

**Национальный исследовательский Университет
ИТМО**

Факультет инфокоммуникационных технологий

Дисциплина: «Введение в распределенные технологии»

Лабораторная работа №2

"Развертывание веб сервиса в Minikube, доступ к веб интерфейсу
сервиса. Мониторинг сервиса."

Выполнил студент:

*Сагайдак Алина Алексеевна,
K4111c*

Проверил преподаватель:

Филянин Иван Викторович

Санкт-Петербург, 2024 год

Описание

В данной лабораторной работе вы познакомитесь с развертыванием полноценного веб сервиса с несколькими репликами.

Цель работы

Ознакомиться с типами "контроллеров" развертывания контейнеров, ознакомится с сетевыми сервисами и развернуть свое веб приложение.

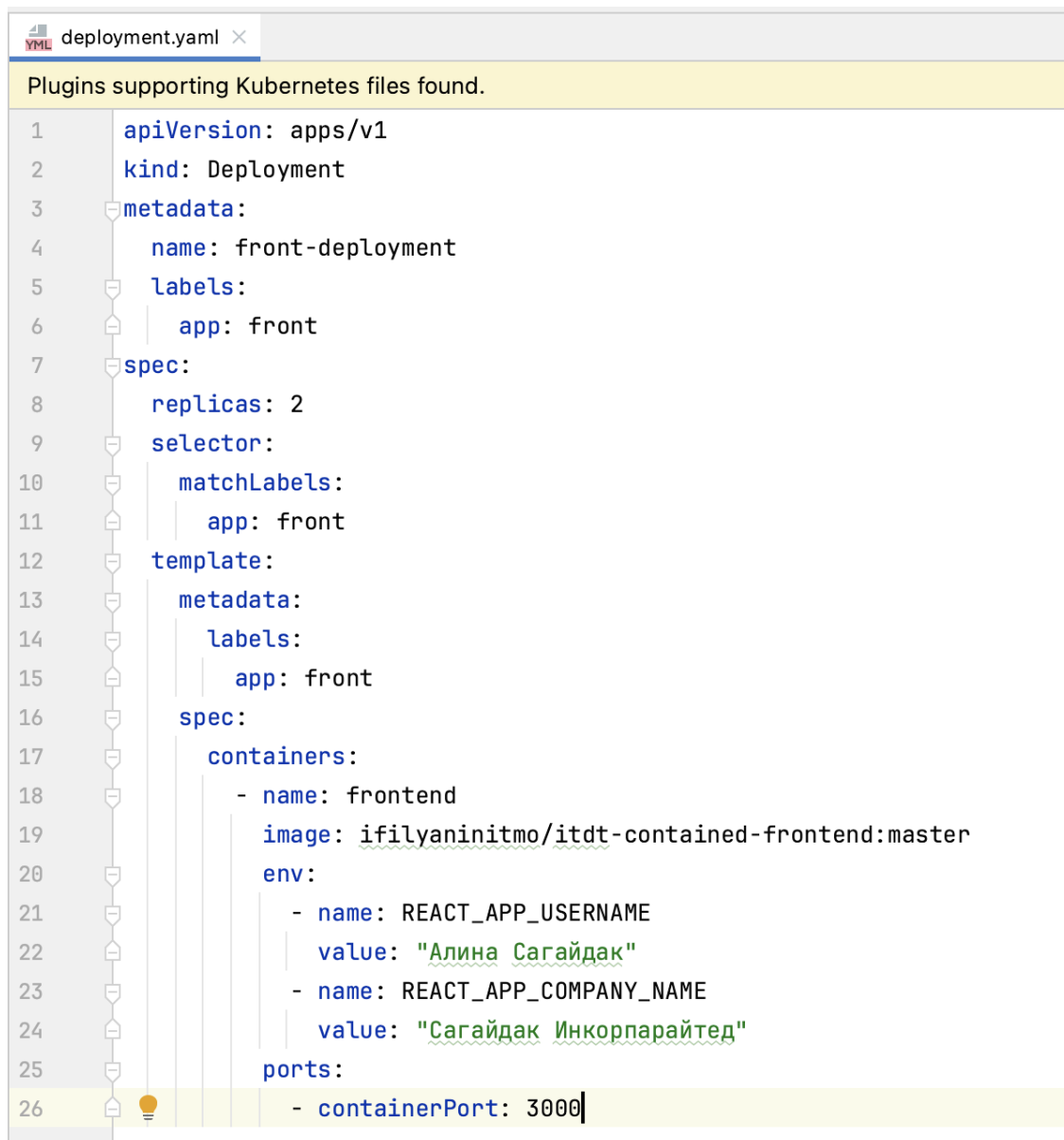
Ход работы

1. Вам необходимо создать **deployment** с 2 репликами контейнера **ifilyaninitmo/itdt-contained-frontend:master** и передать переменные в эти реплики:
REACT_APP_USERNAME,
REACT_APP_COMPANY_NAME.
2. Создать сервис через который у вас будет доступ на эти "поды". Выбор типа сервиса остается на ваше усмотрение.
3. Запустить в **minikube** режим проброса портов и подключиться к вашим контейнерам через веб браузер.
4. Проверьте на странице в веб браузере переменные **REACT_APP_USERNAME,**
REACT_APP_COMPANY_NAME и **Container name.**
Изменяются ли они? Если да то почему?
5. Проверьте логи контейнеров, приложите логи в отчёт.
6. Схема организации контейнеров и сервисов нарисованная вами в **draw.io** или Visio.

Выполнение работы

1. Создание **deployment** с 2 репликами контейнера **ifilyaninitmo/itdt-contained-frontend:master** и передать переменные в эти реплики: **REACT_APP_USERNAME,**
REACT_APP_COMPANY_NAME.

Для начала необходимо описать manifest деплоймента
Поменялся kind - абстракция сущности, добавились поля,
