vi deployment_probes.yaml

```
---
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:1.12
        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
~
                                                          37,10          All
```

kubectl apply -f deployment_probes.yaml

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-89-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Sun Nov  7 18:00:24 2021 from 109.252.20.121
tolstikov@my3vm:~$ vi deployment_probes.yaml
tolstikov@my3vm:~$ kubectl apply -f deployment_probes.yaml
deployment.apps/app-probe created
tolstikov@my3vm:~$
```
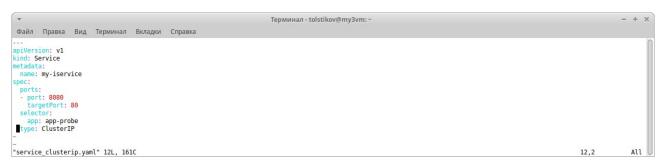
kubectl get pods

```
tolstikov@my3vm:~$ kubectl apply -f deployment_probes.yaml
deployment.apps/app-probe created
tolstikov@my3vm:~$ vi deployment_probes.yaml
tolstikov@my3vm:~$ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
app-probe-f775c7b8f-prn4r    1/1     Running   0          4m52s
pg-db-ff476d4cf-xkxhn        1/1     Running   0          42m
tolstikov@my3vm:~$ kubectl describe pod app-probe-f775c7b8f-prn4r
```

kubectl describe pod app-probe-f775c7b8f-prn4r

```
tolstikov@my3vm:~$ kubectl describe pod app-probe-f775c7b8f-prn4r
Name:         app-probe-f775c7b8f-prn4r
Namespace:    pg
Priority:     0
Node:         minikube/192.168.49.2
Start Time:   Sun, 07 Nov 2021 18:32:38 +0000
Labels:       app=app-probe
              pod-template-hash=f775c7b8f
Annotations:  <none>
Status:       Running
IP:           172.17.0.13
IPs:
  IP:           172.17.0.13
Controlled By:  ReplicaSet/app-probe-f775c7b8f
Containers:
  nginx:
    Container ID:   docker://a9abba1dfc518a66e91e4e2d1a8c10128963730cf532e85a66aba418305c35a0
    Image:          nginx:1.20
    Image ID:       docker-pullable://nginx@sha256:9e37120c97a0787656cad9c24c3a5d5d606a770523144d01eae6734095a92c20
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 07 Nov 2021 18:32:39 +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-dcsvd (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-dcsvd:
    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  10m    default-scheduler  Successfully assigned pg/app-probe-f775c7b8f-prn4r to minikube
  Normal   Pulled     10m    kubelet            Container image "nginx:1.20" already present on machine
  Normal   Created    10m    kubelet            Created container nginx
  Normal   Started    10m    kubelet            Started container nginx
  Warning  Unhealthy  10m    kubelet            Readiness probe failed: Get "http://172.17.0.13:80/": dial tcp 172.17.
0.13:80: connect: connection refused
tolstikov@my3vm:~$
```

vi service_clusterip.yaml

```
---
apiVersion: v1
kind: Service
metadata:
  name: my-iservice
spec:
  ports:
  - port: 8080
    targetPort: 80
  selector:
    app: app-probe
  type: ClusterIP
~
~
"service_clusterip.yaml" 12L, 161C                                           12,2          All
```

kubectl create -f service_clusterip.yaml

```
Events:
 Type     Reason    Age  From              Message
 ----     ------    ---  ----              -------
 Normal   Scheduled 10m  default-scheduler Successfully assigned pg/app-probe-f775c7b8f-prn4r to minikube
 Normal   Pulled    10m  kubelet           Container image "nginx:1.20" already present on machine
 Normal   Created   10m  kubelet           Created container nginx
 Normal   Started   10m  kubelet           Started container nginx
 Warning  Unhealthy 10m  kubelet           Readiness probe failed: Get "http://172.17.0.13:80/": dial tcp 172.17.
0.13:80: connect: connection refused
tolstikov@my3vm:~$ vi service_clusterip.yaml
tolstikov@my3vm:~$ kubectl create -f service_clusterip.yaml
service/my-iservice created
tolstikov@my3vm:~$
```

kubectl get services

