|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №4**

по дисциплине «Технологии виртуализации клиент-серверных приложений»

**Студент группы** ИКБО-20-21 Мухаметшин А. Р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы**  Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Москва 2024

**ОГЛАВЛЕНИЕ**

[Цель работы 2](#_Toc179528546)

[Ход работы 5](#_Toc179528547)

[Выводы 16](#_Toc179528548)

[Ответы на вопросы к практической работе 17](#_Toc179528549)

[Список использованной литературы 19](#_Toc179528550)

# Цель работы

Docker Compose — это инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов. Docker Compose позволяет управлять набором контейнеров, каждый из которых представляет собой один сервис проекта. Управление включает в себя сборку, запуск с учетом зависимостей и конфигурацию. Конфигурация Docker Compose описывается в файле docker-compose.yml, лежащем в корне проекта.

Файл docker-compose.yml выглядит следующим образом:



Для получения дополнительной информации о файле создания вы можете изучить официальную документацию Docker.

Помимо сервисов на серверах необходимы системы мониторинга, чтобы осуществлять контроль над работоспособностью системы. Для этого используются различные виды систем мониторинга:

¾ Инфраструктурный мониторинг (Например Zabbix, Nagios)

¾ Мониторинг ошибок программных платформ (Sentry)

¾ Мониторинг производительности приложений (Prometheus)

¾ Мониторинг безопасности систем (Nesus, OpenVas)

¾ Сбор системных журналов (GrayLog) Zabbix — это универсальный инструмент мониторинга, способный отслеживать динамику работы серверов и сетевого оборудования, быстро реагировать на внештатные ситуации и предупреждать возможные проблемы с нагрузкой. Система мониторинга Zabbix может собирать статистику в указанной рабочей среде и действовать в определенных случаях заданным образом. У Zabbix есть 4 основных инструмента, с помощью которых можно мониторить определенную рабочую среду и собирать о ней полный пакет данных для оптимизации работы.

¾ Сервер

— ядро, хранящее в себе все данные системы, включая статистические, оперативные и конфигурацию. Дистанционно управляет сетевыми сервисами, оповещает администратора о существующих проблемах с оборудованием, находящимся под наблюдением.

¾ Прокси — сервис, собирающий данные о доступности и производительности устройств, который работает от имени сервера. Все собранные данные сохраняются в буфер и загружаются на сервер. Нужен для распределения нагрузки на сервер. Благодаря этому процессу можно уменьшить нагрузку на процессор и жесткий диск. Для работы прокси Zabbix отдельно нужна база данных.

¾ Агент — программа (демон), которая активно мониторит и собирает статистику работы локальных ресурсов (накопители, оперативная память, процессор и др.) и приложений.

¾ Веб-интерфейс — является частью сервера системы и требует для работы веб-сервер. Часто запускается на том же физическом узле, что и Zabbix.

Graylog — это платформа, которая позволяет легко управлять записями структурированных и неструктурированных данных. вместе с отладкой приложений. Он основан на Elasticsearch, MongoDB и Scala. Он имеет главный сервер, который принимает данные от своих клиентов, установленных на разных серверах, и веб-интерфейс, который отображает данные и позволяет работать с записями, добавленными основным сервером. Graylog эффективен при работе с необработанными строками (например, с системным журналом) - инструмент анализирует их на нужные нам структурированные данные.

Основное преимущество Graylog заключается в том, что он предоставляет единый идеальный экземпляр сбора журналов для всей системы.

Prometheus — система мониторинга. Основные преимущества — предоставление возможности создания гибких запросов к данным и хранение значений метрик в базе данных временных рядов, возможность автоматизации при администрировании. Разработана фондом облачных вычислений (Cloud Native Computing Foundation или CNCF). Для получения метрик с удаленных узлов используется метод pull (сервер сам забирает данные). На узлы для сбора информации устанавливаются экспортеры (exporter) — пакеты, получающие данные для операционной системы или конкретного сервиса. Существует большое количество уже написанных экспортеров для различных приложений. Также метрики могут собираться с помощью механизма push — для этого используется компонент pushgateway, который должен быть установлен дополнительно. Довольно часто Prometheus настраивают в связке с Grafana, которая позволяет визуализировать показания наших метрик. В Grafana для этого есть уже настроенный источник, таким образом, настройка выполняется из коробки. Довольно часто Prometheus настраивают в связке с Grafana, которая позволяет визуализировать показания наших метрик. В Grafana для этого есть уже настроенный источник, таким образом, настройка выполняется из коробки. Grafana — универсальная обертка для работы с аналитическими данными, которые хранятся в разных источниках. Она сама ничего не хранит и не собирает, а является лишь универсальным клиентом для систем хранения метрик.

