

Документ-приложение: Мониторинг производительности PostgreSQL в Grafana

Оглавление

1. Введение

2. Архитектура мониторинга

3. Обзор дашборда

3.1 Общая производительность базы данных

- 3.1.1 Транзакции в секунду
- 3.1.2 Активные подключения к базе данных

3.2 Анализ SQL-запросов (**pg_stat_statements**)

- 3.2.1 Топ-10 запросов по количеству вызовов
- 3.2.2 Топ-10 запросов по общему времени выполнения
- 3.2.3 Топ-10 самых медленных запросов (среднее время)

3.3 Таблица детализации SQL-запросов

- 3.3.1 Детализация SQL-запросов

4. Переменные шаблонов (Templating)

5. Использование дашборда для оптимизации

6. Заключение

1. Введение

Настоящий документ описывает дашборд Grafana, предназначенный для комплексного мониторинга производительности баз данных PostgreSQL. Дашборд разработан для предоставления оперативной информации о ключевых метриках СУБД, позволяя выявлять узкие места, анализировать поведение SQL-запросов и принимать обоснованные решения для оптимизации.

Основной целью дашборда является агрегация и визуализация данных, собираемых Prometheus-экспортером PostgreSQL, что обеспечивает глубокое понимание состояния и производительности базы данных.

2. Архитектура мониторинга

Система мониторинга построена на следующей архитектуре:

- **PostgreSQL:** Отслеживаемая база данных.
- **pg_stat_statements:** Расширение PostgreSQL, собирающее статистику по выполненным SQL-запросам.
- **Prometheus Exporter (postgres_exporter):** Агент, который подключается к PostgreSQL, собирает метрики (включая данные из pg_stat_statements) и предоставляет их в формате, понятном Prometheus. В вашей конфигурации используется prometheuscommunity/postgres-exporter с следующими параметрами:
 - **DATA_SOURCE_NAME:**
"postgres://postgres:postgres@192.1xx.xx.xx:5432/postgres?sslmode=disable" - Определяет строку подключения к базе данных PostgreSQL, указывая пользователя (postgres), пароль (postgres), IP-адрес (192.1xx.xx.xx), порт (5432) и имя базы данных (postgres). sslmode=disable отключает SSL-шифрование для подключения.
 - **PG_EXPORTER_EXTEND_QUERY_PATH:**
"/etc/postgres_exporter/queries.yaml" - Указывает путь к файлу с дополнительными пользовательскими запросами, которые экспортер будет выполнять для сбора метрик.
 - **PG_EXPORTER_DISABLE_DEFAULT_METRICS:** "false" - Означает, что стандартные метрики, предоставляемые экспортером по умолчанию, **не будут** отключены.
 - **PG_EXPORTER_DISABLE_SETTINGS_METRICS:** "false" - Означает, что метрики настроек PostgreSQL (например, max_connections) **не будут** отключены.
 - **Порты:** 9xxx:91xx - Экспортер слушает на порту 91xx внутри контейнера,

который маппируется на порт 9xxx на хостовой машине, откуда Prometheus будет собирать метрики.

- **Volumes:**

/home/user/postgres_queries.yaml:/etc/postgres_exporter/queries.yaml:ro - Монтирует локальный файл postgres_queries.yaml с хоста (/home/user/) в контейнер (/etc/postgres_exporter/queries.yaml) в режиме "только для чтения" (ro), предоставляя экспортеру доступ к пользовательским запросам.

- **Сеть:** monitor-net - Контейнер подключен к сети monitor-net, что позволяет ему взаимодействовать с другими компонентами мониторинга (например, Prometheus) в той же сети.

```
1 postgres-exporter:
2   image: prometheuscommunity/postgres-exporter
3   container_name: postgres-exporter
4   environment:
5     DATA_SOURCE_NAME: "postgresql://postgres:postgres@192.16x.xx.xx:5432/postgres?sslmode=disable"
6     PG_EXPORTER_EXTEND_QUERY_PATH: "/etc/postgres_exporter/queries.yaml"
7     PG_EXPORTER_DISABLE_DEFAULT_METRICS: "false"
8     PG_EXPORTER_DISABLE_SETTINGS_METRICS: "false"
9   ports:
10     - "9xxx:91xx"
11   volumes:
12     - /home/user/postgres_queries.yaml:/etc/postgres_exporter/queries.yaml:ro
13   networks:
14     - monitor-net
15   deploy:
16     resources:
17       limits:
18         memory: "128m"
19         cpus: "0.05"
20   restart: unless-stopped
```

- **Prometheus:** Система сбора и хранения временных рядов, которая периодически опрашивает экспортер для получения метрик.
- **Grafana:** Платформа для визуализации данных, которая подключается к Prometheus и отображает метрики в виде интерактивных дашбордов.

