|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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-20-21 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 3](#_Toc145588020)

[Ход работы 6](#_Toc145588021)

[Выводы 10](#_Toc145588022)

[Ответы на вопросы к практической работе 11](#_Toc145588023)

[Список использованной литературы 13](#_Toc145588024)

# Цель работы

Цель данной практической работы заключается в освоении процесса развертывания приложения в среде контейнеризации с использованием Kubernetes (k8s). Данная задача предполагает изучение основных принципов работы с Kubernetes, включая подготовку кластера для развертывания, конфигурацию манифестов, развертывание и масштабирование приложения, а также обеспечение его работоспособности и отслеживание его состояния с помощью инструментов Kubernetes.

# Ход работы



Рисунок 1 – Конфигурационный файл deployment

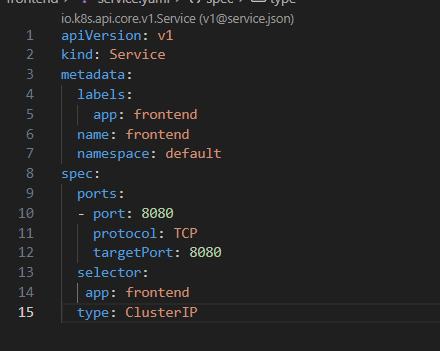


Рисунок 2 – Конфигурационный файл сервиса

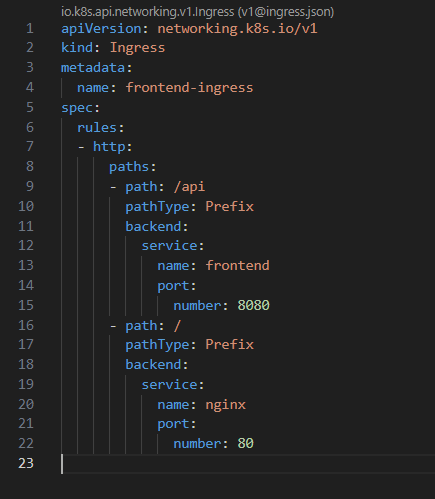


Рисунок 3 – Конфигурационный файл ingress

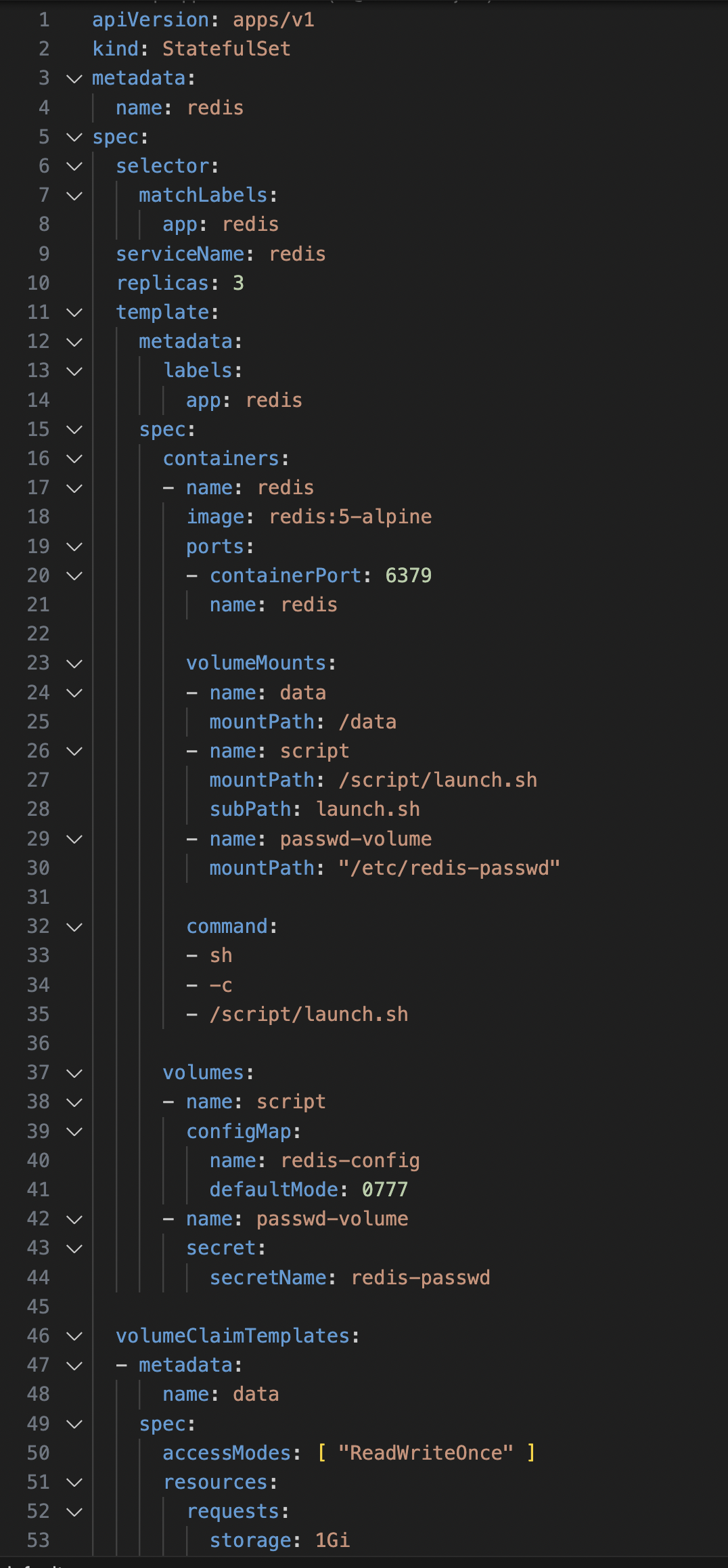


Рисунок 4 – Конфигурационный файл statefulset redis