# Ход работы

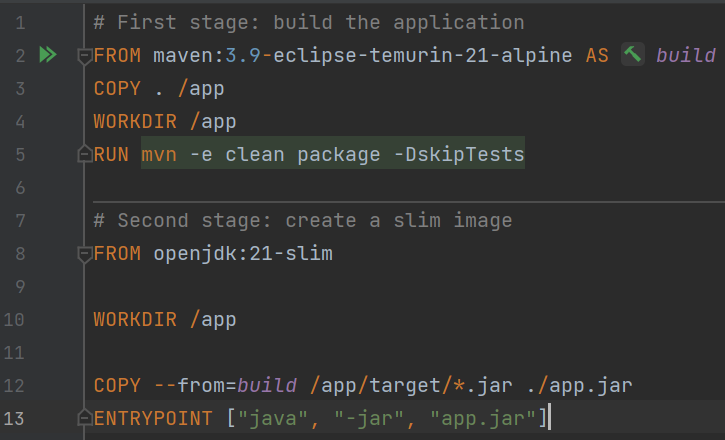


Рисунок 1 – Dockerfile приложения

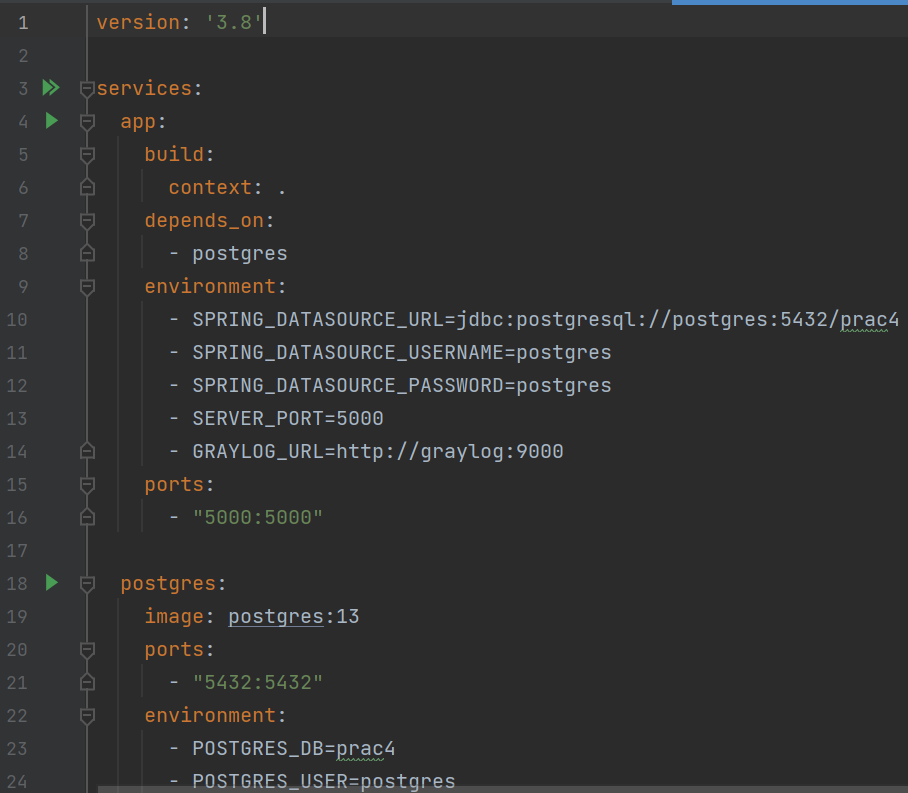


Рисунок 2 – Docker-compose файл

