|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №6**

по дисциплине «Технологии виртуализации клиент-серверных приложений»

**Студент группы** ИКБО-20-19 Московка Артём Александрович

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

(подпись студента)

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

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

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

**Задание**

Создать приложения для записи журнала в Redis используя Kubernetes, Ingress.

**Ход работы**

Dockerfile (Рисунок 1).

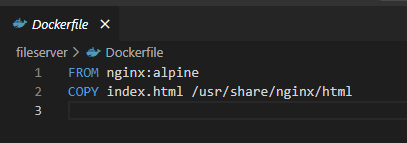
****

Рисунок 1 - dockerfile

fileserver-service.yaml (Рисунок 2).

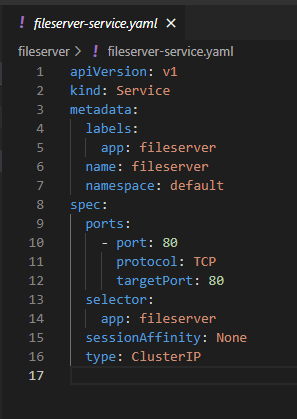


Рисунок 2 - fileserver-service.yaml

fileserver-deployment.yaml (Рисунок 3).

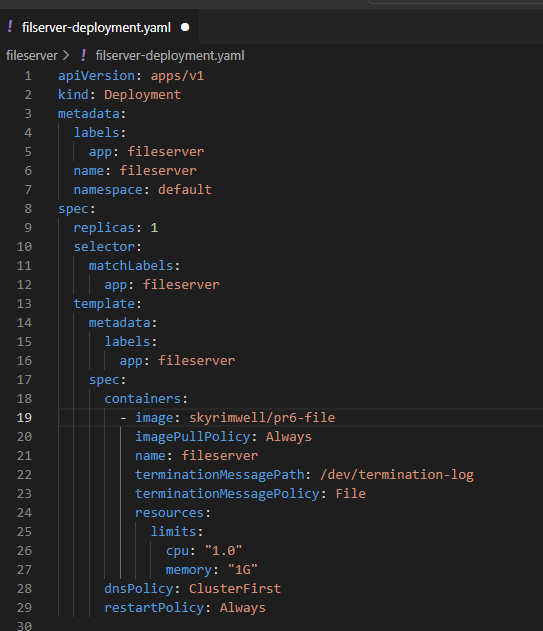
****

Рисунок 3 - fileserver-deployment.yaml

frontend-deployment.yaml (Рисунок 4).

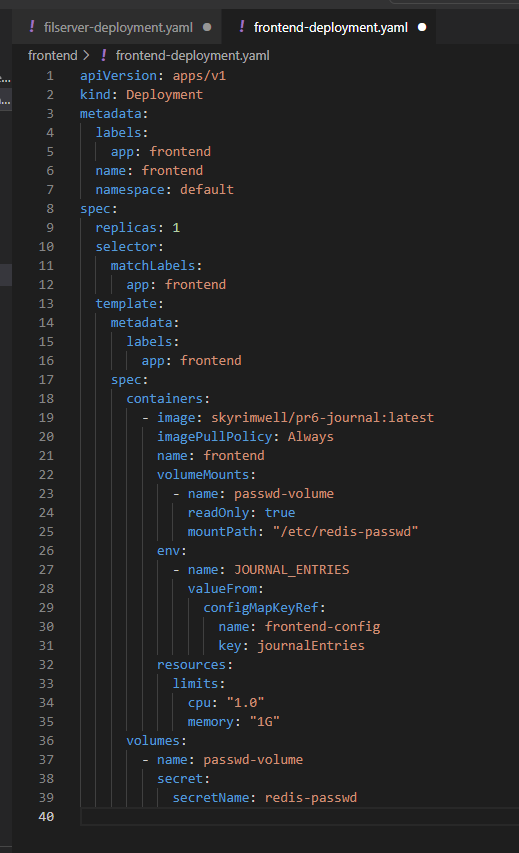


Рисунок 4 - frontend-deployment.yaml

fileserver-service.yaml(Рисунок 5)

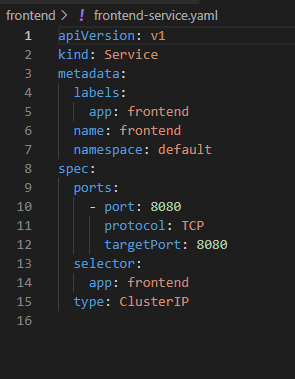


Рисунок 5 - fileserver-service.yaml

ingress.yaml (Рисунок 6)

