

1. График с загрузкой процессора в процессе и на узле;
2. График с использованием памяти;
3. График с длительностью работы сборщика мусора;
4. График с использованием памяти вне кучи (non-heap);
5. График с кол-вом потоков, работающих в фоне и не в фоне;
6. График с кол-вом загруженных и выгруженных классов;
7. График с кол-вом активных подключений к базе данных;
8. График с кол-вом простаивающих подключений к базе данных;
9. График с кол-вом потоков, ожидающих подключения к базе данных.

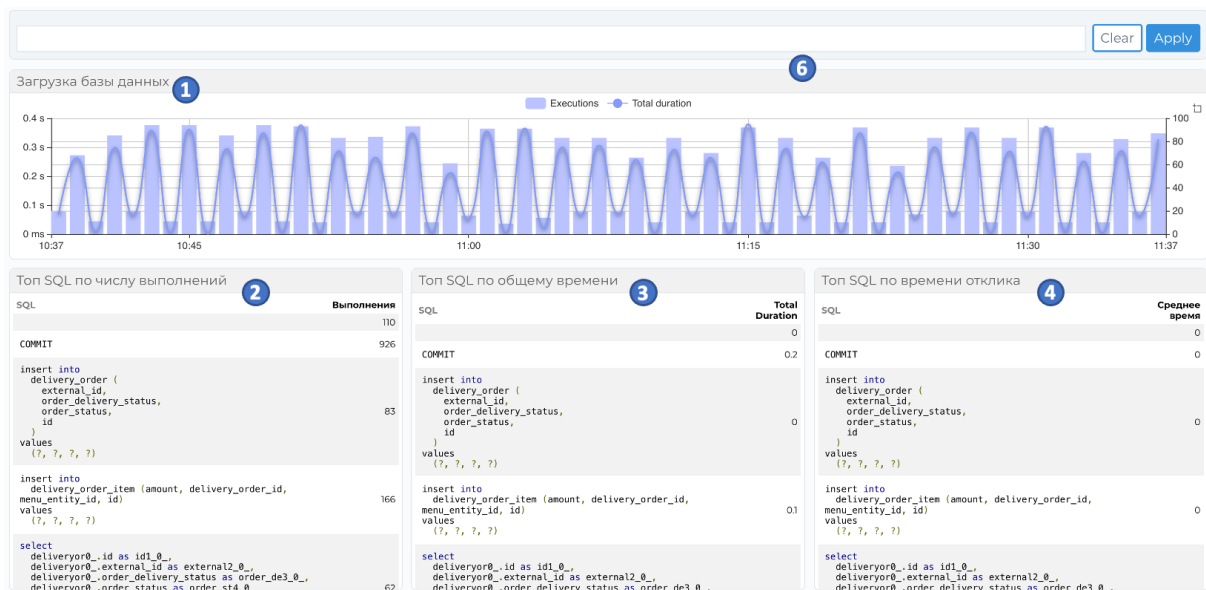
Детализация по пользователю



В данном разделе представлены следующие виджеты:

- 1. Информация о работе пользователя: среднее время выполнения трейсов, максимальное время выполнения 95% трейсов, максимальное время выполнения 99% трейсов, максимальное время выполнения;
- 2. График с указанием числа трейсов, трейсов с ошибками, процентным соотношением числа трейсов с ошибками к общему числу трейсов;
- 3. Список типов транзакций с указанием числа их выполнения;
- 4. Список сессий пользователя с указанием числа действий и длительностью сессии;
- 5. Список эндпоинтов пользователя с указанием числа их выполнения и средним временем выполнения;
- 6. Таблица с самыми медленными трейсами пользователя, где указано время выполнения трейса, длительность (поддерживается переход на вкладку с деревом профилирования трейса), наименование эндпоинта (поддерживается переход на вкладку с детализацией по эндпоинту), наименование сервиса (поддерживается переход на вкладку с детализацией по сервису), тип транзакции и http статус.