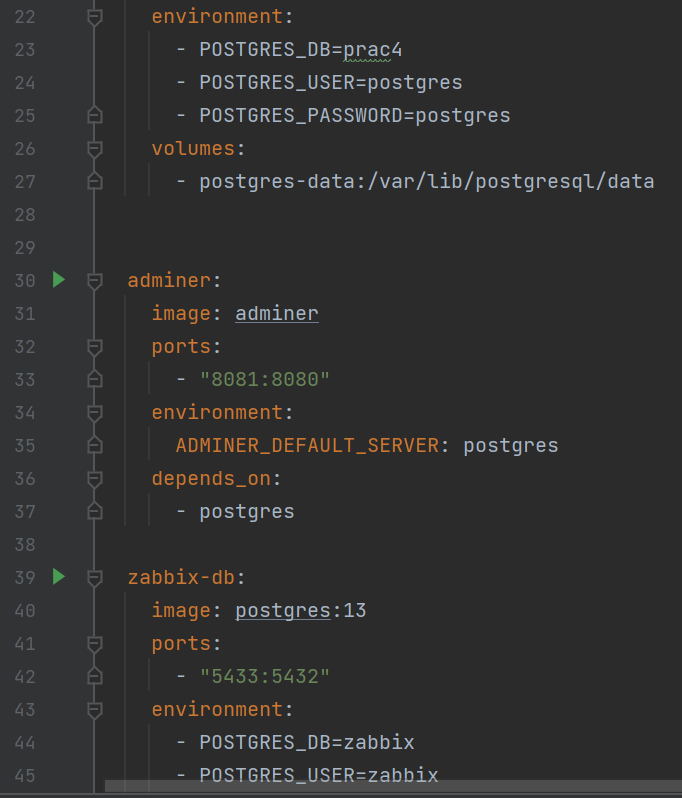


Рисунок 3 – Docker-compose файл

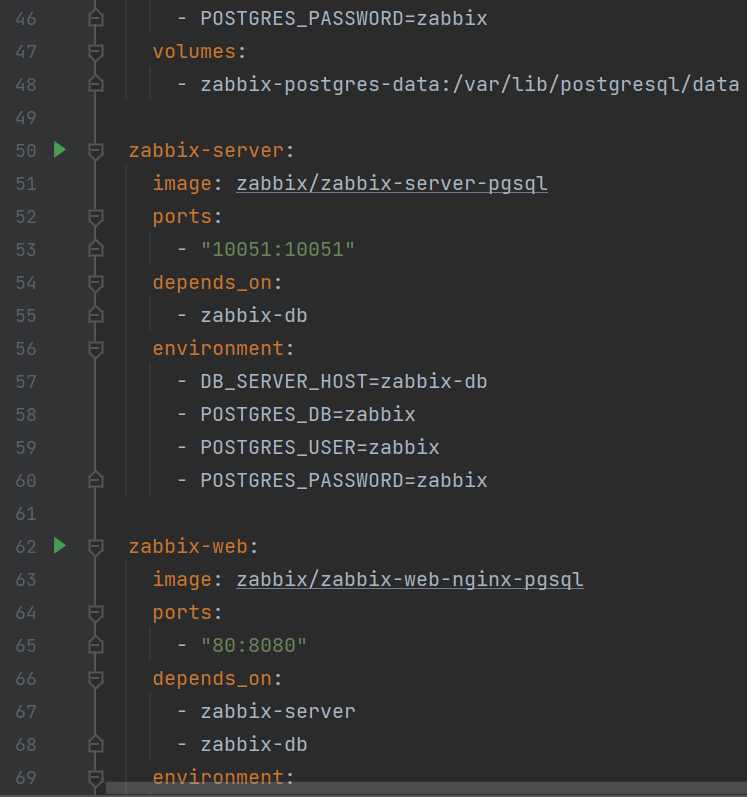


Рисунок 4 – Docker-compose файл

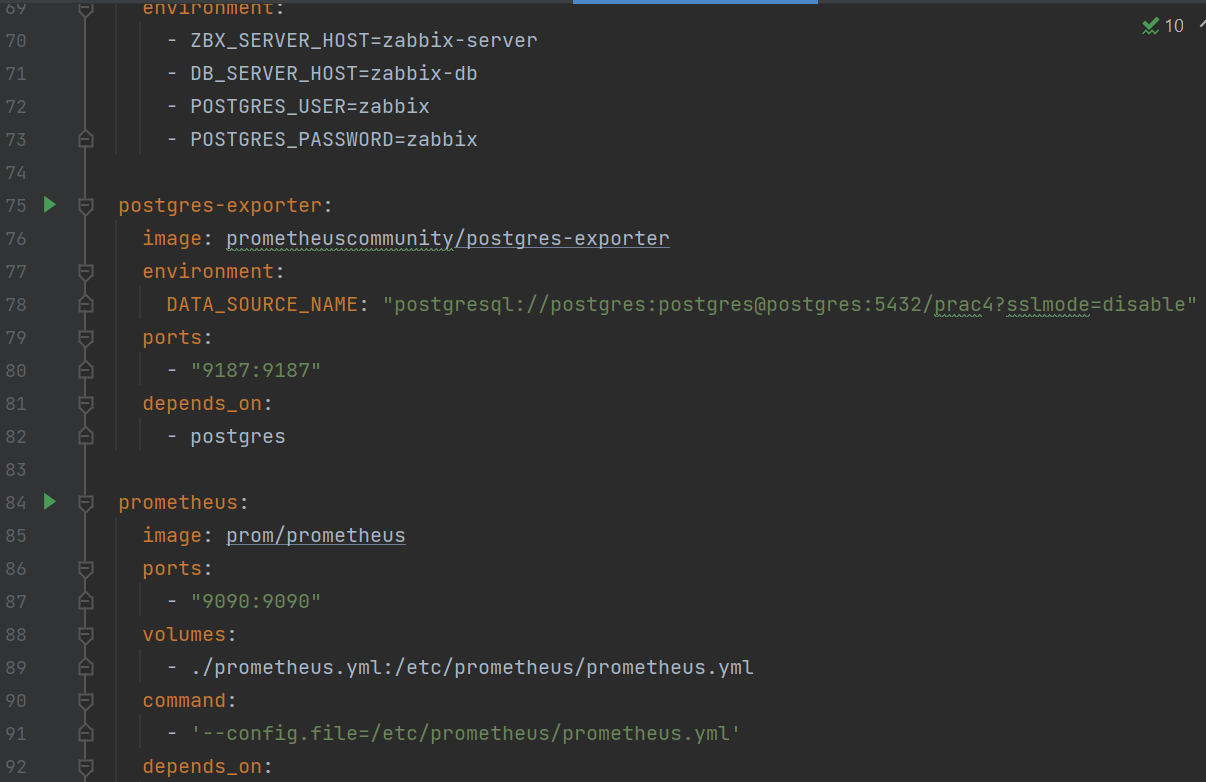