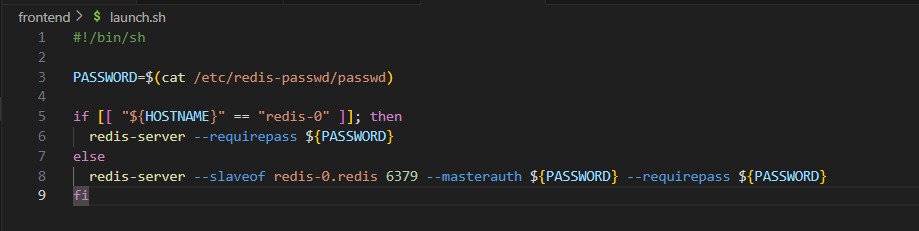


Рисунок 5 – Скрипт запуска redis по принципу master-slave

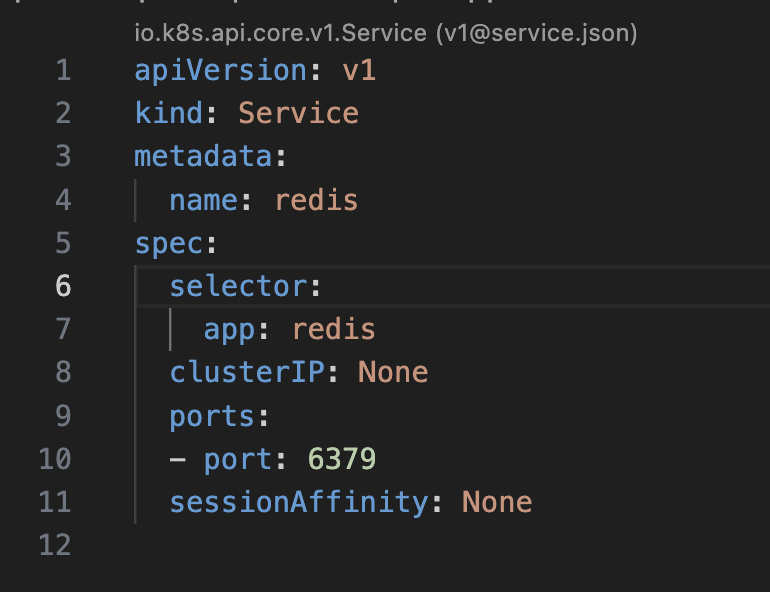


Рисунок 6 – Конфигурационный файл headless сервиса для redis

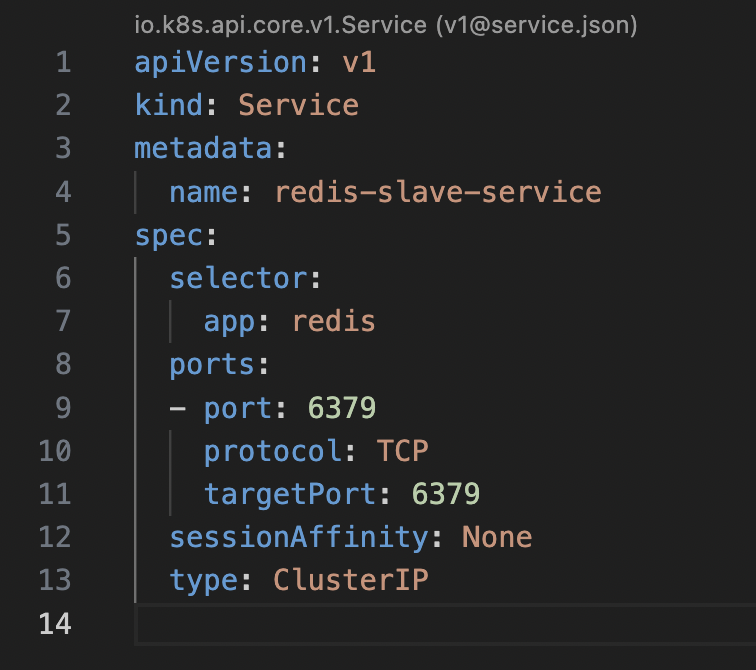


Рисунок 7 – Конфигурационный файл сервиса для redis

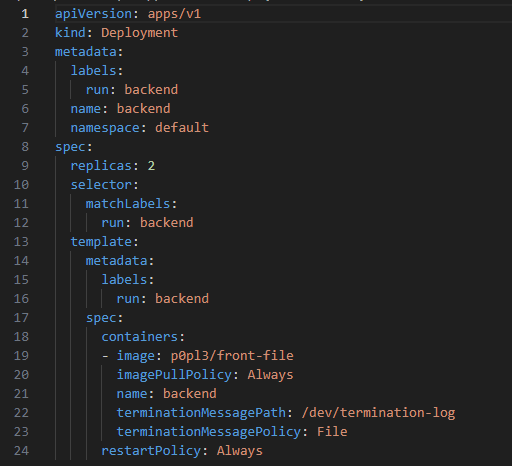


Рисунок 8 – Конфигурационный файл deployment для серверной части

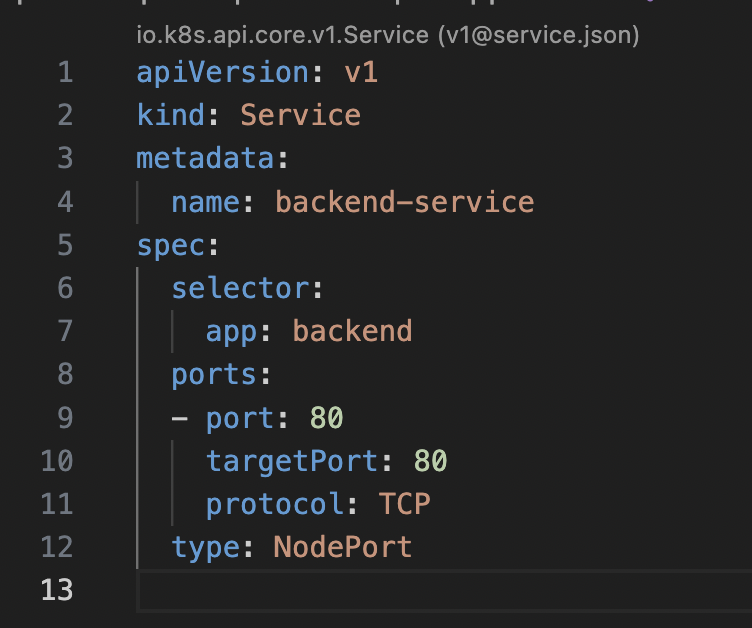


Рисунок 9 – Конфигурационный файл сервиса для серверной части

Запустим приложения в minikube и проверим их работоспособность. Это показано на рисуках 10, 11.



Рисунок 10 – Cодержимое кластера

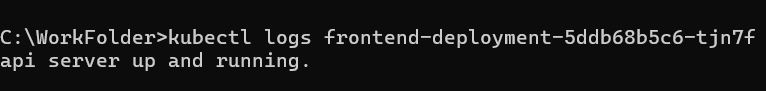


Рисунок 11 – Проверка работоспособности приложения

# Выводы

В ходе выполнения практической работы были освоены ключевые навыки работы с системой Kubernetes. Это включает в себя развертывание типового приложения в кластере Kubernetes, а также изучение основных понятий, таких как сервисы, деплойменты, масштабирование. Было получено глубокое понимание принципов работы Kubernetes и его преимуществ в контексте развертывания и управления масштабируемыми приложениями.