3. Обзор дашборда

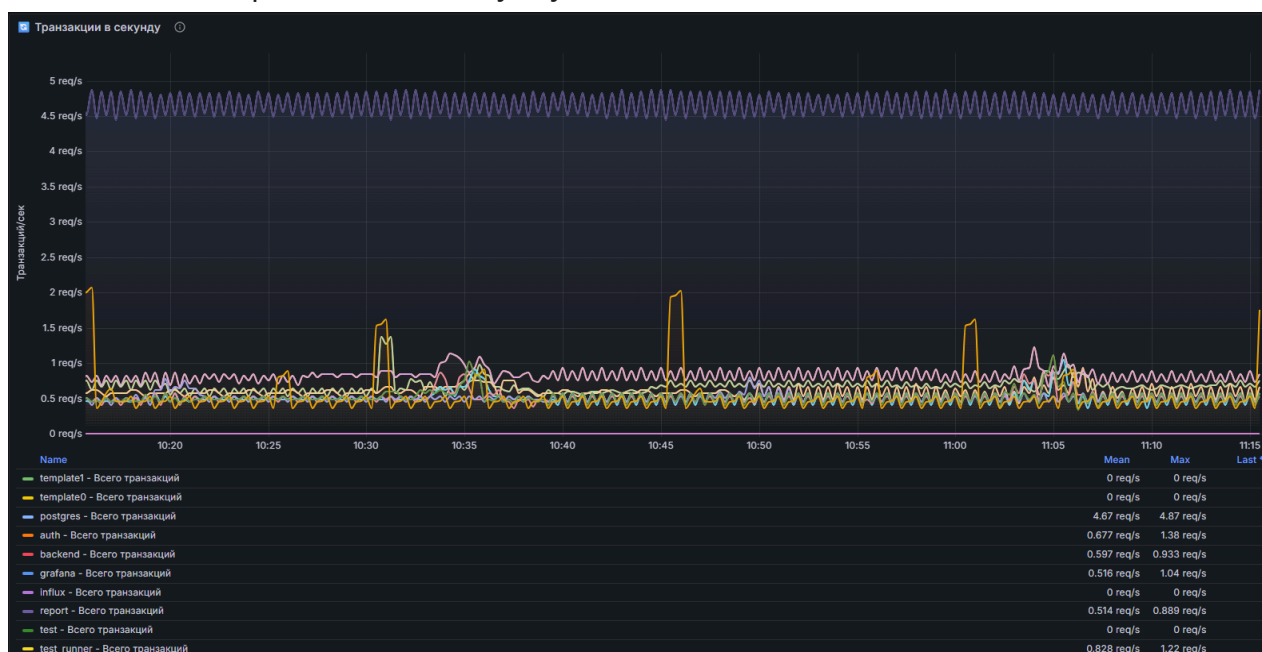
Дашборд состоит из нескольких ключевых секций (строк), каждая из которых объединяет панели, посвященные определенным аспектам производительности базы данных.

3.1. Общая производительность базы данных (Row ID: 10)

Эта секция предоставляет высокоуровневый обзор состояния базы данных.

3.1.1. Транзакции в секунду (Panel ID: 5)

Рис. 1: Панель "Транзакции в секунду"



- **Описание:** Эта панель отображает общее количество транзакций (коммитов и откатов) в секунду. Высокие значения могут указывать на активную нагрузку на базу данных.
- **Отлавливает:**
 - **Общую активность:** Позволяет быстро оценить текущий уровень нагрузки на базу данных.
 - **Пики нагрузки:** Выявляет моменты повышенной активности, которые могут требовать дальнейшего анализа.
 - **Соотношение коммитов/откатов:** Отдельные графики для коммитов и откатов помогают понять успешность транзакций. Высокий процент откатов может указывать на проблемы в приложении или конкуренцию за ресурсы.

ресурсы.