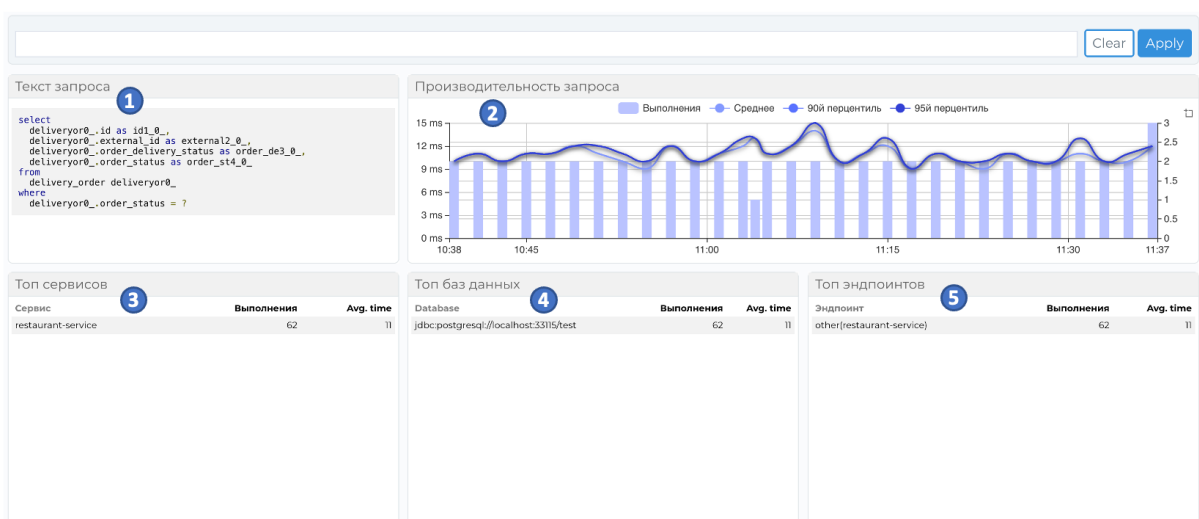
Детализация по базе данных



В данном разделе представлены следующие виджеты:

1. График загрузки базы данных, где представлено число запросов в единицу времени и суммарное время выполнения запросов в единицу времени;
2. Список SQL запросов, выполненных в текущий базе данных, с указанием числа их выполнения (поддерживается переход на вкладку с детализацией по запросу);
3. Список SQL запросов, выполненных в текущий базе данных, с указанием суммарного времени их выполнения (поддерживается переход на вкладку с детализацией по запросу);
4. Список SQL запросов, выполненных в текущий базе данных, с указанием среднего времени их выполнения (поддерживается переход на вкладку с детализацией по запросу).

Детализация по SQL запросу

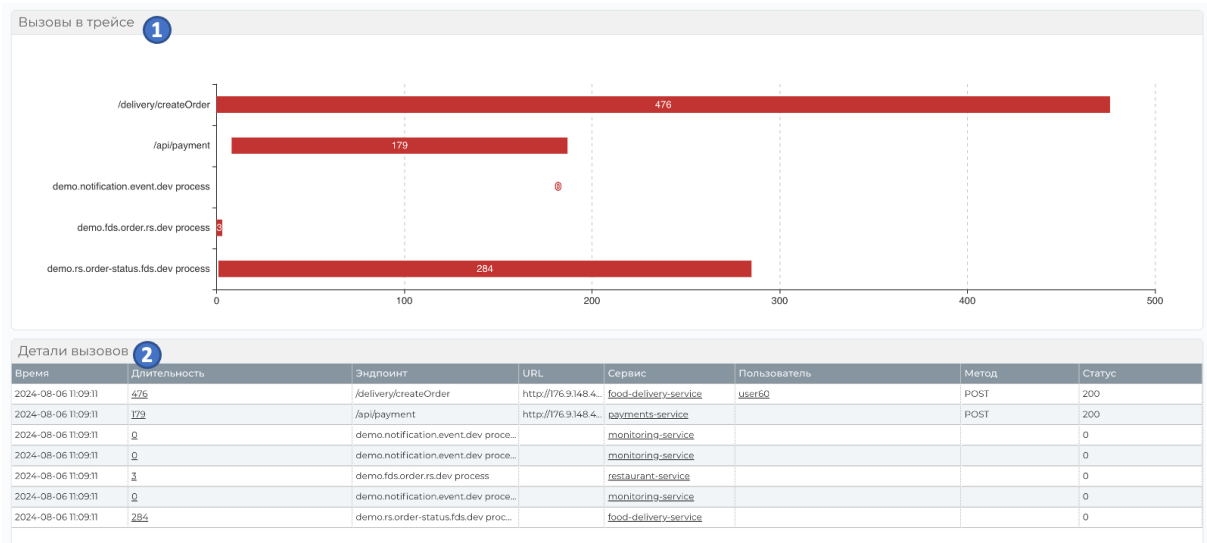


В данном разделе представлены следующие виджеты:

1. Текст запроса;

2. График с указанием числа выполнений этого запроса и средним временем выполнения запроса;
3. Список сервисов, выполняющих данный запрос, с указанием числа выполнений запроса (поддерживается переход на вкладку с детализацией по сервису);
4. Список баз данных, выполняющих данный запрос, с указанием числа выполнений запроса (поддерживается переход на вкладку с детализацией по базе данных);
5. Список эндпоинтов, выполняющих данный запрос, с указанием числа выполнений запроса (поддерживается переход на вкладку с детализацией по эндпоинту).