# Ответы на вопросы к практической работе

1. **Для чего нужен ресурс Deployment?**

Ресурс Deployment в Kubernetes используется для управления развертыванием приложений. Он обеспечивает декларативное обновление приложений на основе их контейнеров и поддерживает желаемое состояние, гарантируя масштабирование, обновление и откат приложений.

1. **Почему не стоит хранить пароли в ConfigMap?**

Хранение паролей в ConfigMap не рекомендуется из-за их небезопасности. ConfigMap предназначен для хранения конфигурационных данных, а не для чувствительной информации, такой как пароли. Для хранения секретной информации следует использовать Kubernetes Secrets, которые обеспечивают шифрование и более безопасное хранение информации.

1. **Что необходимо для настройки внешнего доступа для HTTP-трафика? Назовите шаги.**

Для настройки внешнего доступа для HTTP-трафика необходимо: Создать службу типа LoadBalancer или NodePort, которая будет проксировать трафик на поды, на которых развернуто приложение. Настроить правила файрвола для разрешения трафика на порты, используемые службой.

1. **Чем отличается развертывание stateful от развертывания клиентского приложения?**

Развертывание stateful-приложений отличается от развертывания клиентского приложения в том, что stateful-приложения обычно имеют состояние, которое должно быть сохранено, например, базы данных. Для таких случаев необходимо использовать StatefulSet в Kubernetes, чтобы обеспечить уникальность имени, стабильное хранилище и уникальные сетевые идентификаторы.

1. **Где хранится том с секретными данными?**

Том с секретными данными хранится в объекте Kubernetes под названием "Secret". Этот ресурс используется для хранения конфиденциальной информации, такой как пароли, ключи API и другие секреты.

1. **Как работает связка PersistentVolume и PersistentVolumeClaim?**

PersistentVolume (PV) и PersistentVolumeClaim (PVC) в Kubernetes работают вместе для обеспечения постоянного хранилища для приложений. PV представляет собой ресурс, который предоставляет постоянное хранилище, в то время как PVC запрашивает хранилище и "претендует" на PV определенного типа и размера. PV и PVC создают абстракцию между хранилищем и потребителями данных, позволяя эффективно управлять хранилищем в кластере Kubernetes.

# 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/