

# Kubernetes

Дано:

- Приложение запущенное в docker

Задание:

- Запустить приложение в kubernetes с единой точкой входа и подключенным хранилищем для хранения данных mongodb.

Имею приложение запущенное в докер.  
Приложение состоит из сервисов:

- mongodb
- comment
- ui
- post

```
root@alex:/home/alex# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
f82f2059c4cb   mongo:latest   "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes  27017/tcp                          mongodb
be4a30229f79   r2d2k/comment  "puma"                   2 minutes ago  Up 2 minutes                               comment
588814f79356   r2d2k/ui       "puma"                   3 minutes ago  Up 3 minutes  0.0.0.0:9292->9292/tcp, :::9292->9292/tcp  ui
7eaa35160665   r2d2k/post     "python3 post_app.py"     16 minutes ago Up 16 minutes                               post
root@alex:/home/alex#
```

Приложение находится по ip адресу сервера и порту 9292

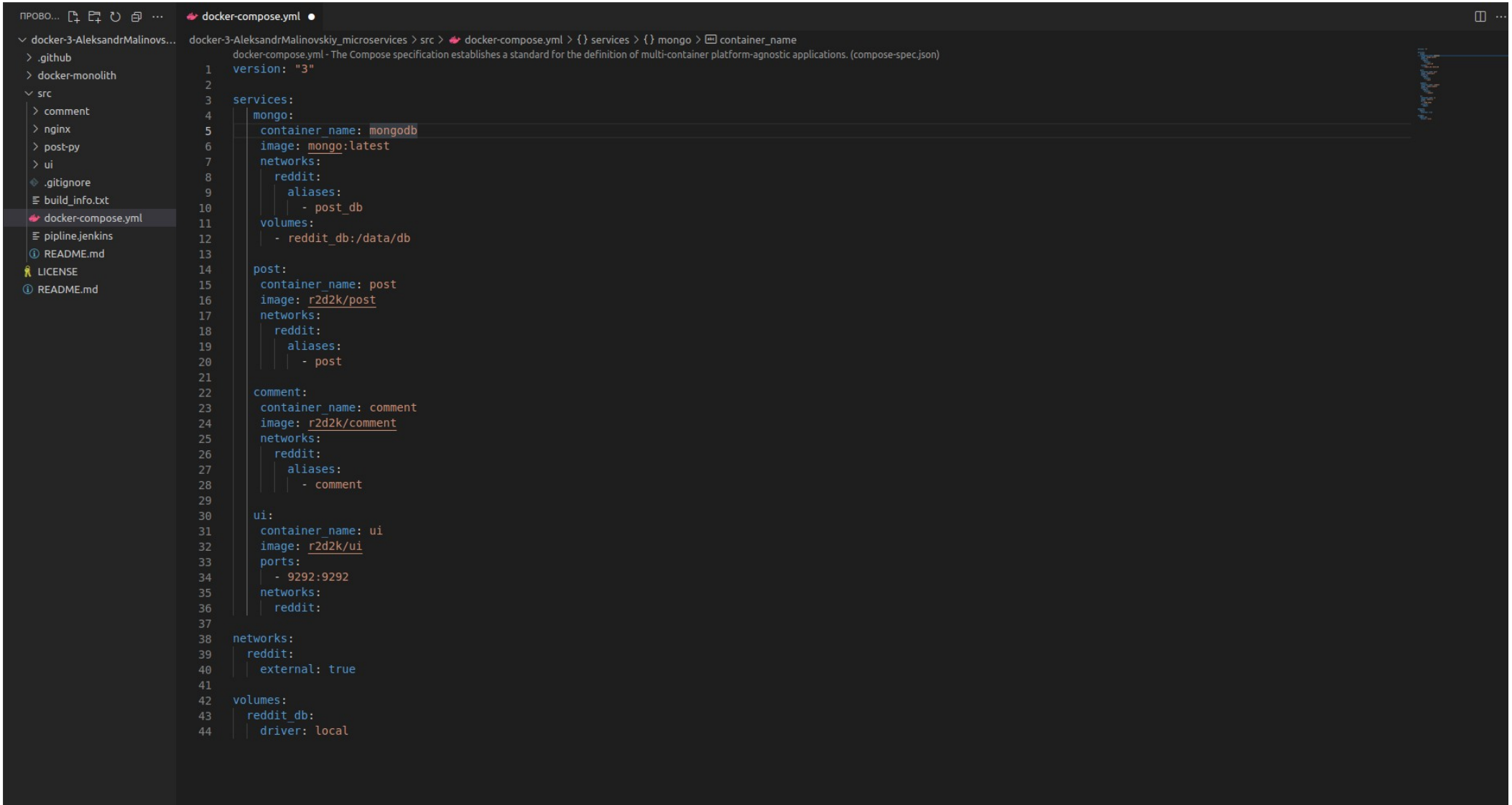
Microservices Reddit in 588814f79356 container