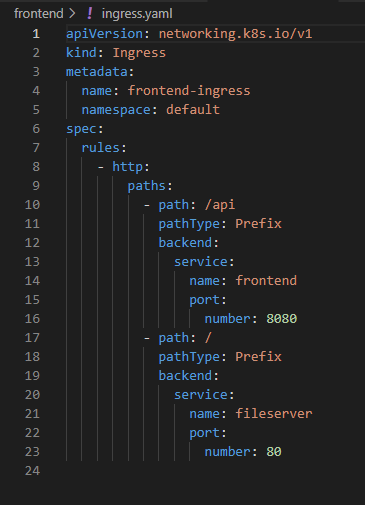
****

Рисунок - ingress.yaml

redis-read.yaml (Рисунок 7).

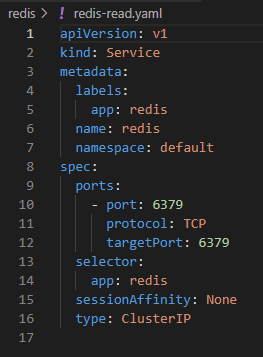
****

Рисунок 7 - redis-read.yaml

redis-write.yaml (Рисунок 8).

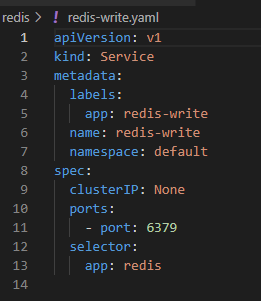


Рисунок 8 - redis-write.yaml

stateful-set.yaml (Рисунок 9).

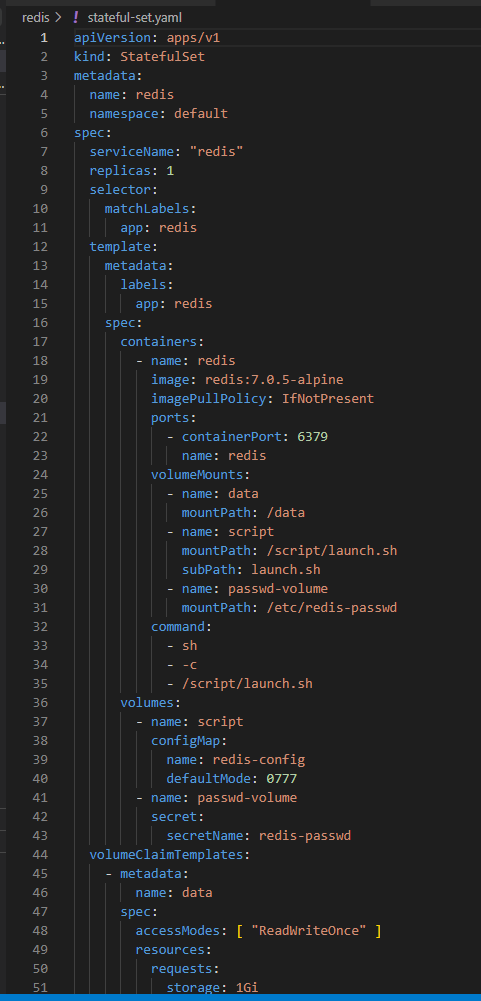


Рисунок 9 - stateful-set.yaml

launch.sh (Рисунок 10).

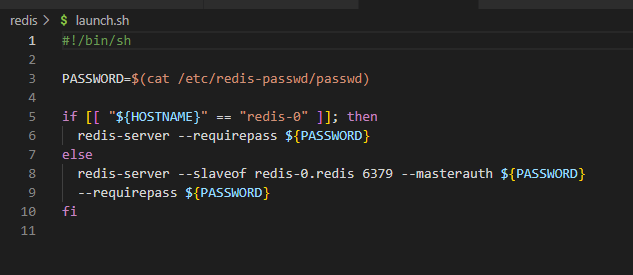


Рисунок 10 - launch.sh

Запустим minikube (Рисунок 11).