- **Метрики Prometheus:**

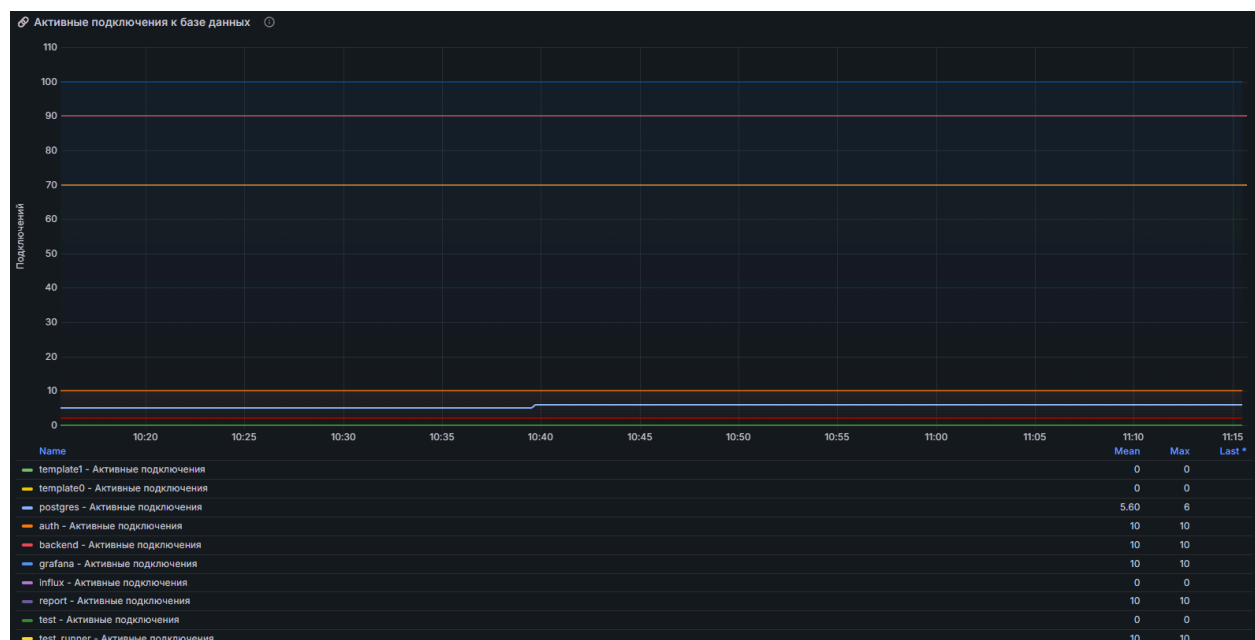
- `rate(pg_stat_database_xact_commit[1m]) + rate(pg_stat_database_xact_rollback[1m])`: Общее количество транзакций (коммиты + откаты) в секунду за последнюю минуту.
- `rate(pg_stat_database_xact_commit[1m])`: Количество коммитов в секунду за последнюю минуту.
- `rate(pg_stat_database_xact_rollback[1m])`: Количество откатов в секунду за последнюю минуту.

- **Пороги (Thresholds):**

- Зеленый: До 100 транзакций/сек
- Желтый: От 100 до 500 транзакций/сек (потенциальная нагрузка)
- Красный: Свыше 500 транзакций/сек (высокая нагрузка, требует внимания)

3.1.2. Активные подключения к базе данных (Panel ID: 4)

Рис. 2: Панель "Активные подключения к базе данных"



- **Описание:** Эта панель отслеживает количество активных подключений к базе данных и сравнивает их с максимальным разрешенным количеством подключений. Превышение лимитов может привести к отказам в обслуживании.
- **Отлавливает:**
 - **Проблемы с пулом подключений:** Помогает выявить ситуации, когда приложение создает слишком много подключений.