Рисунок 5 – Docker-compose файл

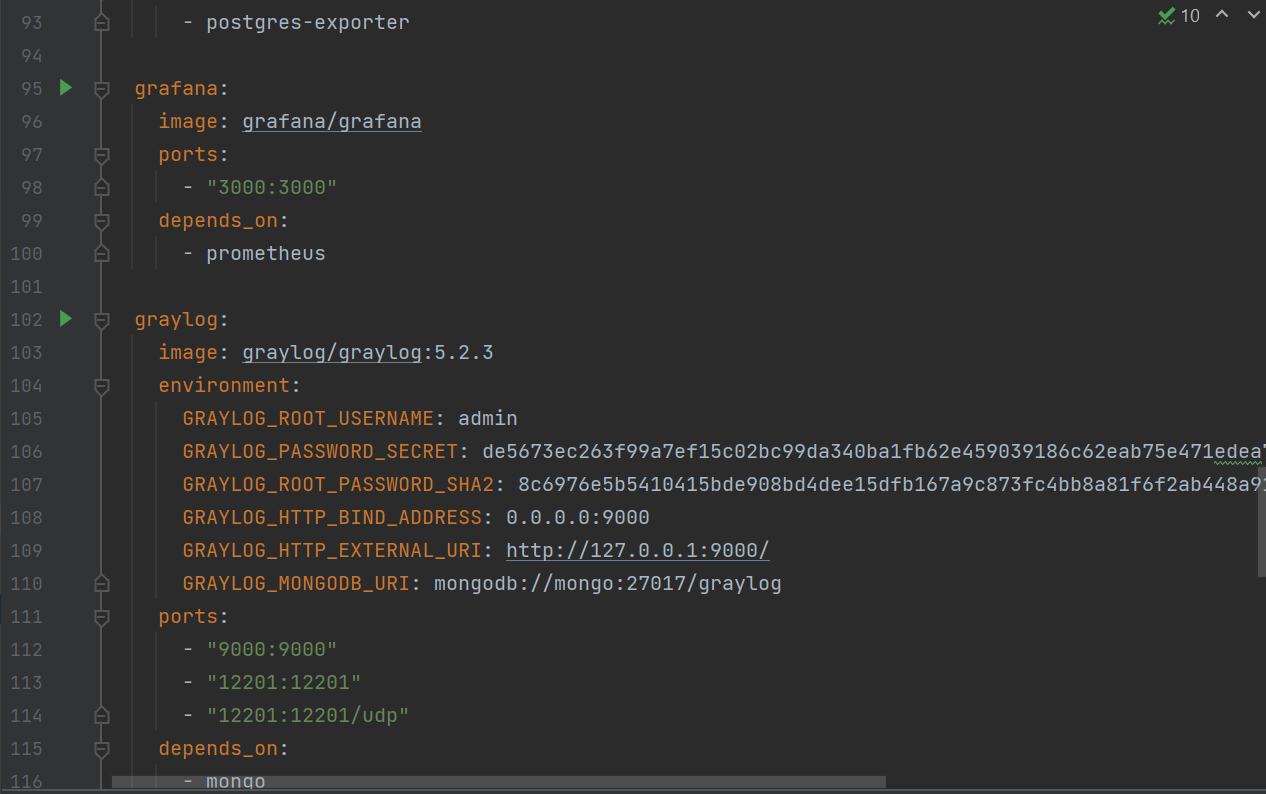


Рисунок 6 – Docker-compose файл

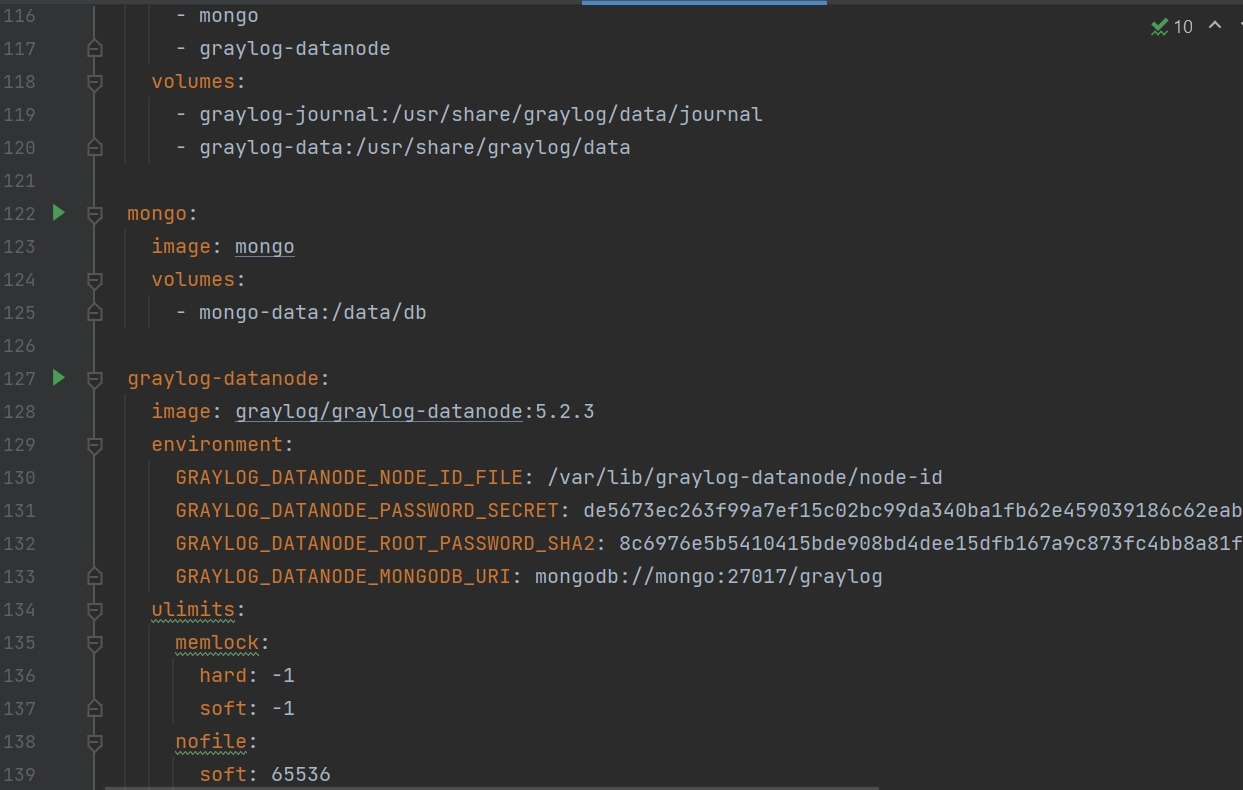


Рисунок 7 – Docker-compose файл