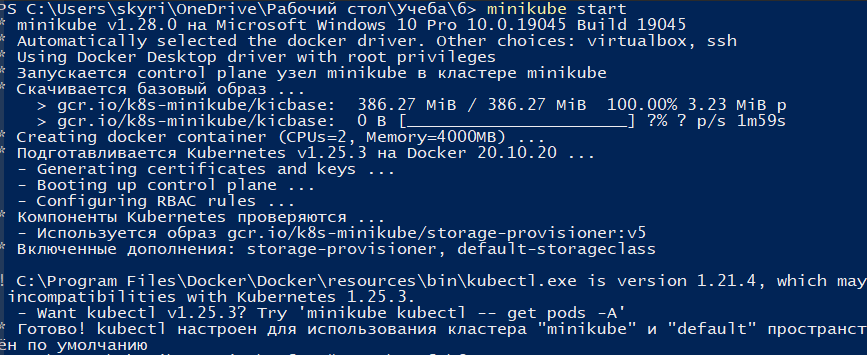
****

Рисунок – Демонстрация запуска minikube

Создадим configmap’ы (Рисунок 12).

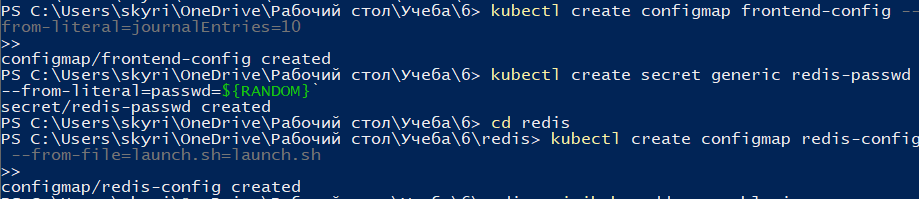
****

Рисунок – Демонстрация создания configmap

Включим аддон Ingress (Рисунок 13).

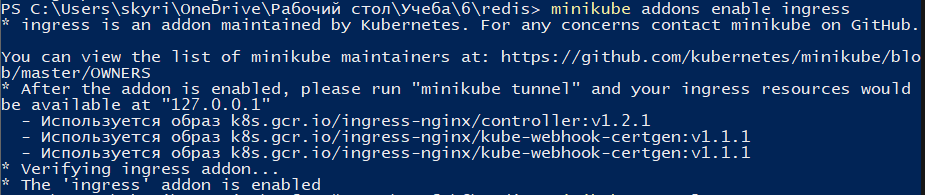
****

Рисунок – Демонстрация включения аддона

Включим туннелирование трафика (Рисунок 14).

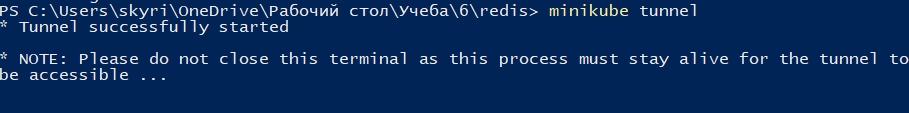
****

Рисунок – Демонстрация включения туннелирования

Применим настройки (Рисунок 15).

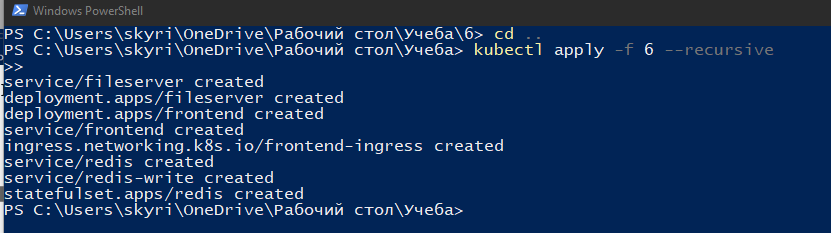
****

Рисунок – Демонстрация применения настроек

Проверим хостинг (Рисунок 16).

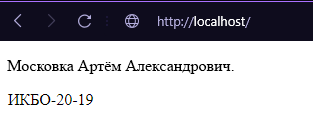


Рисунок – Демонстрация результата проверки хостинга

Проверим что находится сейчас в памяти Redis (Рисунок 17).

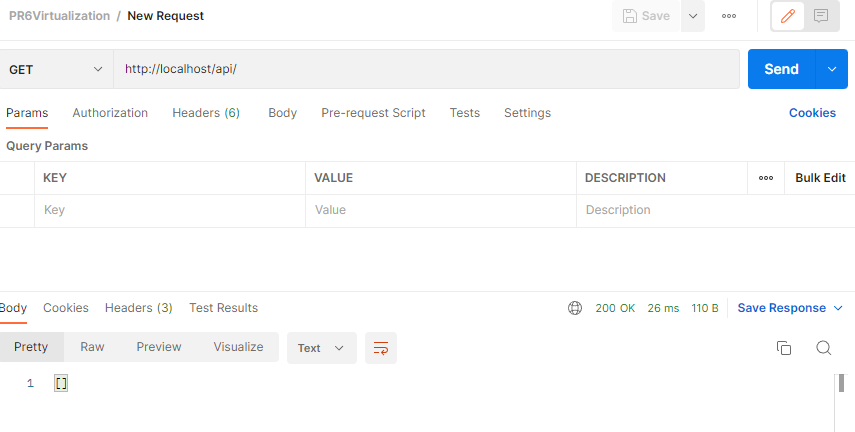
****

Рисунок – Демонстрация результата проверки памяти

Запишем в память первую запись (Рисунок 18).