#### Menu

[All posts](#)

[New post](#)

# Docker-compose файл для запуска приложения в докер.

A screenshot of a code editor with a dark theme. On the left is a sidebar showing a file tree for a project named 'docker-3-AleksandrMalinovskiy\_microservices'. The tree includes folders like '.github', 'docker-monolith', 'src', and files like '.gitignore', 'build\_info.txt', 'docker-compose.yml' (which is selected and highlighted), 'pipeline.jenkins', 'README.md', and 'LICENSE'. The main editor area displays the content of 'docker-compose.yml'. The file starts with a version '3' and defines several services: 'mongo' (using 'mongo:latest' image, connected to a 'reddit' network, with an alias 'post\_db' and a volume 'reddit\_db:/data/db'), 'post' (using 'r2d2k/post' image, connected to the 'reddit' network, with an alias 'post'), 'comment' (using 'r2d2k/comment' image, connected to the 'reddit' network, with an alias 'comment'), and 'ui' (using 'r2d2k/ui' image, exposing port 9292, connected to the 'reddit' network). At the bottom, there is a 'networks' section defining the 'reddit' network as external, and a 'volumes' section defining the 'reddit\_db' volume as local. Line numbers 1 through 44 are visible on the left side of the code editor.

```
1 version: "3"
2
3 services:
4   mongo:
5     container_name: mongodb
6     image: mongo:latest
7     networks:
8       reddit:
9         aliases:
10          - post_db
11     volumes:
12       - reddit_db:/data/db
13
14   post:
15     container_name: post
16     image: r2d2k/post
17     networks:
18       reddit:
19         aliases:
20          - post
21
22   comment:
23     container_name: comment
24     image: r2d2k/comment
25     networks:
26       reddit:
27         aliases:
28          - comment
29
30   ui:
31     container_name: ui
32     image: r2d2k/ui
33     ports:
34       - 9292:9292
35     networks:
36       reddit:
37
38 networks:
39   reddit:
40     external: true
41
42 volumes:
43   reddit_db:
44     driver: local
```

Порядок действий для деплоя приложения в kubernetes.  
Для разграничения стендов разработки необходимо создать namespace dev.