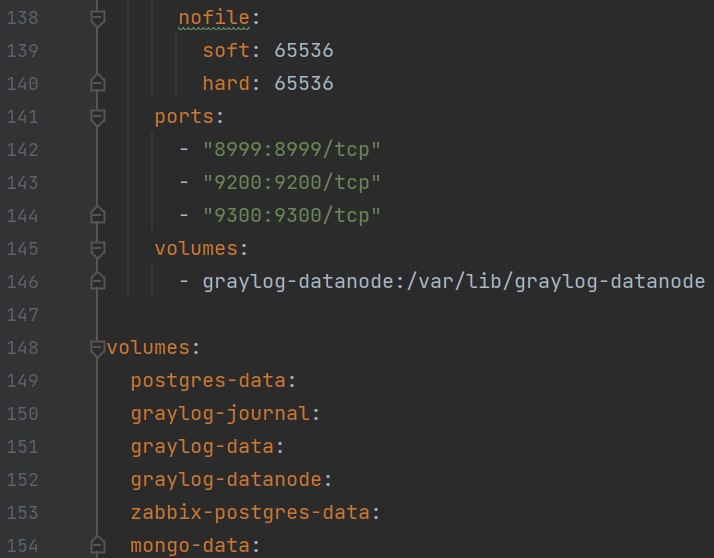


Рисунок 8 – Docker-compose файл

Рисунок 9 – GrayLog собирающий метрики

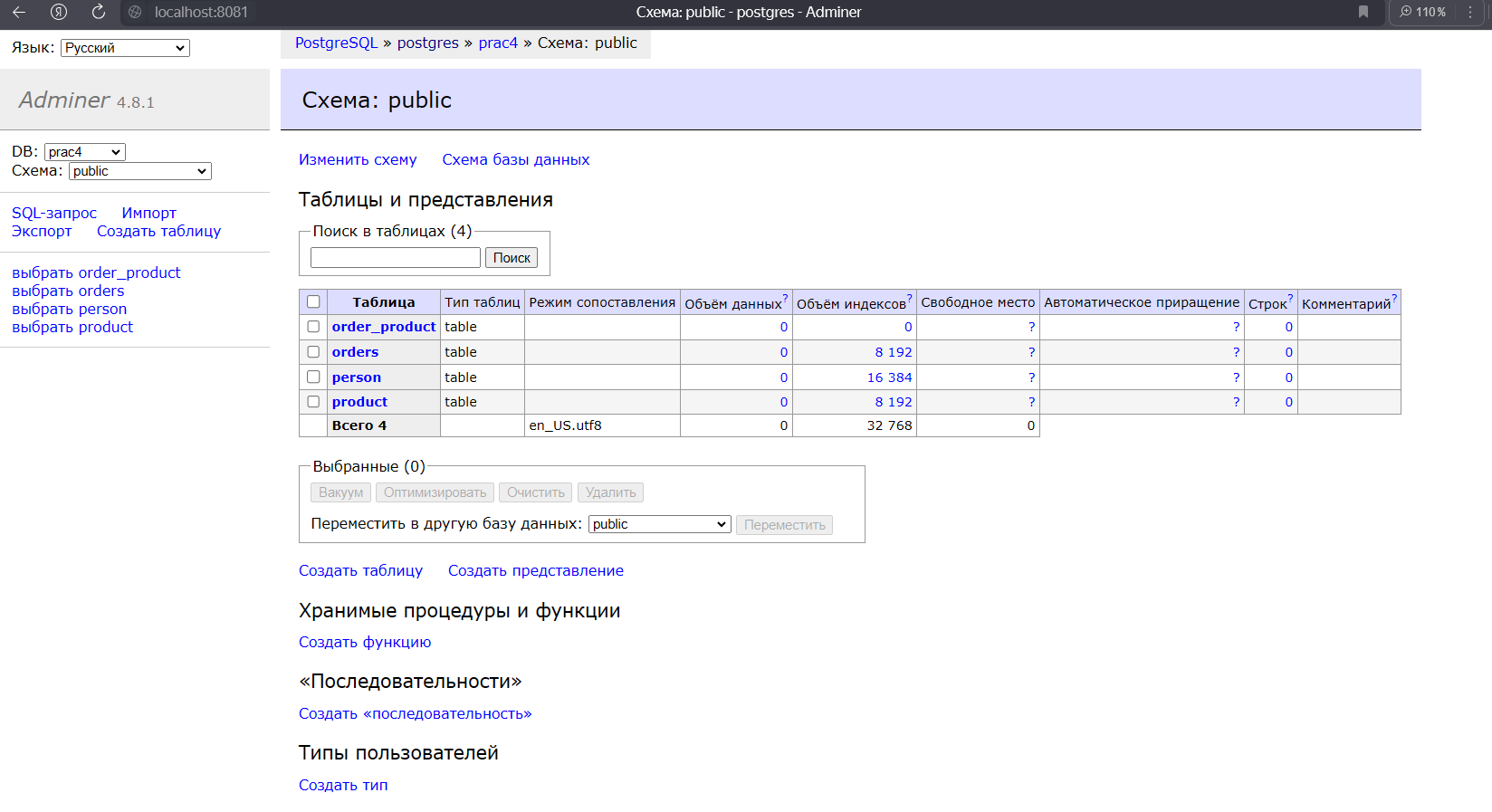
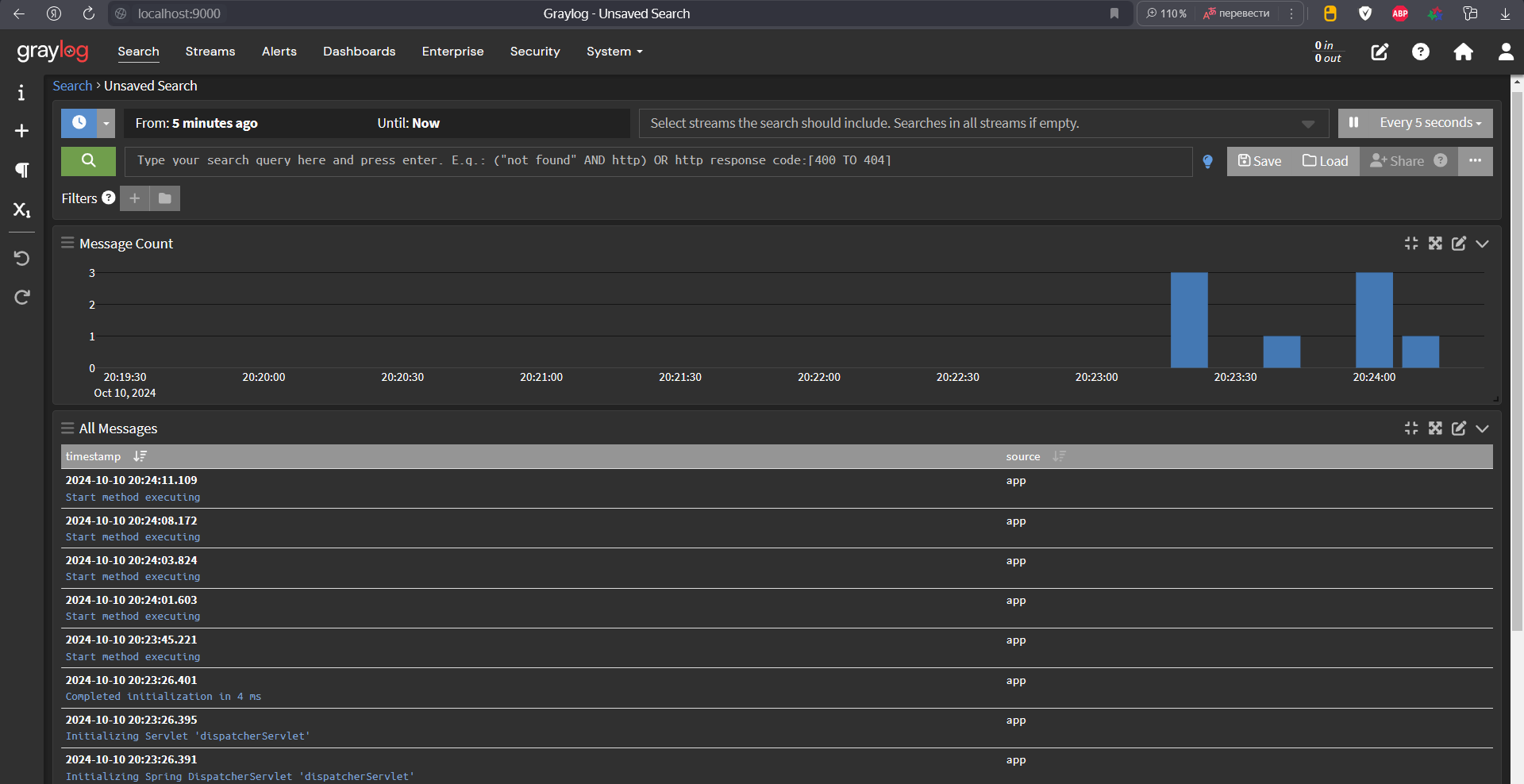