отвечающие за количество подов, а также за переменные
запуска



```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: front-deployment
5    labels:
6      app: front
7  spec:
8    replicas: 2
9    selector:
10     matchLabels:
11       app: front
12    template:
13     metadata:
14       labels:
15         app: front
16     spec:
17       containers:
18         - name: frontend
19           image: ifilyaninitmo/itdt-contained-frontend:master
20           env:
21             - name: REACT_APP_USERNAME
22               value: "Алина Сагайдак"
23             - name: REACT_APP_COMPANY_NAME
24               value: "Сагайдак Инкорпарайтед"
25           ports:
26             - containerPort: 3000
```

Создаем объект Deployment с помощью команды:
minikube kubectl -- apply -f deployment.yaml

```
[asagaydak@alinas-air lab2 % minikube kubectl -- apply -f deployment.yaml]
deployment.apps/front-deployment created
```

и проверяем созданные поды и деплоймент

```
[asagaydak@alinas-air lab2 % minikube kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
front-deployment-57d7c598fb-dwshp   1/1     Running   0           95s
front-deployment-57d7c598fb-svz7p   1/1     Running   0           95s
[asagaydak@alinas-air lab2 % minikube kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
front-deployment    2/2     2             2           101s
```

2. Создать сервис через который у вас будет доступ на эти "поды". Выбор типа сервиса остается на ваше усмотрение.

Создаем сервис с помощью команды:

```
minikube kubectl -- expose deployment front-deployment
--port=3000 --name=front-service --type=ClusterIP
```

Тип сервиса указан ClusterIP, данный тип означает, что сервис будет доступен только внутри кластера Minikube.