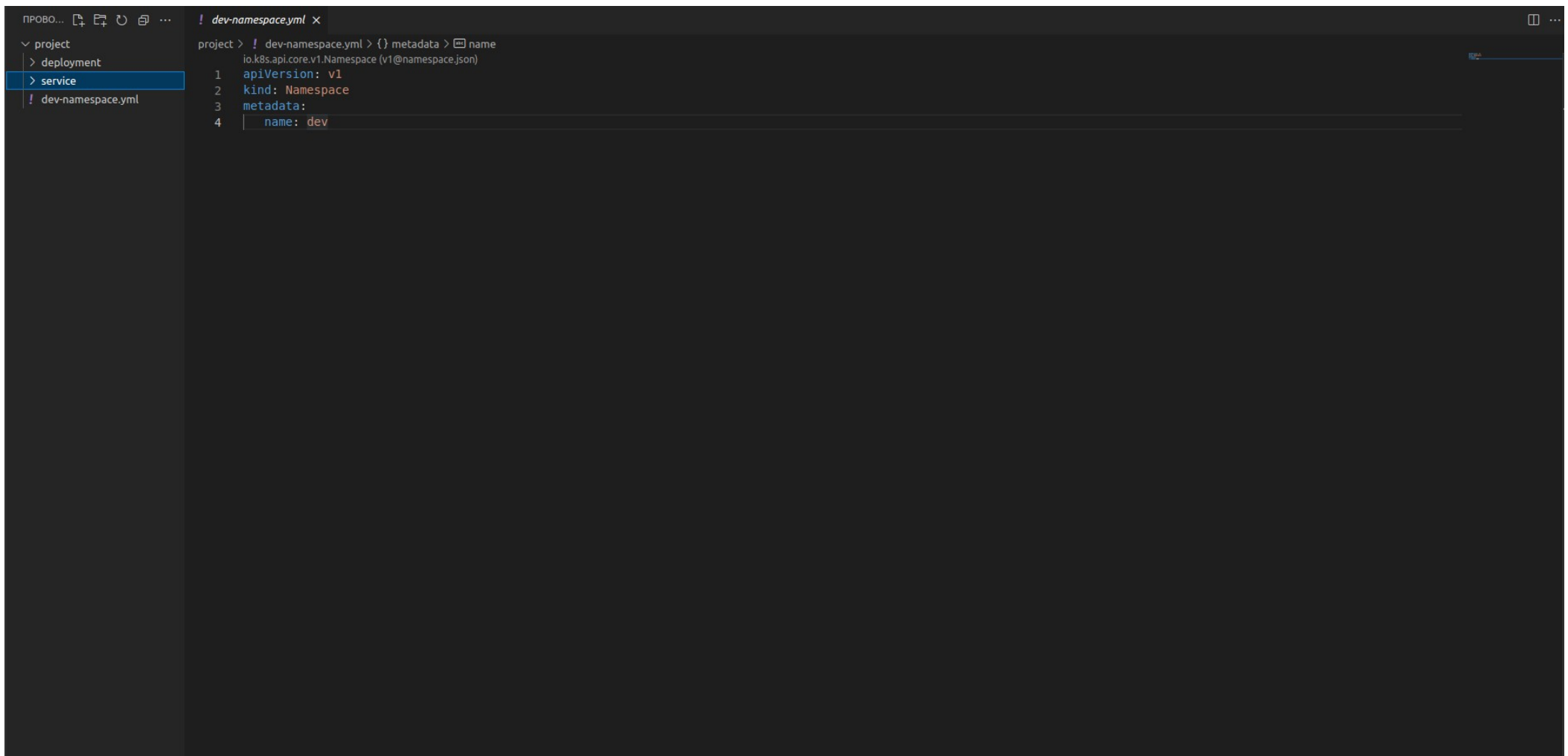
Что бы развернуть приложение необходимо создать для каждого сервиса deployment манифесты.

Указать в них:

- Label для дальнейшего определения каждого сервиса в кластере.
- Количество реплик приложения.
- Название image контейнера размещенного в docker hub
- В некоторых описаниях контейнера указать в env переменные.

Для связи компонентов между собой и с внешним миром необходимо создать, для каждого сервиса, service манифесты. По итогу будут созданы DNS записи по которым можно будет обращаться с любого Pod -а приложения текущего namespace.

Создал Namespace для разграничения стендов разработки  
Namespaces - это, по сути, виртуальный кластер Kubernetes внутри самого Kubernetes. Внутри каждого такого кластера находятся свои объекты(POD-ы, Service-ы, Deployment-ы и т. д.), кроме объектов, общих на все namespace-ы (nodes, ClusterRoles, PersistentVolumes)



