

```
vi deployment_probes.yaml
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-probe
spec:
  replicas: 1
  selector:
    matchLabels:
      app: app-probe
  template:
    metadata:
      labels:
        app: app-probe
    spec:
      containers:
      - image: nginx:1.20
        name: nginx
        ports:
        - containerPort: 80
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /
            port: 80
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 60
        livenessProbe:
          failureThreshold: 3
          httpGet:
            path: /
            port: 80
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 60
          initialDelaySeconds: 10
:WQ
```

```
kubectl apply -f deployment_probes.yaml
```

```
kubectl get pods
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ vi deployment_probes.yaml
igor@ubuntu-server:~$ kubectl apply -f deployment_probes.yaml
deployment.apps/app-probe created
igor@ubuntu-server:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
app-probe-f775c7b8f-866hw           1/1     Running   0           17s
database-0                           0/1     Pending   0           130d
geekbrains-55df48c88d-klqx7         1/1     Running   9 (25h ago)  130d
geekbrains-55df48c88d-rz2qz         1/1     Running   9 (25h ago)  130d
gitlab-runner-7467648488-jmlsl      0/1     Running   139 (7m6s ago)  130d
my-deployment-76499ffb7-5cj6v       1/1     Running   1 (25h ago)  25h
my-deployment-76499ffb7-5jcmc       1/1     Running   1 (25h ago)  25h
my-replicaset-47chw                 1/1     Running   1 (25h ago)  25h
my-replicaset-rcb4c                 1/1     Running   1 (25h ago)  25h
my-replicaset-xqjt7                 1/1     Running   1 (25h ago)  25h
node-exporter-n5xqk                 1/1     Running   12 (25h ago)  134d
prometheus-0                        0/1     Pending   0           134d
web-7695dfdcdbd-ftsdg               0/1     Pending   0           61m
web-7695dfdcdbd-mlnfd               0/1     Pending   0           61m
igor@ubuntu-server:~$
```

```
kubectl describe pod app-probe-f775c7b8f-866hw
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ kubectl describe pod app-probe-f775c7b8f-866hw
Name:          app-probe-f775c7b8f-866hw
Namespace:     default
Priority:       0
Node:          minikube/192.168.49.2
Start Time:    Sat, 26 Mar 2022 21:02:19 +0000
Labels:        app=app-probe
               pod-template-hash=f775c7b8f
Annotations:   <none>
Status:        Running
IP:            172.17.0.18
IPs:           IP: 172.17.0.18
Controlled By: ReplicaSet/app-probe-f775c7b8f
Containers:
  nginx:
    Container ID:  docker://992a2b0c4093a3c0a02b233b6e8d31b50be9033acced280bcfafa81b030439fe
    Image:         nginx:1.20
    Image ID:      docker-pullable://nginx@sha256:70ffa5088f7a3be061ae2b0816abc36f52db185c023c4bc039d7fb8fe490fcf8
    Port:          80/TCP
    Host Port:     0/TCP
    State:         Running
      Started:     Sat, 26 Mar 2022 21:02:31 +0000
    Ready:         True
    Restart Count: 0
    Liveness:       http-get http://:80/ delay=10s timeout=60s period=10s #success=1 #failure=3
    Readiness:      http-get http://:80/ delay=0s timeout=60s period=10s #success=1 #failure=3
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-mr7hb (ro)
Conditions:
  Type            Status
  Initialized     True
  Ready           True
  ContainersReady True
  PodScheduled    True
Volumes:
  kube-api-access-mr7hb:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:       kube-root-ca.crt
    ConfigMapOptional:   <nil>
    DownwardAPI:         true
    QoS Class:           BestEffort
    Node-Selectors:      <none>
    Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                        node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   90s   default-scheduler   Successfully assigned default/app-probe-f775c7b8f-866hw to minikube
  Normal   Pulling    90s   kubelet         Pulling image "nginx:1.20"
  Normal   Pulled     81s   kubelet         Successfully pulled image "nginx:1.20" in 8.437542896s
  Normal   Created    79s   kubelet         Created container nginx
  Normal   Started    79s   kubelet         Started container nginx
igor@ubuntu-server:~$
```

vi service_clusterip.yaml

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: v1
kind: Service
metadata:
  name: my-iservice
spec:
  ports:
  - port: 8080
    targetPort: 80
  selector:
    app: app-probe
  type: ClusterIP
~
:wq
```