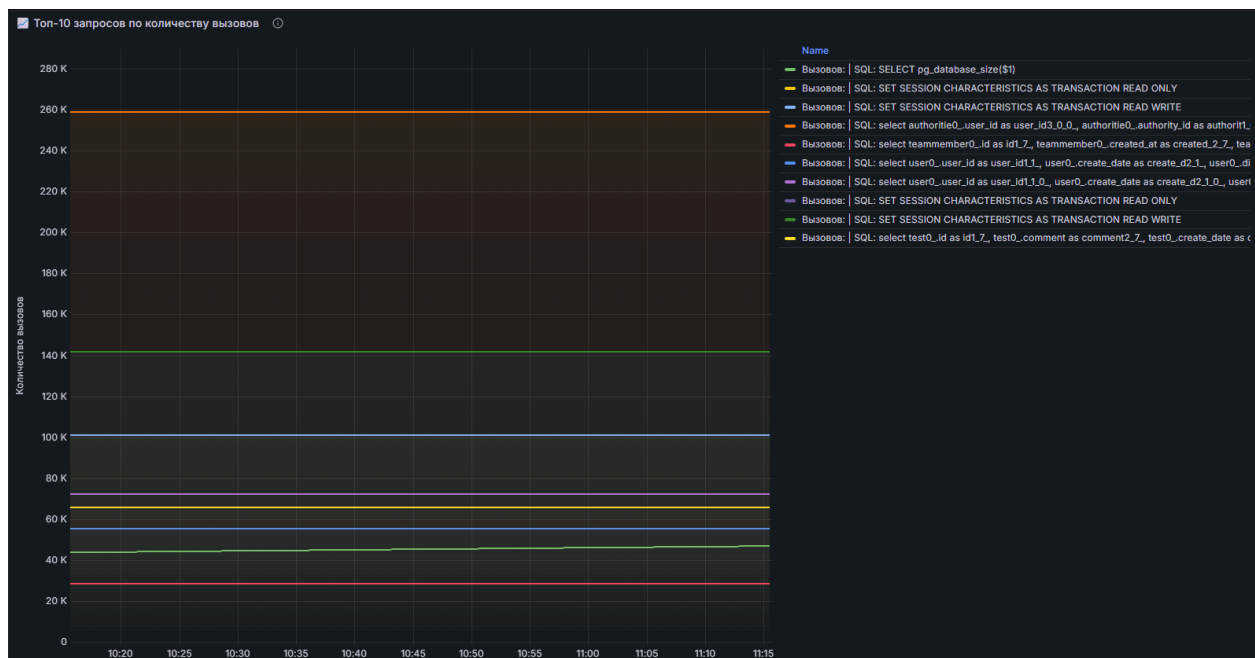
- **Исчерпание лимитов:** Сигнализирует о приближении к максимальному количеству подключений, что может вызвать ошибки "too many connections".
- **Недостаточность ресурсов:** Указывает на необходимость увеличения max_connections или оптимизации работы с подключениями.
- **Метрики Prometheus:**
 - pg_stat_database_numbackends: Текущее количество активных подключений.
 - pg_settings_max_connections: Максимальное количество разрешенных подключений.
- **Пороги (Thresholds):**
 - Зеленый: До 70% от максимального количества подключений
 - Оранжевый: От 70% до 90%
 - Красный: Свыше 90% (критический уровень, возможно исчерпание подключений)

3.2. Анализ SQL запросов (pg_stat_statements) (Row ID: 11)

Эта секция фокусируется на детальном анализе производительности отдельных SQL-запросов с использованием данных из расширения pg_stat_statements.

3.2.1. Топ-10 запросов по количеству вызовов (Panel ID: 2)

Рис. 3: Панель "Топ-10 запросов по количеству вызовов"



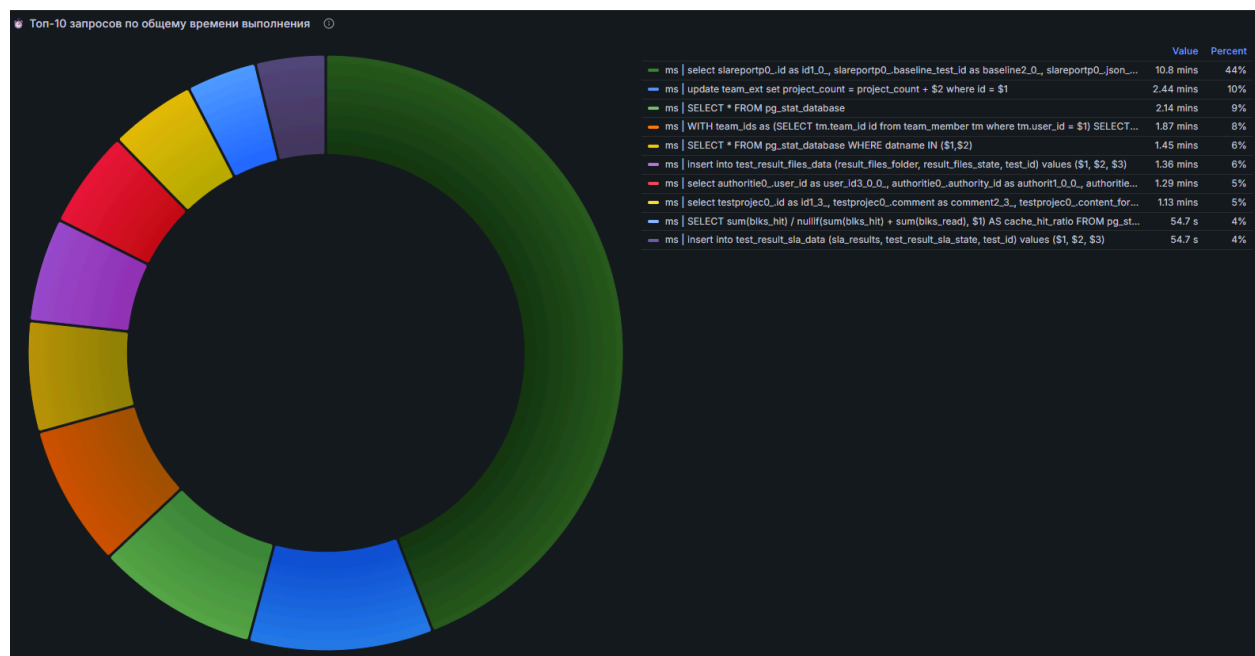
- **Описание:** Отображает 10 наиболее часто выполняемых SQL-запросов.