Рисунок 10 – Работа Adminer

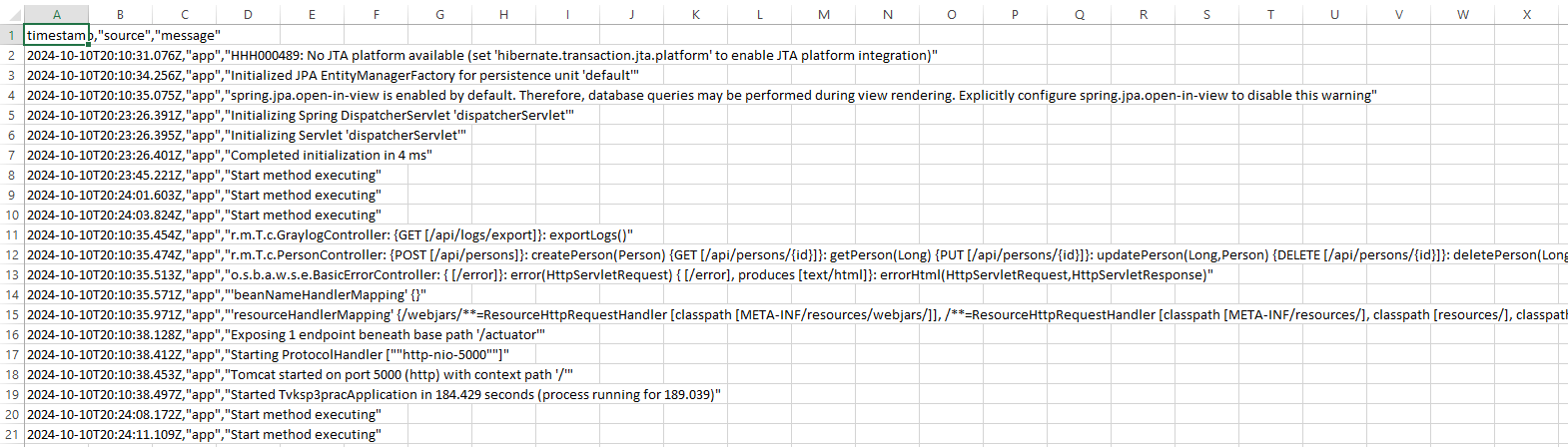


Рисунок 11 – Получение логов в формате CSV

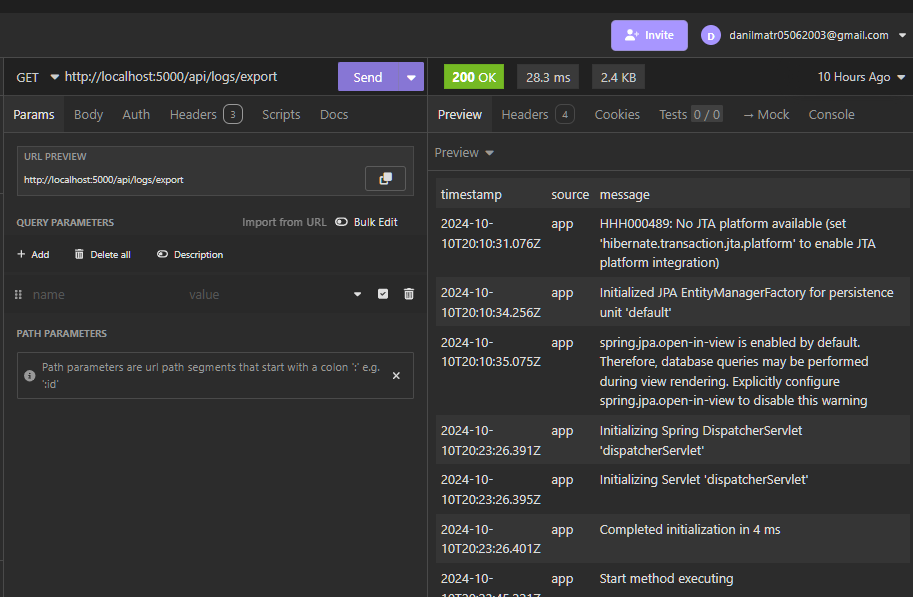


Рисунок 12 – Endpoint для выгрузки логов



Рисунок 13 – Graphana Dashboard

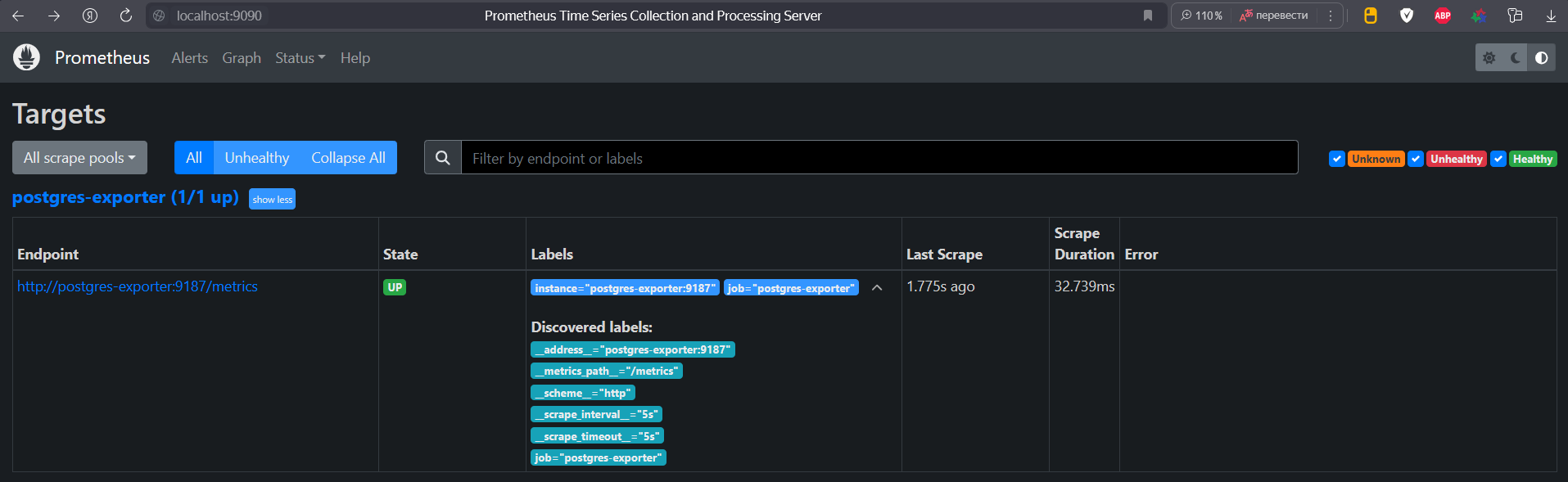


Рисунок 14 – Сбор метрик в Prometheus

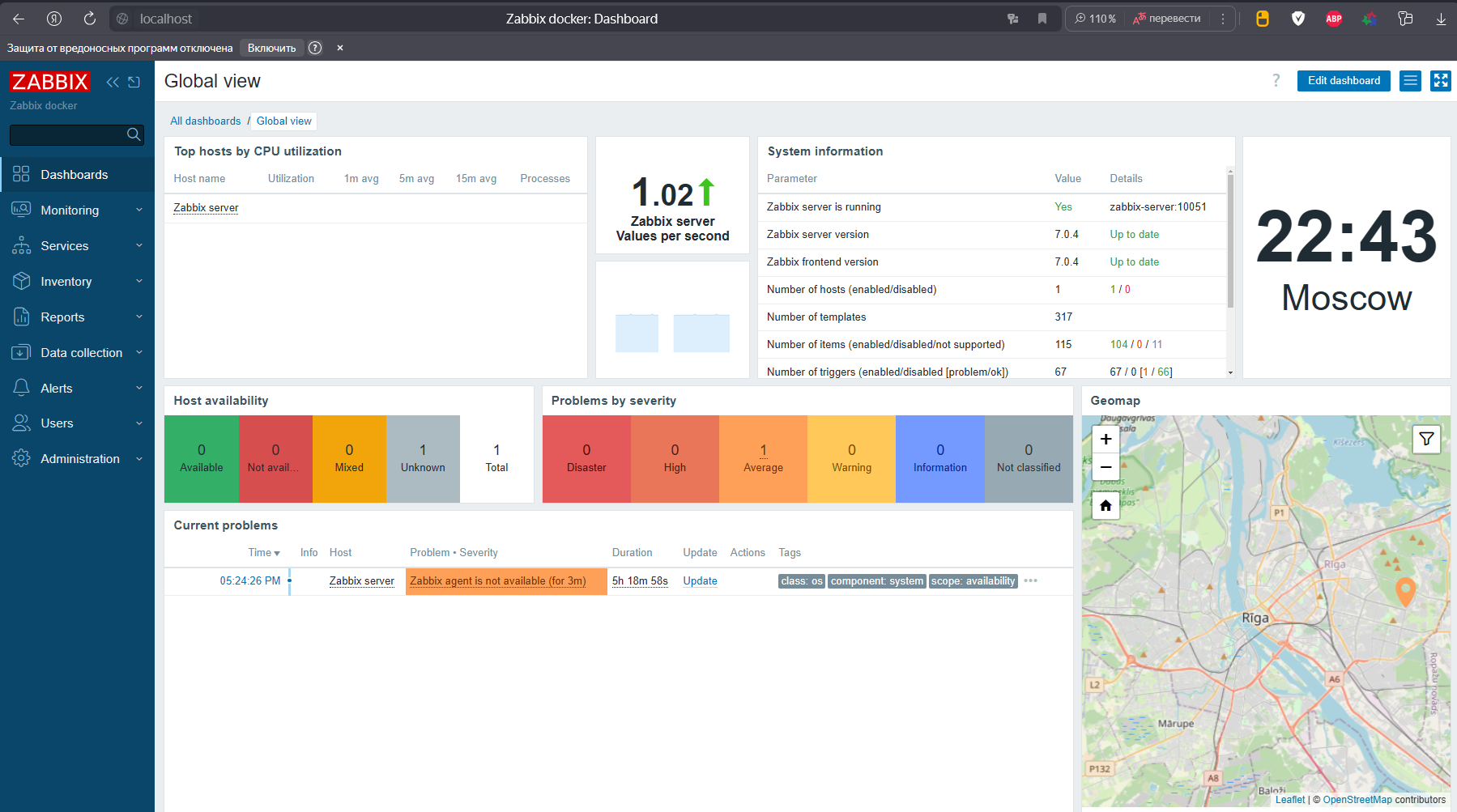


Рисунок 15 – Главная страница Zabbix

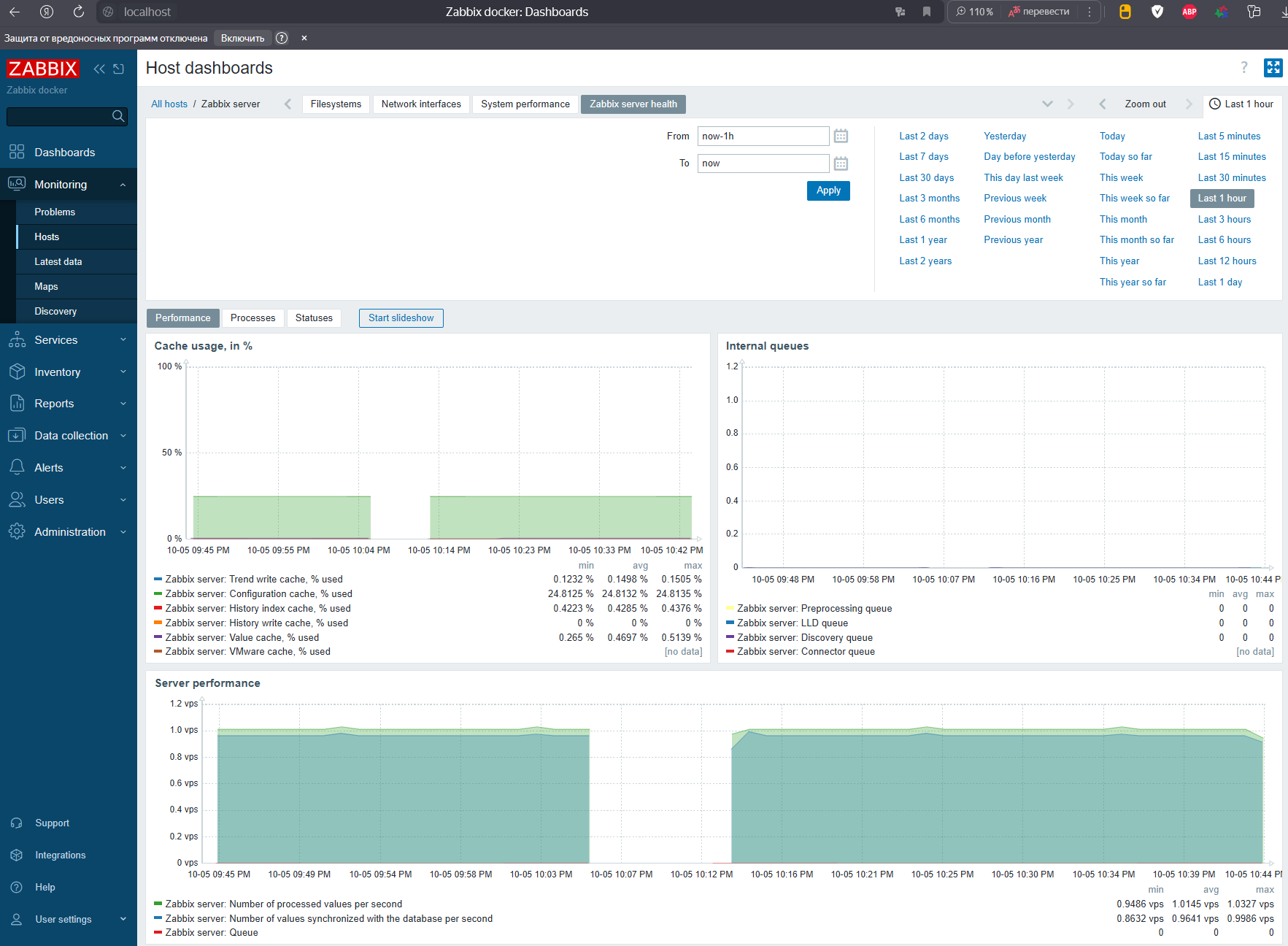


Рисунок 16 – Графики в zabbix

# Выводы

В этой практической работе был разработан Spring Boot сервис, который включает все необходимые компоненты для реализации функциональности CRUD, интеграции с базой данных PostgreSQL, а также для систем мониторинга и логирования.

# Ответы на вопросы к практической работе

1. **Назовите основные различия между Prometheus и Zabbix.**

Метод сбора данных: Prometheus: Имеет активный метод сбора метрик (pull), где он периодически запрашивает статус других систем. Zabbix: Использует как активный (пуллинг), так и пассивный (триггеры) методы сбора данных. Архитектура: Prometheus: Данный инструмент самостоятелен и не требует дополнительных серверов для хранения данных. Zabbix: Требует отдельного сервера Zabbix для хранения и обработки данных. Хранение данных: Prometheus: Использует временные ряды для хранения метрик, что делает его оптимальным для мониторинга. Zabbix: Хранит не только метрики, но и может использовать реляционные базы данных для хранения различных типов информации.