```
 Normal   Scheduled 10m  default-scheduler Successfully assigned pg/app-probe-f775c7b8f-prn4r to minikube
 Normal   Pulled    10m  kubelet           Container image "nginx:1.20" already present on machine
 Normal   Created   10m  kubelet           Created container nginx
 Normal   Started   10m  kubelet           Started container nginx
 Warning  Unhealthy 10m  kubelet           Readiness probe failed: Get "http://172.17.0.13:80/": dial tcp 172.17.
0.13:80: connect: connection refused
tolstikov@my3vm:~$ vi service_clusterip.yaml
tolstikov@my3vm:~$ kubectl create -f service_clusterip.yaml
service/my-iservice created
tolstikov@my3vm:~$ kubectl get services
NAME          TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
my-iservice   ClusterIP  10.107.75.36   <none>        8080/TCP   97s
tolstikov@my3vm:~$
```

kubectl describe service my-iservice

```
tolstikov@my3vm:~$ kubectl describe service my-iservice
Name:              my-iservice
Namespace:         pg
Labels:            <none>
Annotations:       <none>
Selector:          app=app-probe
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.107.75.36
IPs:               10.107.75.36
Port:              <unset>  8080/TCP
TargetPort:        80/TCP
Endpoints:         172.17.0.13:80
Session Affinity:  None
Events:            <none>
tolstikov@my3vm:~$
```

minikube addons enable ingress

```
IP Families:       IPv4
IP:                10.107.178.92
IPs:               10.107.178.92
Port:              <unset>  8080/TCP
TargetPort:        80/TCP
Endpoints:         172.17.0.10:80
Session Affinity:  None
Events:            <none>
tolstikov@my3vm:~$ minikube addons enable ingress
    ▪ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
    ▪ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
    ▪ Using image k8s.gcr.io/ingress-nginx/controller:v1.0.0-beta.3
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
tolstikov@my3vm:~$
```
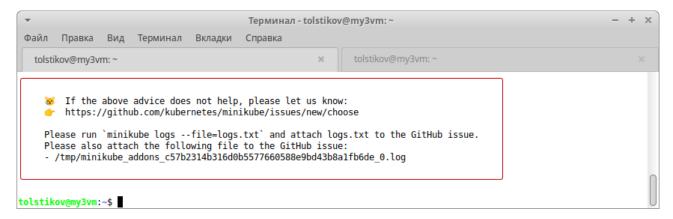
kubectl get pods

```
    ■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
    ■ Using image k8s.gcr.io/ingress-nginx/controller:v1.0.0-beta.3
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
tolstikov@my3vm:~$ kubectl get pods
NAME                              READY   STATUS    RESTARTS        AGE
app-probe-f775c7b8f-58g9w         1/1     Running   0               45m
hello-minikube-6ddfcc9757-lxjpj   1/1     Running   1 (3d13h ago)   3d15h
my-deployment2-76499ffb7-c2w4g    1/1     Running   0               27h
my-deployment2-76499ffb7-xzqqf    1/1     Running   0               27h
my-pod                            1/1     Running   0               28h
my-replicaset-qb674               1/1     Running   0               28h
my-replicaset-v5k2j               1/1     Running   0               28h
my-replicaset-wzfqm               1/1     Running   0               28h
tolstikov@my3vm:~$
```

kubectl get pods -n ingress-nginx

```
NAME                              READY   STATUS    RESTARTS        AGE
app-probe-f775c7b8f-58g9w         1/1     Running   0               45m
hello-minikube-6ddfcc9757-lxjpj   1/1     Running   1 (3d13h ago)   3d15h
my-deployment2-76499ffb7-c2w4g    1/1     Running   0               27h
my-deployment2-76499ffb7-xzqqf    1/1     Running   0               27h
my-pod                            1/1     Running   0               28h
my-replicaset-qb674               1/1     Running   0               28h
my-replicaset-v5k2j               1/1     Running   0               28h
my-replicaset-wzfqm               1/1     Running   0               28h
tolstikov@my3vm:~$ kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create--1-fxdbm     0/1     Completed   0          5m34s
ingress-nginx-admission-patch--1-kx587      0/1     Completed   0          5m34s
ingress-nginx-controller-69bdbc4d57-xwvpp   1/1     Running     0          5m35s
tolstikov@my3vm:~$
```

```
    😿  If the above advice does not help, please let us know:
    👉  https://github.com/kubernetes/minikube/issues/new/choose