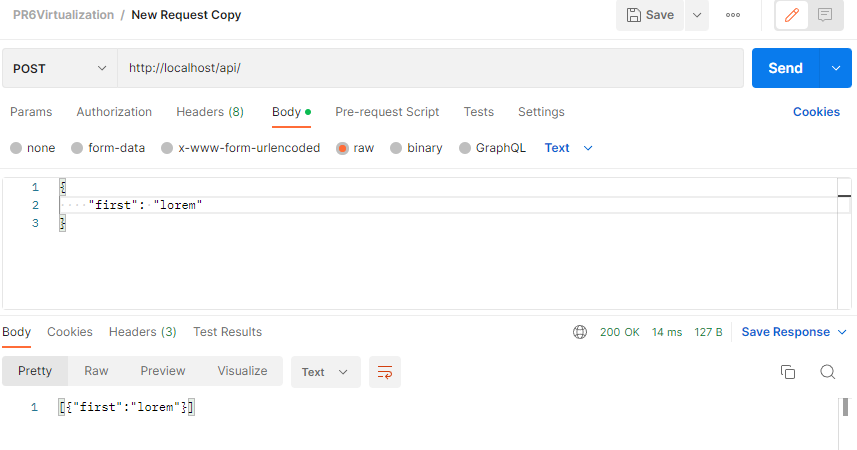
****

Рисунок – Демонстрация результата создания записи

Проверим память (Рисунок 19).

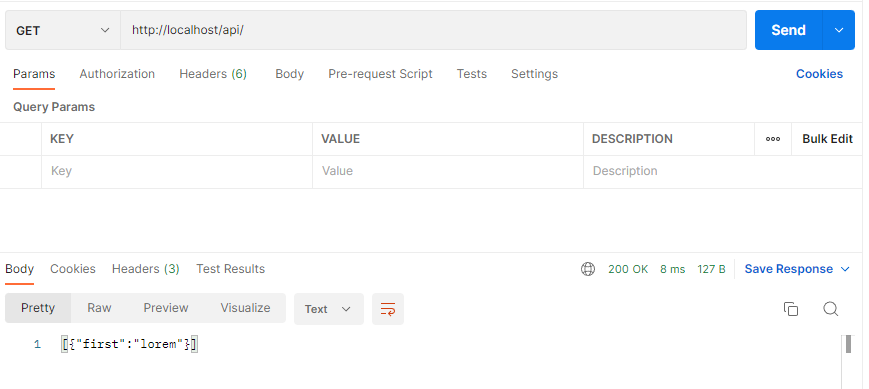
****

Рисунок – Демонстрация результата проверки памяти

Создадим вторую запись (Рисунок 20).

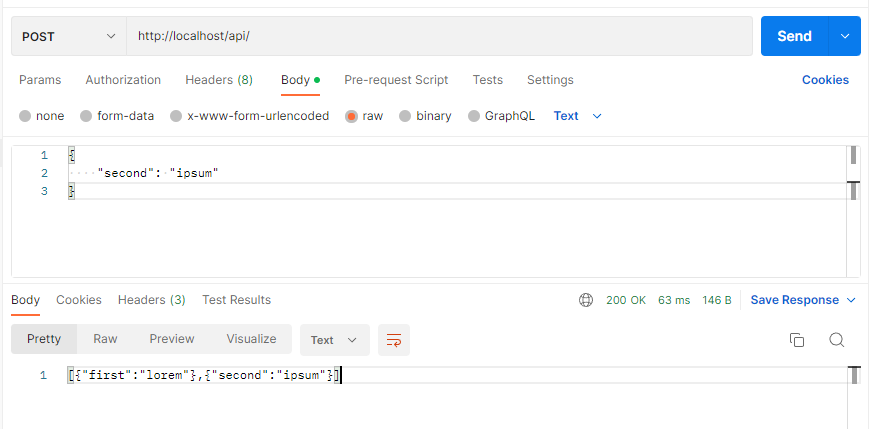
****

Рисунок – Демонстрация результата создания записи

Проверим память (Рисунок 21).