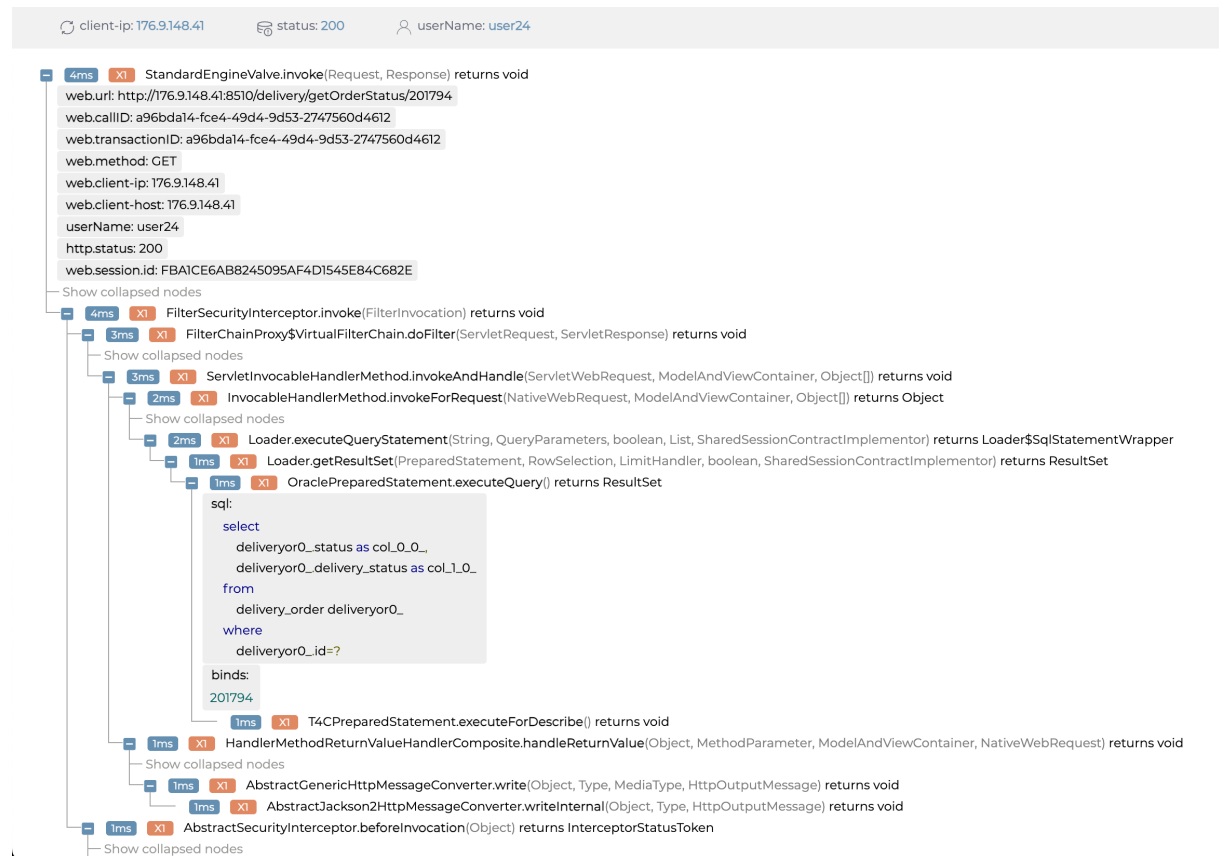
Детализация по трейсу



В данном разделе представлены следующие виджеты:

- 1. График, где представлены выполнения эндпоинтов, которые вызывались в рамках этого трейса с указанием длительности выполнения и времени начала вызова;
- 2. Таблица с указанием вызовов эндпоинтов внутри трейса с указанием времени начала вызова, длительности вызова (поддерживается переход на вкладку с деревом профилирования), наименования эндпоинта, url эндпоинта, сервиса (поддерживается переход на вкладку с детализацией по сервису), пользователя (поддерживается переход на вкладку с детализацией по пользователю), http метод, http статус.