```
kubectl create -f service_clusterip.yaml
```

```
kubectl get services
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl create -f service_clusterip.yaml
service/my-iservice created
igor@ubuntu-server:~$ kubectl get services
NAME          TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
database      ClusterIP    10.107.154.203 <none>       5432/TCP    130d
geekbrains     ClusterIP    10.109.211.170 <none>       8080/TCP    130d
kubernetes     ClusterIP    10.96.0.1      <none>       443/TCP     145d
my-iservice    ClusterIP    10.106.23.4    <none>       8080/TCP    17s
prometheus     ClusterIP    10.105.48.84   <none>       80/TCP      134d
igor@ubuntu-server:~$
```

```
kubectl describe service my-iservice
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl describe service my-iservice
Name:          my-iservice
Namespace:     default
Labels:        <none>
Annotations:    <none>
Selector:      app=app-probe
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.106.23.4
IPs:           10.106.23.4
Port:          <unset> 8080/TCP
TargetPort:    80/TCP
Endpoints:     172.17.0.18:80
Session Affinity: None
Events:        <none>
igor@ubuntu-server:~$
```

Для выполнения домашнего задания

```
minikube addons enable ingress
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ minikube addons enable ingress
  ■ Using image k8s.gcr.io/ingress-nginx/controller:v1.0.0-beta.3
  ■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
  ■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
  Verifying ingress addon...
  ✨ The 'ingress' addon is enabled
igor@ubuntu-server:~$
```

```
kubectl get pods
```

```
minikube version
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
app-probe-f775c7b8f-866hw          1/1     Running   0           21m
database-0                          0/1     Pending   0           130d
geekbrains-55df48c88d-klqx7        1/1     Running   9 (25h ago) 130d
geekbrains-55df48c88d-rz2qz        1/1     Running   9 (25h ago) 130d
gitlab-runner-7467648488-jmlsl     0/1     Running   142 (5m8s ago) 130d
my-deployment-76499ffb7-5cj6v      1/1     Running   1 (25h ago) 25h
my-deployment-76499ffb7-5jcmc      1/1     Running   1 (25h ago) 25h
my-replicaset-47chw                1/1     Running   1 (25h ago) 26h
my-replicaset-rcb4c                1/1     Running   1 (25h ago) 26h
my-replicaset-xqjt7                1/1     Running   1 (25h ago) 26h
node-exporter-n5xqk                1/1     Running   12 (25h ago) 134d
prometheus-0                       0/1     Pending   0           134d
web-7695dfdcdbd-ftsdg              0/1     Pending   0           82m
web-7695dfdcdbd-mlnfd              0/1     Pending   0           82m

igor@ubuntu-server:~$ minikube version
minikube version: v1.23.2
commit: 0a0ad764652082477c00d51d2475284b5d39ceed
igor@ubuntu-server:~$
```

```
kubectl get pods -n ingress-nginx
```

```
kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0
```

```
error: required flag(s) "image" not set
```

```
kubectl create deployment web2
--image=gcr.io/google-samples/hello-app:1.0
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-admission-create--1-5z22r  0/1     Completed   0           138d
ingress-nginx-admission-patch--1-sgdmc    0/1     Completed   1           138d
ingress-nginx-controller-69bdbc4d57-z67vb 1/1     Running     13 (25h ago) 138d

igor@ubuntu-server:~$ kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0
error: required flag(s) "image" not set
igor@ubuntu-server:~$ kubectl create deployment web2 --image=gcr.io/google-samples/hello-app:1.0
error: required flag(s) "image" not set
igor@ubuntu-server:~$ kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0
error: failed to create deployment: deployments.apps "web" already exists
igor@ubuntu-server:~$ kubectl create deployment web2 --image=gcr.io/google-samples/hello-app:1.0
deployment.apps/web2 created
igor@ubuntu-server:~$
```