    Please run `minikube logs --file=logs.txt` and attach logs.txt to the GitHub issue.
    Please also attach the following file to the GitHub issue:
    - /tmp/minikube_addons_c57b2314b316d0b5577660588e9bd43b8a1fb6de_0.log

tolstikov@my3vm:~$
```

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

```
hello-minikube-6ddfcc9757-lxjpj   1/1     Running   1 (3d13h ago)   3d15h
my-deployment2-76499ffb7-c2w4g    1/1     Running   0               27h
my-deployment2-76499ffb7-xzqqf    1/1     Running   0               27h
my-pod                            1/1     Running   0               28h
my-replicaset-qb674               1/1     Running   0               28h
my-replicaset-v5k2j               1/1     Running   0               28h
my-replicaset-wzfqm               1/1     Running   0               28h
tolstikov@my3vm:~$ kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create--1-fxdbm     0/1     Completed   0          5m34s
ingress-nginx-admission-patch--1-kx587      0/1     Completed   0          5m34s
ingress-nginx-controller-69bdbc4d57-xwvpp   1/1     Running     0          5m35s
tolstikov@my3vm:~$ kubectl create deployment web —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$
```

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

```
my-deployment2-76499ffb7-xzqqf    1/1     Running   0               27h
my-pod                            1/1     Running   0               28h
my-replicaset-qb674               1/1     Running   0               28h
my-replicaset-v5k2j               1/1     Running   0               28h
my-replicaset-wzfqm               1/1     Running   0               28h
tolstikov@my3vm:~$ kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create--1-fxdbm     0/1     Completed   0          5m34s
ingress-nginx-admission-patch--1-kx587      0/1     Completed   0          5m34s
ingress-nginx-controller-69bdbc4d57-xwvpp   1/1     Running     0          5m35s
tolstikov@my3vm:~$ kubectl create deployment web —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$ kubectl create deployment web2 —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$
```
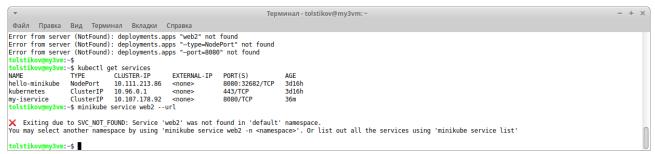
kubectl expose deployment web2 —type=NodePort —port=8080

```
Термина - tolstikov@my3vm:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
my-replicaset-wzfqm                 1/1    Running   0          28h
tolstikov@my3vm:~$ kubectl get pods -n ingress-nginx
NAME                                          READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create--1-fxdbm       0/1     Completed   0          5m34s
ingress-nginx-admission-patch--1-kx587        0/1     Completed   0          5m34s
ingress-nginx-controller-69bdbc4d57-xwvpp     1/1     Running     0          5m35s
tolstikov@my3vm:~$ kubectl create deployment web —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$ kubectl create deployment web2 —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$ kubectl expose deployment web2 —type=NodePort —port=8080
Error from server (NotFound): deployments.apps "web2" not found
Error from server (NotFound): deployments.apps "—type=NodePort" not found
Error from server (NotFound): deployments.apps "—port=8080" not found
tolstikov@my3vm:~$ 
```

kubectl get services

```
Термина - tolstikov@my3vm:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
tolstikov@my3vm:~$ kubectl create deployment web —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$ kubectl create deployment web2 —image=gcr.io/google-samples/hello-app:1.0
Error: required flag(s) "image" not set
tolstikov@my3vm:~$ kubectl expose deployment web2 —type=NodePort —port=8080
Error from server (NotFound): deployments.apps "web2" not found
Error from server (NotFound): deployments.apps "—type=NodePort" not found
Error from server (NotFound): deployments.apps "—port=8080" not found
tolstikov@my3vm:~$ 
tolstikov@my3vm:~$ kubectl get services
NAME            TYPE         CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube  NodePort     10.111.213.86    <none>        8080:32682/TCP   3d16h
kubernetes      ClusterIP    10.96.0.1        <none>        443/TCP          3d16h
my-iservice     ClusterIP    10.107.178.92    <none>        8080/TCP         36m
tolstikov@my3vm:~$ 
```

minikube service web2 —url

 Exiting due to SVC_NOT_FOUND: Service 'web2' was not found in 'default' namespace.
You may select another namespace by using 'minikube service web2 -n <namespace>'. Or list out all the services using 'minikube service list'