Дерево профилирования



В данном разделе представлено дерево профилирования, где представлена информация о времени и кол-ве выполнений методов, выполняемых SQL запросах с параметрами выполнения, выполняемых вызовах других сервисов, работа с очередями сообщений.

Фильтрация трейсов

В системе доступен язык выражений, позволяющий выполнять фильтрацию. Ниже описаны доступные для фильтрации поля и примеры их использования:

Для поиска доступны следующие поля:

1. **serviceName** (строка) - название сервиса
2. **appName** (строка) - название приложения (окружения)
3. **topMethodName** (строка) – полное название метода, с которого начинается спан
4. **threadName** (строка) – название потока, в котором выполнялся спан
5. **duration** (число) – длительность спана в миллисекундах
6. **webUrl** (строка) – URL спана (для http спанов)
7. **webQuery** (строка) – параметры http спана
8. **httpStatus** (строка) – http status
9. **httpMessage** (строка) – http сообщение
10. **webException** (строка) – текст ошибки
11. **userName** (строка) – имя пользователя
12. **param** (вложенная структура):
 - key** (строка) – ключ;
 - value** (строка) - значение.

Параметры событий вызова, например: отправка сообщения в топик (key = kafka.send, value = <название топика>), чтение из топика (key = kafka.poll, value = <название топика>), проходящие через MDC атрибуты (например, key = X-B3-TraceId, value = <айди трейса>).

13. **timedParam** (вложенная структура):
 - name** (строка) – название события (например, sql, kafka.send, kafka.poll);
 - duration** (целое) – длительность события (реализовано для sql событий);
 - extraLong** (целое) – дополнительное поле (например, для sql хранится айди запроса);
 - extraValue** (строка) – дополнительное поле (например, для событий SQL хранится URL до СУБД, для Kafka событий – название топика).

Параметры событий вызова с дополнительной информацией, например, идентификаторами SQL запросов (**extraLong**, соответствует полю sqlHash словаря SQL запросов) и мета-информацией (**extraValue**, для SQL запросов – это URL до БД, для работы Kafka – это название топика).

Нужно помнить, что вложенные структуры – это отдельные массивы и работать с ними надо как с массивами. Ниже примеры:

Найти все вызовы, где значение любого параметра – это c3623288b2f2ef0d:

```
arrayFilter(value -> value = 'c3623288b2f2ef0d', param.value) != []
```

Найти все вызовы, где значение параметра X-B3-TraceId равно d6c3b85aa9156c76:

```
arrayFilter((key, value) -> key = 'X-B3-TraceId' and value = 'd6c3b85aa9156c76', param.key, param.value) != []
```

Найти все вызовы, где была отправка в топик order-topic:

```
arrayFilter((key, value) -> key = 'kafka.send' and value = order-topic', param.key, param.value)
!= []
```

Найти все вызовы, где было чтение из топика order-topic:

```
arrayFilter((key, value) -> key = 'kafka.poll' and value = order-topic', param.key, param.value)
!= []
```

Найти все вызовы, где был запрос, содержащий подстроку update:

```
hasAny(timedParam.extraLong, (select groupArray(sqlHash) from sql_dict_default where
sqlText like '%update%'))
```

Найти все вызовы, где была работа с СУБД PROD:

```
arrayFilter(value -> value like '%PROD%', timedParam.extraValue) != []
```