```
kubectl expose deployment web2 --type=NodePort --port=8080
```

```
kubectl get services
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ kubectl expose deployment web2 --type=NodePort --port=8080
service/web2 exposed
igor@ubuntu-server:~$ kubectl get services
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
database  ClusterIP   10.107.154.203   <none>       5432/TCP         130d
geekbrains ClusterIP   10.109.211.170   <none>       8080/TCP         130d
kubernetes ClusterIP   10.96.0.1         <none>       443/TCP          145d
my-iservice ClusterIP   10.106.23.4       <none>       8080/TCP         30m
prometheus ClusterIP   10.105.48.84      <none>       80/TCP           134d
web2      NodePort    10.106.200.161   <none>       8080:31153/TCP  32s

igor@ubuntu-server:~$
```

```
minikube service web2 --url
```

```
http://192.168.49.2:31153
```

```
curl http://192.168.49.2:31153
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ minikube service web2 --url
http://192.168.49.2:31153
igor@ubuntu-server:~$ curl http://192.168.49.2:31153
Hello, world!
Version: 1.0.0
Hostname: web2-5b669f8984-86wvp
igor@ubuntu-server:~$
```

```
vi hw.yaml
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
  annotation:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: hello-world.info
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web2
            port:
              number: 8080
~
:wq
```

```
kubectl apply -f hw.yaml
```

```
error: error parsing hw.yaml: error converting YAML to JSON: yaml:
line 13: mapping values are not allowed in this context
```

```
wget https://k8s.io/examples/service/networking/example-
ingress.yaml
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@ubuntu-server:~$ wget https://k8s.io/examples/service/networking/example-ingress.yaml
--2022-03-26 21:49:01-- https://k8s.io/examples/service/networking/example-ingress.yaml
Resolving k8s.io (k8s.io)... 2600:1901:0:26f3::, 34.107.204.206
Connecting to k8s.io (k8s.io)|2600:1901:0:26f3::|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://kubernetes.io/examples/service/networking/example-ingress.yaml [following]
--2022-03-26 21:49:02-- https://kubernetes.io/examples/service/networking/example-ingress.yaml
Resolving kubernetes.io (kubernetes.io)... 147.75.40.148
Connecting to kubernetes.io (kubernetes.io)|147.75.40.148|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 390 [application/x-yaml]
Saving to: 'example-ingress.yaml'

example-ingress.yaml      100%[=====>]      390  --.-KB/s   in 0s

2022-03-26 21:49:03 (37,7 MB/s) - 'example-ingress.yaml' saved [390/390]

igor@ubuntu-server:~$
```

vi example-ingress.yaml

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: hello-world.info
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web
            port:
              number: 8080

1,1  Весь
```

kubectl apply -f example-ingress.yaml

ingress.networking.k8s.io/example-ingress created

kubectl get ingress

```
igor@ubuntu-server:~$ kubectl apply -f example-ingress.yaml
ingress.networking.k8s.io/example-ingress created
igor@ubuntu-server:~$ kubectl get ingress
NAME          CLASS  HOSTS          ADDRESS          PORTS  AGE
example-ingress  nginx  hello-world.info  192.168.49.2    80     2m7s
prometheus     nginx  *              192.168.49.2    80     134d

igor@ubuntu-server:~$
```

vi example-ingress.yaml


```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-2
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: hello-world2.info
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web2
            port:
              number: 8080
~
:wq
```

kubectl apply -f example-ingress.yaml

ingress.networking.k8s.io/ingress-2 created

kubectl get ingress

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl apply -f example-ingress.yaml
ingress.networking.k8s.io/ingress-2 created
igor@ubuntu-server:~$ kubectl get ingress
NAME          CLASS    HOSTS          ADDRESS          PORTS    AGE
example-ingress  nginx    hello-world.info  192.168.49.2    80      8m20s
ingress-2      nginx    hello-world2.info  192.168.49.2    80      60s
prometheus     nginx    *              192.168.49.2    80      134d
igor@ubuntu-server:~$
```

curl hello-world2.info

curl: (6) Could not resolve host: hello-world2.info

kubectl describe ingress ingress-2

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ curl hello-world2.info
curl: (6) Could not resolve host: hello-world2.info
igor@ubuntu-server:~$ kubectl describe ingress ingress-2
Name:          ingress-2
Labels:         <none>
Namespace:     default
Address:        192.168.49.2
Default backend: default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host          Path  Backends
  ----          -
  hello-world2.info  /    web2:8080 (172.17.0.19:8080)
Annotations:    nginx.ingress.kubernetes.io/rewrite-target: /$1
Events:
  Type    Reason    Age          From          Message
  ----    -
  Normal  Sync      2m50s (x2 over 3m35s)  nginx-ingress-controller  Scheduled for sync
igor@ubuntu-server:~$
```

