Система

- minikube v1.36.0 on Ubuntu 24.04

- minikube запускается с --device=docker

Развертывание кластера через команду

```
./deploy.sh
```

(подразумевает локальную установку утилиты istioctl)

ДЗ2:

Финальные тесты для проверки работы Istio-конфигураций:

1. Проверка установки Istio

```
# Проверим версию Istio и статус компонентов
istioctl version
kubectl -n istio-system get pods
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ # Проверим версию Istio и статус компонентов
istioctl version
minikube kubectl -- -n istio-system get pods
client version: 1.18.0
control plane version: 1.18.0
data plane version: 1.18.0 (6 proxies)
NAME                                      READY   STATUS    RESTARTS   AGE
istio-egressgateway-69c59fffbc-x8vhl      1/1     Running   0          2m
istio-ingressgateway-56d44ff5ff-wg7jh     1/1     Running   0          2m
istiod-7d5b649b57-flvhs                   1/1     Running   0          2m26s
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

2. Проверка Gateway и базовой маршрутизации

```
# Получим параметры доступа
export INGRESS_PORT=$(kubectl -n istio-system get service istio-
ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}')
export INGRESS_HOST=$(minikube ip)

# Тест основных эндпоинтов
curl -v http://$INGRESS_HOST:$INGRESS_PORT/
curl -v http://$INGRESS_HOST:$INGRESS_PORT/status
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ export INGRESS_PORT=$(minikube kubectl -- -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")
].nodePort}')
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ export INGRESS_HOST=$(minikube ip)
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ curl -v http://$INGRESS_HOST:$INGRESS_PORT/
curl -v http://$INGRESS_HOST:$INGRESS_PORT/status
*   Trying 192.168.49.2:32528...
* Connected to 192.168.49.2 (192.168.49.2) port 32528
> GET / HTTP/1.1
> Host: 192.168.49.2:32528
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< content-type: text/html; charset=utf-8
< content-length: 25
< server: istio-envoy
< date: Sun, 25 May 2025 21:53:15 GMT
< x-envoy-upstream-service-time: 7
<
* Connection #0 to host 192.168.49.2 left intact
Welcome to the custom app*   Trying 192.168.49.2:32528...
* Connected to 192.168.49.2 (192.168.49.2) port 32528
> GET /status HTTP/1.1
> Host: 192.168.49.2:32528
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< content-type: application/json
< content-length: 57
< server: istio-envoy
< date: Sun, 25 May 2025 21:53:15 GMT
< x-envoy-upstream-service-time: 1
<
{"hostname":"custom-app-57798d57c8-n77nq","status":"ok"}
* Connection #0 to host 192.168.49.2 left intact
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ []
```

3. Проверка обработки неизвестных маршрутов

```
curl -v http://$INGRESS_HOST:$INGRESS_PORT/wrong-url
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ curl -v http://$INGRESS_HOST:$INGRESS_PORT/wrong-url
*   Trying 192.168.49.2:32528...
* Connected to 192.168.49.2 (192.168.49.2) port 32528
> GET /wrong-url HTTP/1.1
> Host: 192.168.49.2:32528
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 404 Not Found
< content-length: 9
< content-type: text/plain
< date: Sun, 25 May 2025 21:54:46 GMT
< server: istio-envoy
<
* Connection #0 to host 192.168.49.2 left intact
Not Found(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

4. Проверка балансировки нагрузки (LEAST_CONN)

```
# Запустим 10 последовательных запросов к /status
for i in {1..10}; do
  curl -s http://$INGRESS_HOST:$INGRESS_PORT/status | jq .hostname
done
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ for i in {1..10}; do..10}; do
  curl -s http://$INGRESS_HOST:$INGRESS_PORT/status | jq .hostname
done
"custom-app-57798d57c8-n77nq"
"custom-app-57798d57c8-xnlhs"
"custom-app-57798d57c8-xnlhs"
"custom-app-57798d57c8-nw94v"
"custom-app-57798d57c8-n77nq"
"custom-app-57798d57c8-n77nq"
"custom-app-57798d57c8-xnlhs"
"custom-app-57798d57c8-nw94v"
"custom-app-57798d57c8-nw94v"
"custom-app-57798d57c8-xnlhs"
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

PROF

5. Проверка логов Istio-прокси

```
minikube kubectl -- logs $POD_NAME -c istio-proxy | grep -E
'(timeout|retry)'
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ minikube kubectl -- logs $POD_NAME -c istio-proxy | grep -E '(timeout|retry)'
2025-05-25T21:50:52.190760Z     info    FLAG: --s2a_timeout="3s"
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

6. Полная проверка системы

```
# Проверим все ключевые компоненты
minikube kubectl -- get gateway,virtualservice,destinationrule
minikube kubectl -- get pods -l app=custom-app
minikube kubectl -- get pods -l app=log-agent
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ minikube kubectl get gateway,virtualservice,destinationrule
NAME                                               AGE
gateway.networking.istio.io/custom-gateway      9m57s

NAME                                               GATEWAYS           HOSTS    AGE
virtualservice.networking.istio.io/custom-app-vs   ["custom-gateway"] ["*"]    9m57s

NAME                                               HOST                                       AGE
destinationrule.networking.istio.io/custom-app-dr  custom-app-svc.default.svc.cluster.local   9m57s
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ minikube kubectl -- get pods -l app=custom-app
NAME                         READY   STATUS    RESTARTS   AGE
custom-app-57798d57c8-n77nq  2/2     Running   0          10m
custom-app-57798d57c8-nw94v  2/2     Running   0          10m
custom-app-57798d57c8-xnlhs  2/2     Running   0          10m
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```

```
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$ minikube kubectl -- get pods -l app=log-agent
NAME             READY   STATUS    RESTARTS   AGE
log-agent-crdkg  2/2     Running   0          11m
(base) badmotherfucker@BadMotherFuckerPC:~/study/test_4_ppprpo$
```