|  |  |  |
| --- | --- | --- |
|  | Изображение выглядит как зарисовка, рисунок, символ, корона  Автоматически созданное описание |  |
| 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 №8**

по дисциплине «Технологии виртуализации клиент-серверных приложений»

**Тема практической работы: «Взаимодействие с minikube»**

**Студент группы** ИКБО-20-21 Сидоров С.Д.

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

(подпись студента)

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

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

(подпись руководителя)

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

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

Москва 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ РАБОТЫ 3](#_Toc181980139)

[ВОПРОСЫ 16](#_Toc181980140)

[ВЫВОД 18](#_Toc181980141)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc181980142)

ЦЕЛЬ РАБОТЫ

Развернуть в кластере Kubernetes с помощью minikube 3 сервиса. Один из которых взаимодействует с redis, два оставшихся должны взаимодействовать друг с другом через kafka. Для логирования применить graylog, для мониторинга использовать jaeger, grafana, prometheus. В качестве базы данных должна использоваться PostgreSQL, развернутая отдельно для каждого сервиса.

ХОД РАБОТЫ

Основные составляющие сервиса представлены на рисунках



Рисунок 1 – Конфигурация подключения Redis Для AuthService



Рисунок 2 – Конфигурация работы с OpenTracing

**ВЫВОД**

В рамках данной практической работы были получены навыки установки лимитов по RAM и CPU для подов, а также был получен опыт по обнаружению ошибок, связанных с превышением установленных ограничений.

вопросы

1. 3 QoS-класса (Quality of Service) в Kubernetes:

Guaranteed: Все запросы и лимиты ресурсов указаны, контейнеры гарантированно получают ресурсы.

Burstable: Заданы запросы ресурсов (requests), но не все лимиты или запросы меньше лимитов.

BestEffort: нет указанных запросов или лимитов, используется оставшаяся производительность узла.

1. Основные ресурсы системы и их единицы измерения в Kubernetes:

**CPU:** измеряется в миллиядрах (mCPU) или ядрах (1 CPU = 1000mCPU).

**Memory**: измеряется в байтах, но обычно используют Mi или Gi (гибибайты).

Storage: измеряется в Gb.

1. Для чего нужен HPA (Horizontal Pod Autoscaler):

Автоматически масштабирует количество подов (горизонтальное масштабирование) в зависимости от нагрузки, основываясь на метриках, таких как использование CPU, памяти или пользовательские метрики.

1. Для чего необходимо устанавливать ограничения в Kubernetes:

Предотвращает перерасход ресурсов контейнерами.

Обеспечивает стабильность системы, избегая ситуаций, когда один контейнер потребляет все ресурсы узла.

Позволяет эффективно распределять ресурсы между контейнерами и узлами.

1. Что будет с узлом при превышении ограничений:

**Превышение лимитов контейнера:**

CPU: Контейнер ограничивается и не может использовать больше лимита.

Memory: Контейнер будет завершён (OOMKilled) при превышении лимита памяти.

**Превышение ресурсов узла:**

Kubernetes начнёт эвакуировать поды с наименьшим QoS-классом (сначала BestEffort, затем Burstable).

Узел может выйти из строя, если ресурсы полностью исчерпаны.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Docker Documentation | Docker Documentation – Текст: электронный [сайт]. – URL: <https://docs.docker.com/>

2. Kubernetes – Текст: электронный [сайт]. – URL: <https://kubernetes.io/>

3. Docker Compose Documentation | Docker Documentation – Текст: электронный [сайт]. – URL: <https://docs.docker.com/compose/>

4. 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/>