Используется для выявления "горячих точек" в приложении, то есть запросов, которые генерируют наибольшую нагрузку из-за частоты выполнения.

- **Отлавливает:**
 - **Часто вызываемые запросы:** Помогает определить, какие запросы выполняются чаще всего.
 - **Потенциал для кеширования:** Если часто вызываемый запрос возвращает одни и те же данные, его можно кешировать.
 - **Неэффективные циклы:** Может указывать на N+1 проблемы или другие неэффективные паттерны доступа к данным.
- **Метрика Prometheus:** `topk(10, pg_stat_statements_calls)`: Топ-10 запросов по общему количеству вызовов.
- **Пороги (Thresholds):**
 - Зеленый: До 1000 вызовов
 - Оранжевый: От 1000 до 10000 вызовов (требуется внимания)
 - Красный: Свыше 10000 вызовов (высокая частота, критично)

3.2.2. Топ-10 запросов по общему времени выполнения (Panel ID: 1)

Рис. 4: Панель "Топ-10 запросов по общему времени выполнения"



- **Описание:** Круговая диаграмма, показывающая распределение общего времени выполнения среди 10 самых "дорогих" SQL-запросов. Запросы с наибольшим общим временем выполнения требуют первоочередной оптимизации, так как они потребляют наибольшее количество ресурсов СУБД.

- **Отлавливает:**
 - **Самые ресурсоемкие запросы:** Выявляет запросы, которые дольше всего удерживают ресурсы базы данных.
 - **Кандидаты для оптимизации:** Эти запросы являются лучшими кандидатами для детального анализа и оптимизации (например, с помощью EXPLAIN ANALYZE).
- **Метрика Prometheus:** topk(10, pg_stat_statements_total_time): Топ-10 запросов по общему времени выполнения.

3.2.3. 🚩 Топ-10 самых медленных запросов (среднее время) (Panel ID: 3)

Рис. 5: Панель "Топ-10 самых медленных запросов (среднее время)"



- **Описание:** Отображает 10 SQL-запросов с наибольшим средним временем выполнения. Эти запросы являются критичными для производительности системы, так как они могут вызывать задержки для конечных пользователей.
- **Отлавливает:**
 - **Медленные запросы:** Определяет запросы, которые выполняются медленно в среднем, даже если они вызываются не очень часто.
 - **Проблемы с индексами:** Часто медленные запросы указывают на отсутствие или неэффективность индексов.
 - **Сложные запросы:** Может сигнализировать о слишком сложных или неоптимальных запросах.
- **Метрика Prometheus:** topk(10, pg_stat_statements_mean_time): Топ-10 запросов по среднему времени выполнения.
- **Пороги (Thresholds):**

- Зеленый: До 100 мс
- Желтый: От 100 до 500 мс (желательно оптимизировать)
- Оранжевый: От 500 до 1000 мс (требуют внимания)
- Красный: Свыше 1000 мс (критичны)

3.3. Таблица детализации SQL запросов (Row ID: 12)

Эта секция предоставляет подробную таблицу со всеми метриками SQL-запросов для глубокого анализа.