172.17.0.19

vi example-ingress.yaml

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-2
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: hello-world2.info
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web2
            port:
              number: 80
:wq
```

kubectl get services

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl get services
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
database      ClusterIP      10.107.154.203   <none>           5432/TCP         130d
geekbrains     ClusterIP      10.109.211.170   <none>           8080/TCP         130d
kubernetes     ClusterIP      10.96.0.1        <none>           443/TCP          145d
my-iservice    ClusterIP      10.106.23.4      <none>           8080/TCP         61m
prometheus     ClusterIP      10.105.48.84     <none>           80/TCP           134d
web2           NodePort       10.106.200.161   <none>           8080:31153/TCP   32m
igor@ubuntu-server:~$
```

vi example-ingress.yaml

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-2
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
  - host: hello-world2.info
    http:
      paths:
      - path: /
        pathType: Prefix
        defaultBackend:
          service:
            name: web2
            port:
              number: 8080
~
:wq
```

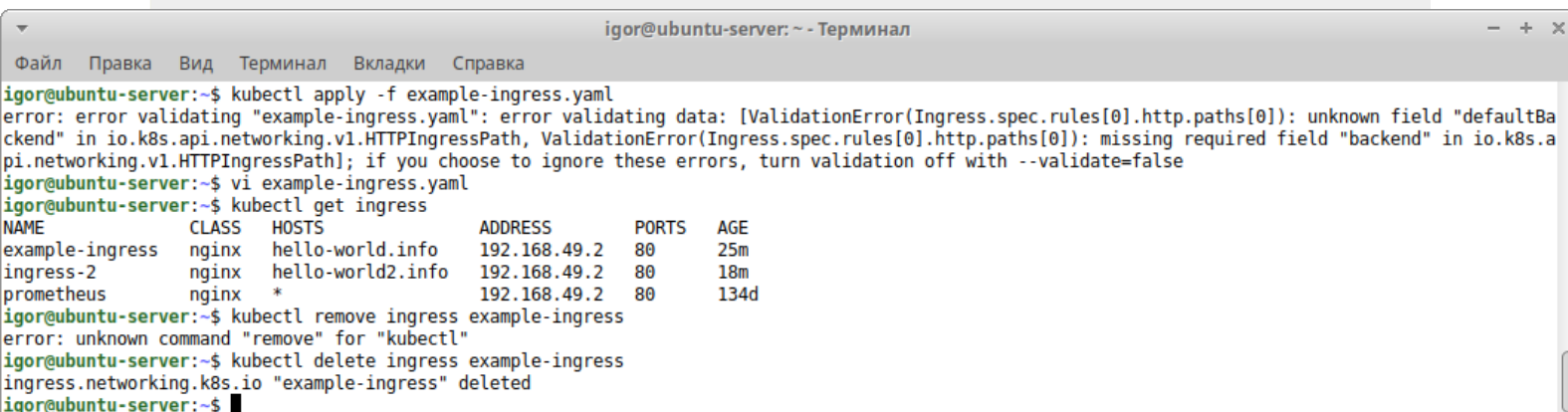
kubectl apply -f example-ingress.yaml


```
error: error validating "example-ingress.yaml": error validating data: [ValidationError(Ingress.spec.rules[0].http.paths[0]): unknown field "defaultBackend" in io.k8s.api.networking.v1.HTTPIngressPath, ValidationError(Ingress.spec.rules[0].http.paths[0]): missing required field "backend" in io.k8s.api.networking.v1.HTTPIngressPath]; if you choose to ignore these errors, turn validation off with --validate=false
```

