|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 Анваржонов Ж. Т

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

(подпись студента)

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

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

(подпись руководителя)

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

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

Москва 2022

**Постановка задачи**

В результате выполнения работы вам необходимо отобразить в отчете все этапы конфигурации программной системы и показать работоспособность

системы, доступ к которой настроен с помощью Ingress, с помощью браузера

**Выполнение работы**

Cоздадим конфигурацию в объекте ConfigMap



Рисунок 1 – конфигурация configMap

Предоставить конфигурационную информацию в виде переменной среды в самом приложении



Рисунок 2 – правки в deployment.yml

Для аутентификации в сервере Redis используем обычный пароль, который будет храниться в объекте Secret:

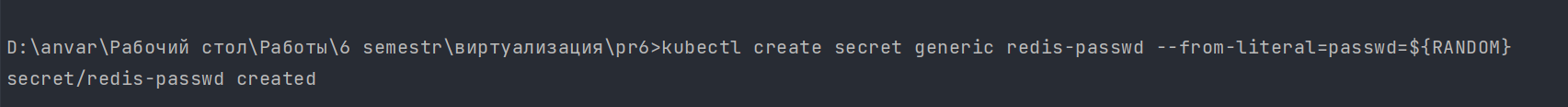


Рисунок 3 – создание объекта Secret

Привяжем наш секрет в наше приложение с помощью volume:

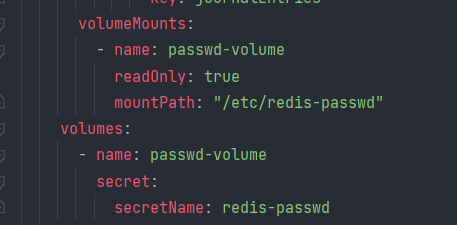


Рисунок 4 – привязывание секрета с помощью томов

Cоздадим дополнение к ReplicaSet – StatefulSet для Redis с постоянными томами:

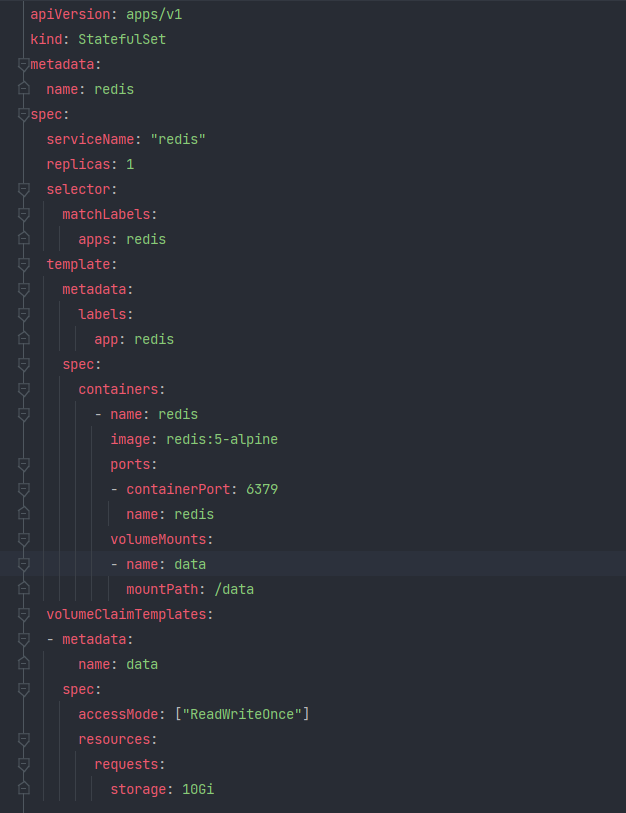
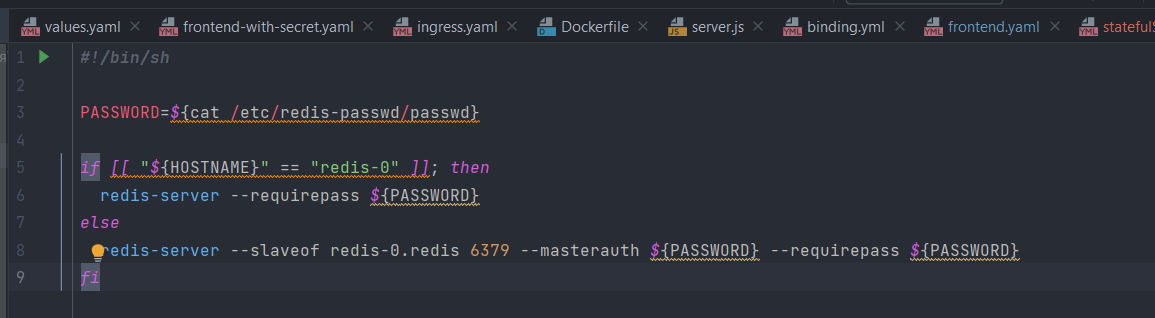