1. **Как можно запустить две базы PostgreSQL в одном docker-compose файле чтобы они работали одновременно и таблицы внутри не пересекались?**

Для запуска двух баз данных PostgreSQL в одном docker-compose.yml файле можно указать разные имена контейнеров и порты

1. **Назовите виды мониторинга систем.**

Мониторинг производительности: Отслеживание использования ресурсов (ЦП, Память, Диск). Сетевой мониторинг: Мониторинг сетевого трафика и доступности сетевых устройств. Мониторинг приложений: Контроль работы приложений и их сервисов. Мониторинг безопасности: Отслеживание событий безопасности и аномалий.

1. **При помощи чего можно передавать конфигурационные переменные в контейнер?**

Конфигурационные переменные можно передавать в контейнер через: Переменные окружения в docker-compose.yml, используя ключ environment. Файлы .env для передачи переменных окружения. Docker Secrets для безопасной конфигурации шифрования данных.

1. **Назовите основные различия Docker Swarm и Docker Compose.**

Цель: Docker Compose: Используется для локальной разработки и управления многоконтейнерными приложениями. Docker Swarm: Предназначен для оркестрации контейнеров в различных узлах кластера. Масштабируемость: Docker Compose: Масштабирует контейнеры на одном хосте. Docker Swarm: Обеспечивает масштабируемость на нескольких хостах и управление распределением нагрузки.

**6. Назовите пример задачи, которую невозможно выполнить используя один лишь docker-compose без Dockerfile**

Невозможно создать образ, содержащий специфические зависимости и настройки приложения, например, установить необходимые пакеты или скопировать код приложения в контейнер, используя только docker-compose. Dockerfile позволяет определить сборку образа от начала до конца, а docker-compose только запускает уже существующие образы.