```
vi example-ingress.yaml
```

```
kubectl get ingress
```

```
kubectl delete ingress example-ingress
```

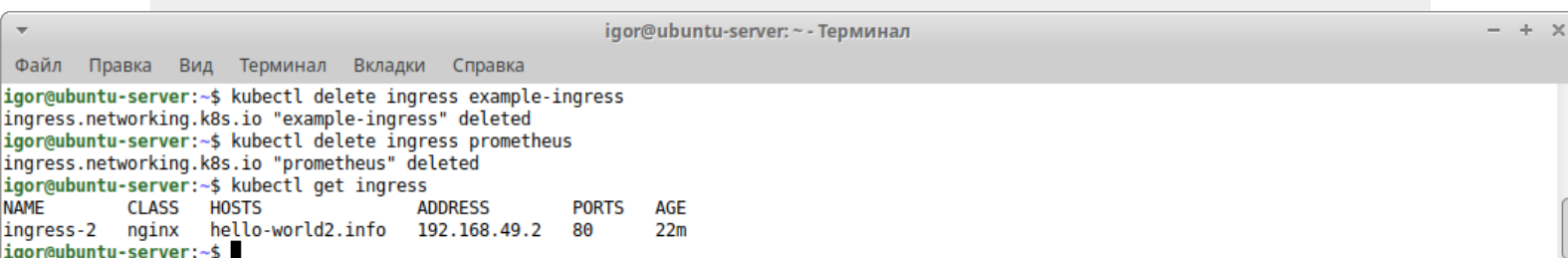


```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl apply -f example-ingress.yaml
error: error validating "example-ingress.yaml": error validating data: [ValidationError(Ingress.spec.rules[0].http.paths[0]): unknown field "defaultBackend" in io.k8s.api.networking.v1.HTTPIngressPath, ValidationError(Ingress.spec.rules[0].http.paths[0]): missing required field "backend" in io.k8s.api.networking.v1.HTTPIngressPath]; if you choose to ignore these errors, turn validation off with --validate=false
igor@ubuntu-server:~$ vi example-ingress.yaml
igor@ubuntu-server:~$ kubectl get ingress
NAME          CLASS    HOSTS          ADDRESS          PORTS    AGE
example-ingress  nginx    hello-world.info  192.168.49.2    80      25m
ingress-2      nginx    hello-world2.info  192.168.49.2    80      18m
prometheus     nginx    *              192.168.49.2    80      134d
igor@ubuntu-server:~$ kubectl remove ingress example-ingress
error: unknown command "remove" for "kubectl"
igor@ubuntu-server:~$ kubectl delete ingress example-ingress
ingress.networking.k8s.io "example-ingress" deleted
igor@ubuntu-server:~$
```

```
kubectl delete ingress example-ingress
```

```
kubectl delete ingress prometheus
```

```
kubectl get ingress
```



```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl delete ingress example-ingress
ingress.networking.k8s.io "example-ingress" deleted
igor@ubuntu-server:~$ kubectl delete ingress prometheus
ingress.networking.k8s.io "prometheus" deleted
igor@ubuntu-server:~$ kubectl get ingress
NAME          CLASS    HOSTS          ADDRESS          PORTS    AGE
ingress-2      nginx    hello-world2.info  192.168.49.2    80      22m
igor@ubuntu-server:~$
```

```
kubectl describe ingress ingress-2
```

```
curl 172.17.0.19:8080
```

```
curl: (7) Failed to connect to 172.17.0.19 port 8080: No route to host
```

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl describe ingress ingress-2
Name:          ingress-2
Labels:        <none>
Namespace:     default
Address:       192.168.49.2
Default backend: default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host          Path  Backends
  ----          -
hello-world2.info
                /   web2:8080 (172.17.0.19:8080)
Annotations:   nginx.ingress.kubernetes.io/rewrite-target: /$1
Events:
  Type    Reason    Age          From          Message
  ----    -
Normal   Sync      23m (x2 over 23m)  nginx-ingress-controller  Scheduled for sync
igor@ubuntu-server:~$ curl 172.17.0.19:8080
curl: (7) Failed to connect to 172.17.0.19 port 8080: No route to host
igor@ubuntu-server:~$
```