Рисунок 5 – ресурс StatefulSet

Напишем сценарий, пригодного для запуска во всех контейнерах StatefulSet’a и оформим его в виде ConfigMap:



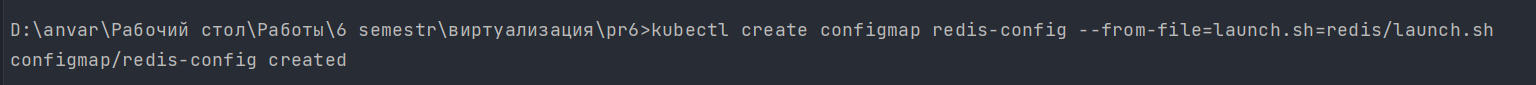


Рисунок 6-7 – добавление скрипта в виде ConfigMap

Добавим объект СonfigMap в StatefulSet и используем его как команду для управления контейнером:

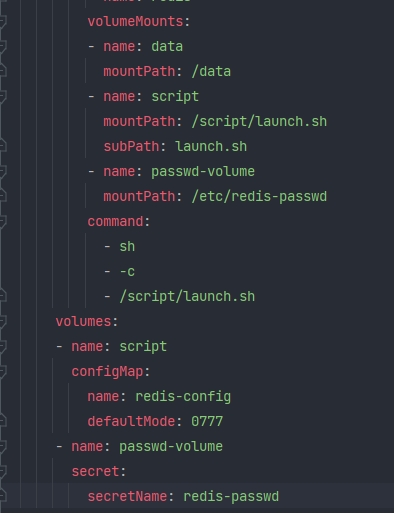


Рисунок 8 - правки в statefulSet.yml

Сделаем наш stateful-сервис Redis доступным для клиентского приложения. Для этого создадим два разных Service.

Первый – будет читать данные из Redis:

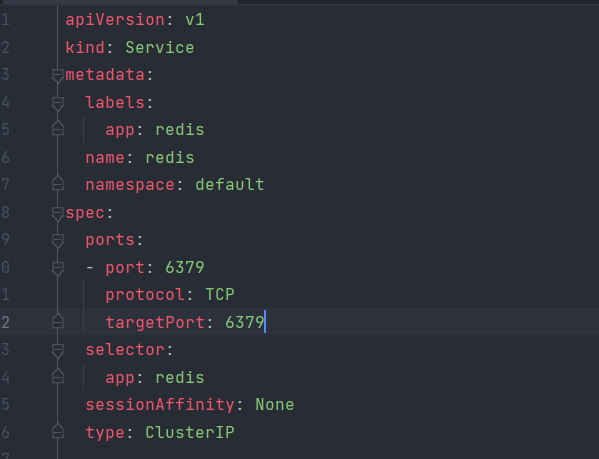


Рисунок 9 – создание сервиса для чтение из Redis

Так как выполнение записи потребует обращения к ведущей реплике Redis (под

номером 0), создадим для этого неуправляемый (headless) Service. У него нет

IP-адреса внутри кластера; вместо этого он задает отдельную DNS-запись для

каждого pod в StatefulSet. То есть мы можем обратиться к нашей ведущей

реплике по доменному имени redis-0.redis:

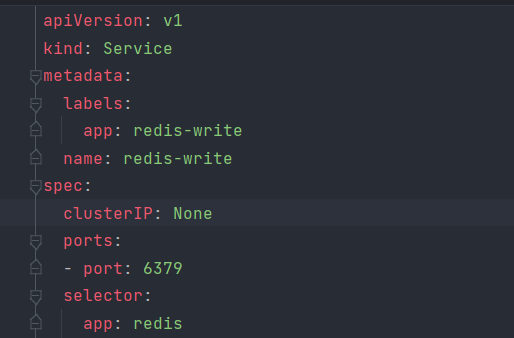


Рисунок 10 – создание сервиса для записи в Redis

Создадим сервер статических файлов:

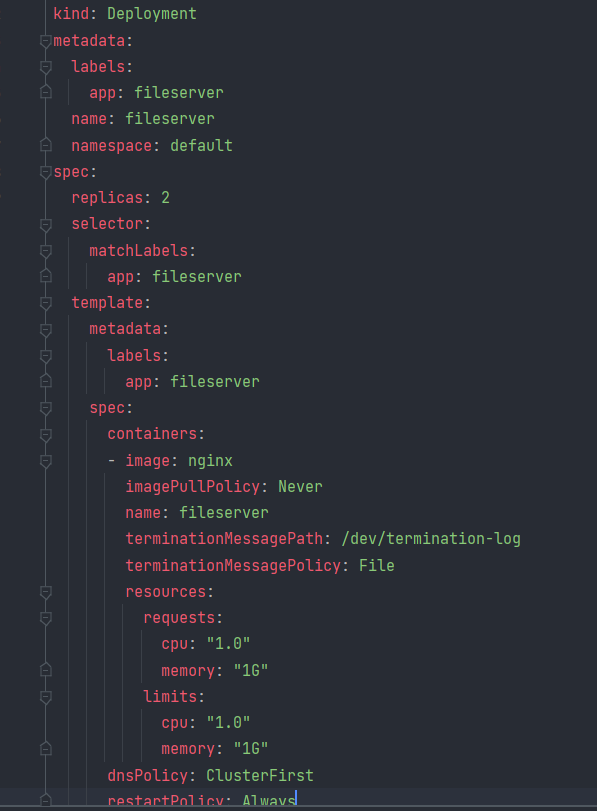


Рисунок 11 – файл deployment.yml для nginx

Добавим в ресурс Ingress путь к fileserver:

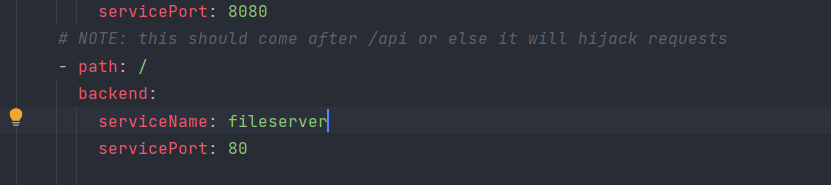


Рисунок 12 – правки в ingress

Поправим DockerFile для успешного запуска нашего frontend сервера:

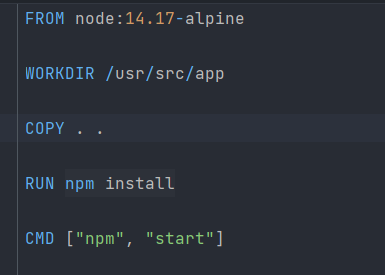
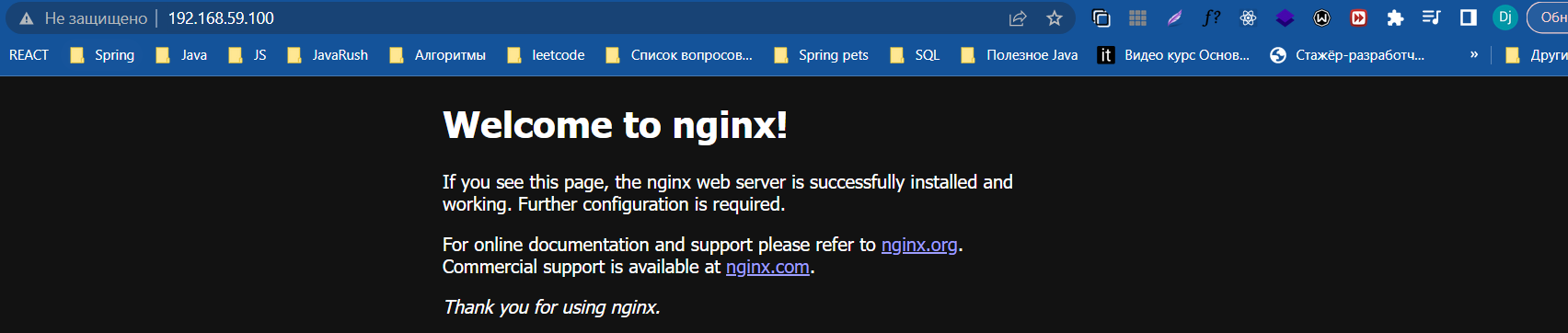