# Список использованной литературы

1. Virtualization and Containers: An Overview [Электронный ресурс]. – URL: https://www.example.com/virtualization-containers-overview (дата обращения: 04.08.2024)
2. Docker Documentation. [Электронный ресурс]. – URL: https://docs.docker.com/ (дата обращения: 04.08.2024)
3. VMware Official Website. [Электронный ресурс]. – URL: https://www.vmware.com/ (дата обращения: 04.08.2024)
4. "Virtual Machines Versus Containers: What's the Diff?" by John Smith. Virtualization Journal, vol. 25, no. 3, 2019, pp. 45-58.
5. "Understanding the Performance Trade-Offs in Virtualization" by Jane Doe. Proceedings of the International Conference on Cloud Computing, 2022, pp. 112-120.
6. 50 вопросов по Docker, которые задают на собеседованиях, и ответы на них | Хабр. — Текст: электронный [сайт]. — URL: https://habr.com/ru/company/southbridge/blog/528206/
7. Docker Documentation | Docker Documentation — Текст: электронный [сайт]. — URL: https://docs.docker.com/
8. Что такое режим Docker Swarm и когда его использовать? — CloudSavvy ИТ | Cpab. — Текст: электронный [сайт]. — URL: https://cpab.ru/chtotakoe-rezhim-docker-swarm-i-kogda-ego-ispo4lzovat-cloudsavvy-it/
9. Dockerfile reference | Docker Documentation — Текст: электронный [сайт]. — URL: https://docs.docker.com/engine/reference/builder/