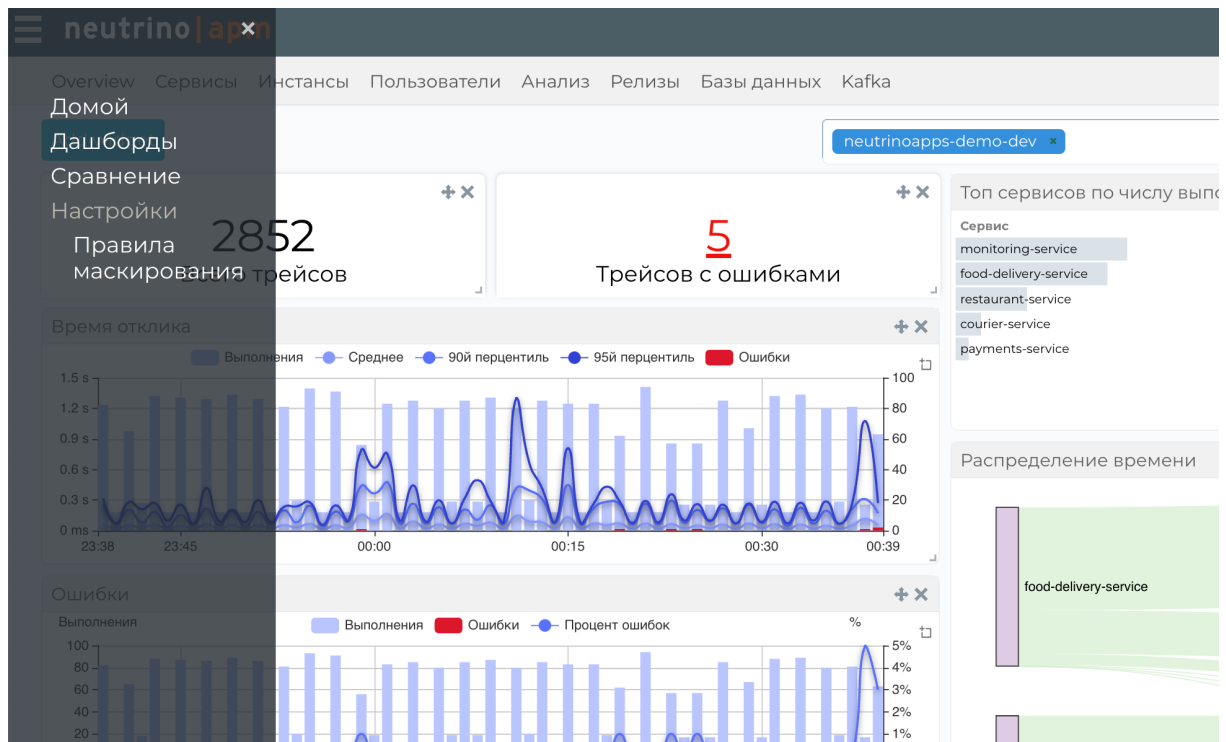
Найти все вызовы, где выполнялись запросы, содержащие подстроку update и выполнявшие дольше 10 миллисекунд:

```
arrayFilter((hash, duration) -> has((select groupArray(sqlHash) from sql_dict_default where
sqlText like '%update%'), hash) and duration > 10, timedParam.extraLong,
timedParam.duration) != []
```

Дашборды

Дашборды – это визуальное представление данных, разработанное для быстрого анализа информации. Дашборды состоят из различных Виджетов – объектов с графическим представлением конкретных данных, которые можно добавлять, редактировать, располагать, удалять или изменять. Neutrino APM позволяет пользователям настраивать несколько Дашбордов.

Дашборды, созданные пользователем, доступны для использования в Меню.



В данном меню представлена таблица со всеми Дашбордами, созданными пользователем в системе.

Дашборды			
Дашборды			
Name	Width	Actions	
Test	10	Edit	Delete

Создание Дашборда.

Для создание нового Дашборда, необходимо нажать «Добавить»

Далее необходимо указать:

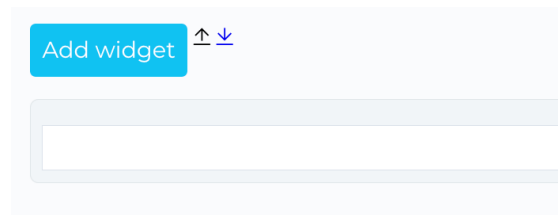
- Наименование Дашборда – Name. Задается уникальное имя Дашборда.
- Ширину Дашборда – Width.

Ширина Дашборда является ключевым параметром, подлежащим настройке с учетом разрешения экрана, на котором осуществляется его отображение. Данная характеристика напрямую влияет на визуальное восприятие и эффективность

использования Дашборда. Рекомендуется учитывать разрешение экрана в процессе определения ширины Дашборда. Для обеспечения оптимального пользовательского опыта и избежания горизонтальной прокрутки рекомендуется установка ширины Дашборда таким образом, чтобы он полностью адаптировался к разрешению экрана.



Далее созданный Дашборд необходимо наполнить Виджетами. Нажимаем кнопку «Add widget»



Удаление Дашборда

Для удаления Дашборда необходимо перейти на список Дашбордов, найти необходимый и нажать Delete.

Настройка агента

Настройки агента содержатся в файле `neutrino.config`. Ниже описаны ключевые параметры:

1. `ClientId` – идентификатор клиента;
2. `LicenceId` – идентификатор лицензии;
3. `AppName` – наименование приложения;
4. `ServiceName` – наименование сервиса;
5. `ServerAddress` – адрес, на который будет отправлять собранная диагностическая информация.