Рисунок 13 – правки в dockerfile

Покажем работоспособность:



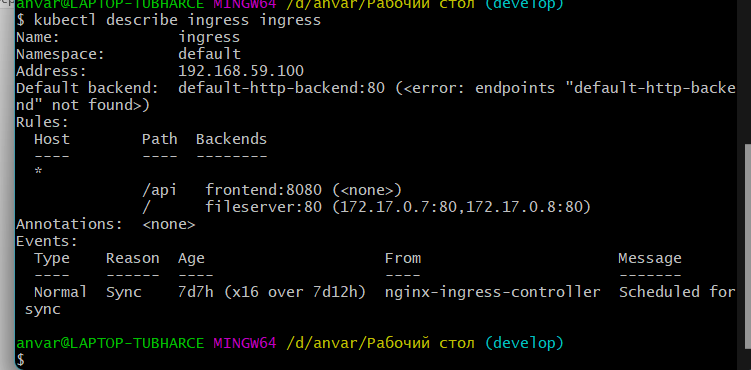
****

Рисунок 14-15 - маршрутизация внутри кластера, главная страница nginx

**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?

Пароли в ConfigMap не рекомендуются хранить, потому что там они хранятся не в зашифрованном виде.

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

Cуществует как минимум три способа предоставления сервиса для внешних сетевых запросов. Они позволяют отправить запрос из-за пределов кластера Kubernetes службе внутри кластера. С помощью Kubernetes NodePort, Ingress и LoadBalancer

NodePort — это параметр конфигурации, который надо объявлять в YAML сервиса. Установите тип спецификации сервиса на NodePort. Затем Kubernetes выделит определенный порт на каждом узле для этой службы, и любой запрос к вашему кластеру на этом порту будет перенаправлен службе.

Ingress в отличие NodePort и LoadBalancer является независимым ресурсом для сервисов. Его можно определять, cоздавать и удалять независимо от сервиса. В ingress для возможности внешнего доступа мы настраиваем правила маршрутизации внутри кластера

4. Чем отличается развертывание stateful от развертывания клиентского

приложения?

Развертывание stateful принципиально не отличается от развертывания клиентского приложения, однако наличие состояния вносит дополнительные сложности. Прежде всего, планирование функционирования pod в Kubernetes зависит от ряда факторов, таких как работоспособность узла, обновление или перебалансировка. Если данные экземпляра Redis хранятся на каком то конкретном сервере или в самом контейнере, то будут потеряны при миграции или перезапуске данного контейнера. Чтобы этого избежать, при выполнении в Kubernetes stateful  приложений нужно обязательно использовать удаленные постоянные тома (PersistentVolumes)

1. Где хранится том с секретными данными?

Уровень управления Kubernetes хранит секретные значения в своем экземпляре etcd. Том с секретными данными монтируется в контейнере по пути, указанному в volumeMounts deployment сервиса

1. Как работает связка PersistentVolume и PersistentVolumeClaim?

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.

**Вывод**

В данной практической работе научились развертывать систему с stateless клиентским приложением, делегируя данную функцию серверу Redis