```
Термина - tolstikov@my3vm:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
Error from server (NotFound): deployments.apps "web2" not found
Error from server (NotFound): deployments.apps "—type=NodePort" not found
Error from server (NotFound): deployments.apps "—port=8080" not found
tolstikov@my3vm:~$ 
tolstikov@my3vm:~$ kubectl get services
NAME            TYPE         CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube  NodePort     10.111.213.86    <none>        8080:32682/TCP   3d16h
kubernetes      ClusterIP    10.96.0.1        <none>        443/TCP          3d16h
my-iservice     ClusterIP    10.107.178.92    <none>        8080/TCP         36m
tolstikov@my3vm:~$ minikube service web2 --url

✗   Exiting due to SVC_NOT_FOUND: Service 'web2' was not found in 'default' namespace.
You may select another namespace by using 'minikube service web2 -n <namespace>'. Or list out all the services using 'minikube service list'

tolstikov@my3vm:~$ 
```

Exiting due to MK_USAGE: You must specify a service name

```
Термина - tolstikov@my3vm:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
   tolstikov@my3vm: ~                              ✗    tolstikov@my3vm: ~                      ✗
tolstikov@my3vm:~$ kubectl expose deployment web2 —type=NodePort —port=8080
Error from server (NotFound): deployments.apps "web2" not found
Error from server (NotFound): deployments.apps "—type=NodePort" not found
Error from server (NotFound): deployments.apps "—port=8080" not found
tolstikov@my3vm:~$ kubectl get services
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
my-iservice   ClusterIP   10.107.75.36    <none>        8080/TCP    17m
tolstikov@my3vm:~$ minikube service web2 —url

✗   Exiting due to MK_USAGE: You must specify a service name

tolstikov@my3vm:~$ 
```

vi hw.yaml

```yaml
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: /
                            pathtape: Prefix
                            backend:
                                    service:
                                            name: web2
                                            port:
                                                    number: 8080
```

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


```
tolstikov@my3vm:~$ wget https://k8s.io/examples/service/networking/example-ingress.yaml
--2021-11-07 19:22:15--  https://k8s.io/examples/service/networking/example-ingress.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://kubernetes.io/examples/service/networking/example-ingress.yaml [following]
--2021-11-07 19:22:15--  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.1'

example-ingress.yaml.1      100%[===========================================>]     390  --.-KB/s    in 0s

2021-11-07 19:22:15 (85.9 MB/s) - 'example-ingress.yaml.1' saved [390/390]

tolstikov@my3vm:~$
```

vi example-ingress.yaml

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

kubectl apply -f example-ingress.yaml

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

kubectl apply -f example-ingress.yaml

kubectl get ingress

curl hello-world2.info

kubectl describe ingress ingress-2

```
tolstikov@my3vm:~$ kubectl get ingress
No resources found in pg namespace.
tolstikov@my3vm:~$ vi example-ingress.yaml
tolstikov@my3vm:~$ 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 t
hese errors, turn validation off with --validate=false
tolstikov@my3vm:~$ kubectl get ingress
No resources found in pg namespace.
tolstikov@my3vm:~$ curl hello-world2.info
curl: (6) Could not resolve host: hello-world2.info
tolstikov@my3vm:~$ curl hello-world3.info
curl: (6) Could not resolve host: hello-world3.info
tolstikov@my3vm:~$ kubectl describe ingress ingress-2
Error from server (NotFound): ingresses.networking.k8s.io "ingress-2" not found
tolstikov@my3vm:~$ kubectl describe ingress ingress-3
Error from server (NotFound): ingresses.networking.k8s.io "ingress-3" not found
tolstikov@my3vm:~$
```

vi example-ingress.yaml

```yaml
kind: Ingress
metadata:
  name: ingress-3
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
    - host: hello-world3.info
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: web3
                port:
                  number: 80

:wq
```

kubectl apply -f example-ingress.yaml

kubectl describe ingress ingress-3

kubectl get ingress

curl 192.168.49.2:80

Файл   Правка   Вид   Терминал   Вкладки   Справка

tolstikov@my3vm: ~   ✕   tolstikov@my3vm: ~   ✕

```
tolstikov@my3vm:~$ curl 192.168.49.2:80
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
tolstikov@my3vm:~$
```