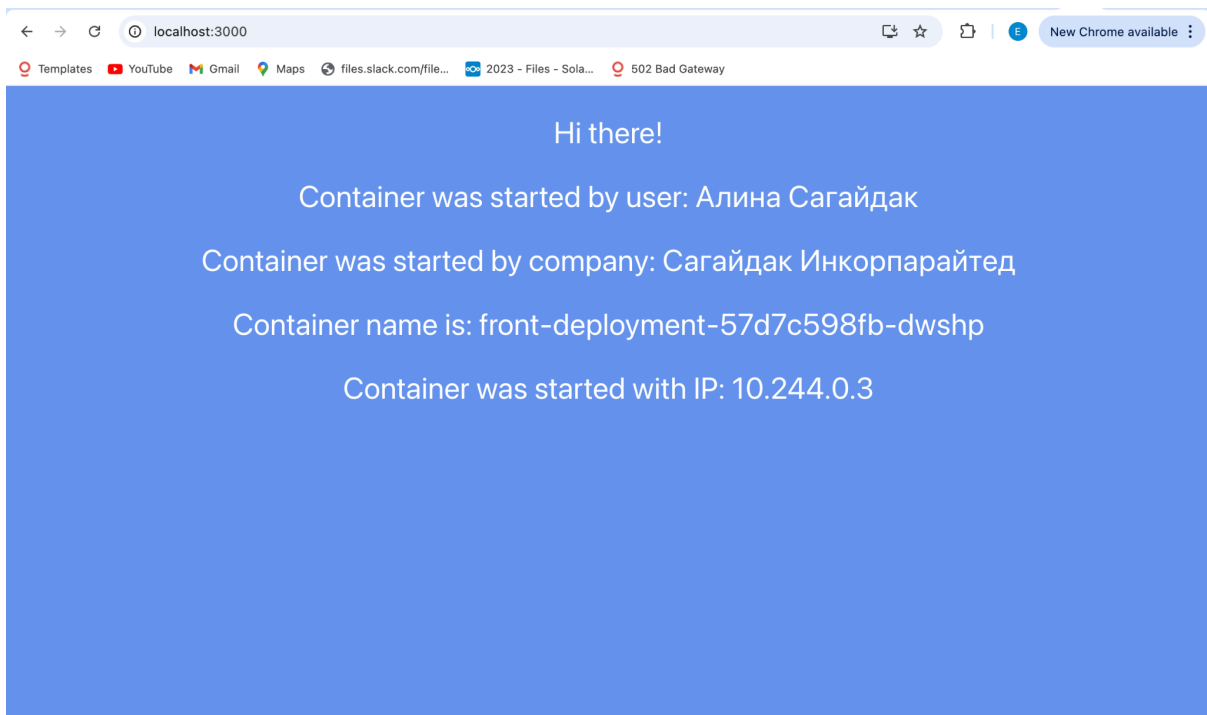
```
[asagaydak@alinas-air lab2 % minikube kubectl -- expose deployment front-deployment --port=3000 --name=front-service --type=ClusterIP
service/front-service exposed
```

3. Запустить в **minikube** режим проброса портов и подключиться к вашим контейнерам через веб браузер.

Для того, чтобы кластер мог получать запросы извне (например, с порта localhost:3000), мы должны перенаправить эти запросы на порт в рамках minikube cluster

```
[asagaydak@alinas-air lab2 % minikube kubectl -- port-forward service/front-service 3000:3000
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
Handling connection for 3000
```

4. Проверьте на странице в веб браузере переменные **REACT_APP_USERNAME**, **REACT_APP_COMPANY_NAME** и **Container name**. Изменяются ли они? Если да то почему?



Как видим переменные `REACT_APP_USERNAME` и `REACT_APP_COMPANY_NAME` не изменились, так как эти переменные окружения заданы в манифесте при создании объекта деплоймент. А переменная `Container name` изменяется в зависимости от того, какой под обрабатывает запрос.

5. Проверьте логи контейнеров, приложите логи в отчёт.

Логи подов представлены ниже:

```
[asagaydak@alinas-air lab2 % minikube kubectl logs front-deployment-57d7c598fb-dwshp
Building frontend
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
build finished
Server started on port 3000
[asagaydak@alinas-air lab2 % minikube kubectl logs front-deployment-57d7c598fb-svz7p
Building frontend
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
build finished
Server started on port 3000
```

6. Схема организации контейнеров и сервисов нарисованная вами в draw.io или Visio.

Схема прохождения запроса к подам