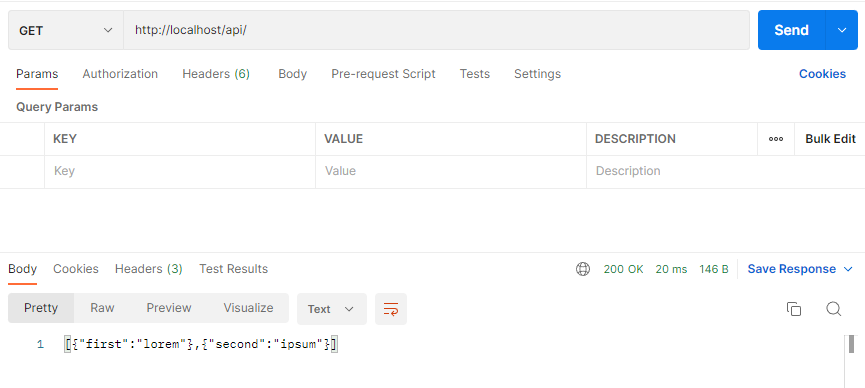


Рисунок – Демонстрация результата проверки памяти

**Вывод**

В результате выполнения данной практической работы были получены навыки работы с Kubernetes, minikube, Ingress.

**Ответы на контрольные вопросы**

1. Для чего нужен ресурс Deployment?

Развёртывание (Deployment) — это абстракция Kubernetes, которая позволяют нам управлять тем, что всегда присутствует в жизненном цикле приложения. Речь идёт об управлении изменениями приложений. Приложения, которые не изменяются, это, так сказать, «мёртвые» приложения. Если же приложение «живёт», то можно столкнуться с тем, что периодически изменяются требования к нему, расширяется его код, этот код упаковывается и разворачивается. При этом на каждом шаге данного процесса могут совершаться ошибки.

Ресурс вида Deployment позволяет автоматизировать процесс перехода от одной версии приложения к другой. Это делается без прерывания работы системы, а если в ходе этого процесса произойдёт ошибка, у нас будет возможность быстро вернуться к предыдущей, рабочей версии приложения.

1. Почему не стоит хранить пароли в ConfigMap?

ConfigMaps хранятся ненадежно, и их значения не зашифрованы. Они не должны содержать конфиденциальные данные, которые могут представлять угрозу безопасности или конфиденциальности в случае утечки.

Не помещайте пароли, ключи API или ключи шифрования в ConfigMap – вместо этого используйте Kubernetes Secret, поскольку они работают так же, как ConfigMaps, но с дополнительной защитой. Системы, которым требуется соединение с базой данных, должны поместить имя хоста в ConfigMap, а учетные данные – в отдельный секрет.

3. Что необходимо для настройки внешнего доступа для HTTP-трафика?

Назовите шаги.

Открыть порты для доступа в поде, туннелировать их с помощью Ingress.

4. Чем отличается развертывание stateful от развертывания клиентского

приложения?

Сильных различий нет, например для redis можно пользоваться удаленными постоянными томами.

5. Где хранится том с секретными данными?

Отдельно от пода, данные загружаются лишь при необходимости, а затем сразу же удаляются

6. Как работает связка PersistentVolume и PersistentVolumeClaim?

Условно Persistent Volumes можно считать аналогом нод в самом кластере Kubernetes. Допустим, у вас есть несколько разных хранилищ. К примеру, одно быстрое на SSD, а другое медленное на HDD. Вы можете создать 2 Persistent Volumes в соответствии с этим, а затем выделять подам место в этих томах. Кубернетис поддерживает огромное количество подключаемых томов с помощью плагинов.

PersistentVolumeClaim (PVC) есть не что иное как запрос к Persistent Volumes на хранение от пользователя. Это аналог создания Pod на ноде. Поды могут запрашивать определенные ресурсы ноды, то же самое делает и PVC.

**Список использованных источников**

1. Kubernetes: лучшие практики. — СПб.: Питер, 2021. — 288 с.: ил. — (Серия «Для профессионалов»).
2. K8S для начинающих. Первая часть — Текст: электронный [сайт]. — URL: https://habr.com/ru/post/589415/
3. Kubernetes или с чего начать, чтобы понять что это и зачем он нужен — Текст: электронный [сайт]. — URL: https://habr.com/ru/company/otus/blog/537162/
4. Основы Kubernetes — Текст: электронный [сайт]. — URL: https://habr.com/ru/post/258443/