3.3.1. Детализация SQL запросов (Panel ID: 6)

Рис. 6: Панель "Детализация SQL запросов"

SQL Query	Количество вызовов	Общее время (мс)	Среднее время (мс)	dbid	server
insert into grafana_user (login, password, user_id, id) values (\$1, \$2, \$3, \$4)	851			16386	192.168.14.80:5432
insert into grafana_user (login, password, user_id, id) values (\$1, \$2, \$3, \$4)		294		16386	192.168.14.80:5432
insert into grafana_user (login, password, user_id, id) values (\$1, \$2, \$3, \$4)			0.345	16386	192.168.14.80:5432
insert into project_version (comment, content_format, content_model_version, created_by_user_id, created_on, location, request_count, settings_id, test_project_id, test_type, total_duration, version_number, id) values (\$1, \$2,	4997			16392	192.168.14.80:5432

- **Описание:** Подробная таблица с метриками всех SQL-запросов, отслеживаемых pg_stat_statements. Это центральная панель для анализа и оптимизации.
- **Отлавливает:**
 - **Полный список запросов:** Предоставляет обзор всех отслеживаемых запросов и их метрик.
 - **Проблемы с отображением данных:** Благодаря использованию last_over_time и range: true для запросов Prometheus, эта таблица должна более стабильно отображать данные, даже если запросы не выполнялись в последние секунды. Это решает проблему "пропадающих" значений, о которой вы упоминали.
 - **Фильтрация и сортировка:** Позволяет пользователю сортировать по столбцам (например, по "Количество вызовов", "Общее время (мс)", "Среднее время (мс)") для быстрого поиска проблемных запросов.
 - **Детальный анализ:** Предоставляет SQL-текст запроса, который можно скопировать для дальнейшего анализа в инструментах типа pgAdmin или psql с использованием EXPLAIN ANALYZE.
- **Метрики Prometheus:**

- `last_over_time(pg_stat_statements_calls[$__range])`: Последнее значение количества вызовов для каждого запроса за выбранный временной диапазон.
 - `last_over_time(pg_stat_statements_total_time[$__range])`: Последнее значение общего времени выполнения для каждого запроса за выбранный временной диапазон.
 - `last_over_time(pg_stat_statements_mean_time[$__range])`: Последнее значение среднего времени выполнения для каждого запроса за выбранный временной диапазон.
- **Трансформации:**
 - **Merge**: Объединяет результаты из разных Prometheus-запросов в одну таблицу.
 - **Organize**: Переименовывает и исключает ненужные столбцы для более читабельного представления:
 - Value #A переименовано в "Количество вызовов"
 - Value #B переименовано в "Общее время (мс)"
 - Value #C переименовано в "Среднее время (мс)"
 - query переименовано в "SQL Query"
 - Исключены: Time, __name__, datid, datname, instance, job, queryid, userid (эти метки используются Prometheus для идентификации, но не всегда нужны в финальной таблице).
 - **Пороги (Thresholds):**
 - **Среднее время (мс):**
 - Зеленый: До 100 мс
 - Желтый: От 100 до 500 мс
 - Оранжевый: От 500 до 1000 мс
 - Красный: Свыше 1000 мс
 - **Количество вызовов:**
 - Зеленый: До 1000 вызовов
 - Желтый: От 1000 до 10000 вызовов
 - Красный: Свыше 10000 вызовов

4. Переменные шаблонов (Templating)

Дашборд использует переменную шаблона database для фильтрации данных по имени базы данных.

- **Имя:** database
- **Метка:** Database

- **Запрос:** `label_values(pg_stat_database_numbackends, datname)`: Получает все уникальные имена баз данных из метрики `pg_stat_database_numbackends`.
- **Функциональность:** Позволяет пользователю выбирать конкретную базу данных для мониторинга или просматривать данные по всем базам данных (All).

5. Использование дашборда для оптимизации

Этот дашборд является мощным инструментом для:

- **Оперативного мониторинга:** Быстрое выявление проблем с производительностью в режиме реального времени.
- **Идентификации проблемных запросов:** Панели "Топ-10" и "Детализация SQL запросов" помогают найти запросы, которые потребляют больше всего ресурсов или выполняются медленно.
- **Анализа тенденций:** Графики позволяют отслеживать изменения производительности с течением времени.
- **Принятия решений по оптимизации:** Основываясь на данных дашборда, можно принимать решения о добавлении индексов, переписывании запросов, изменении конфигурации PostgreSQL или масштабировании ресурсов.

6. Заключение

Представленный дашборд Grafana обеспечивает всесторонний обзор производительности PostgreSQL, делая процесс мониторинга и оптимизации более эффективным и прозрачным. Регулярный анализ данных, представленных на дашборде, является ключевым для поддержания высокой производительности и стабильности вашей базы данных.