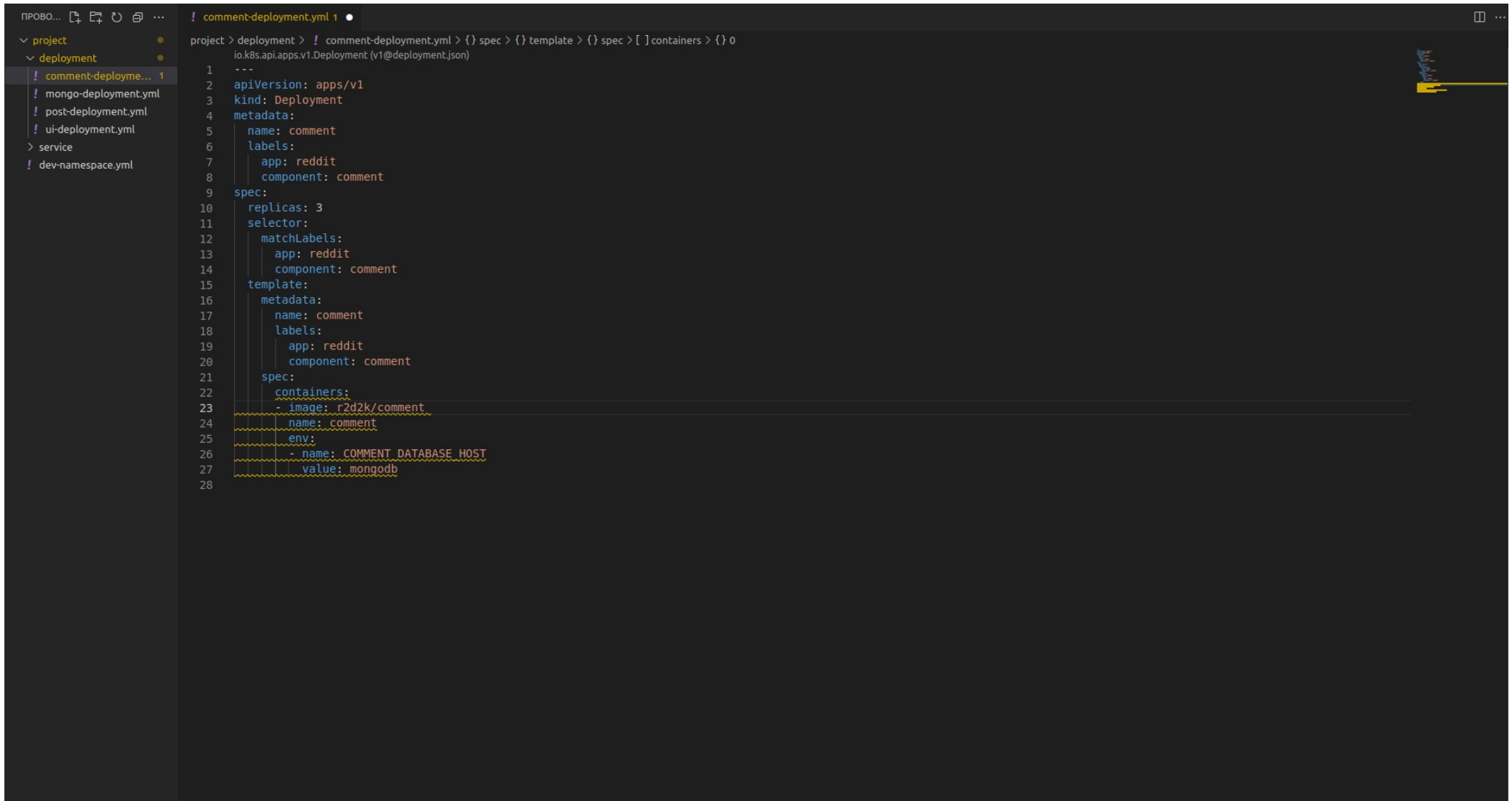
```
project > ! dev-namespace.yml > {} metadata > name
io.k8s.api.core.v1.Namespace (v1@namespace.json)
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: dev
```

The screenshot shows a terminal window with a dark background. On the left, there is a sidebar with a file explorer showing a directory structure: 'project' (expanded), 'deployment', 'service', and 'dev-namespace.yml'. The main area of the terminal displays the command 'project > ! dev-namespace.yml > {} metadata > name' and its output, which is a YAML configuration for a Kubernetes Namespace named 'dev'. The output is formatted with line numbers 1 through 4, showing 'apiVersion: v1', 'kind: Namespace', 'metadata:', and 'name: dev'.

# Deployment для сервиса comment

Количество реплик 3

В env указан COMMENT\_DATABASE\_HOST mongodb - это переменная окружения для обращения к базе



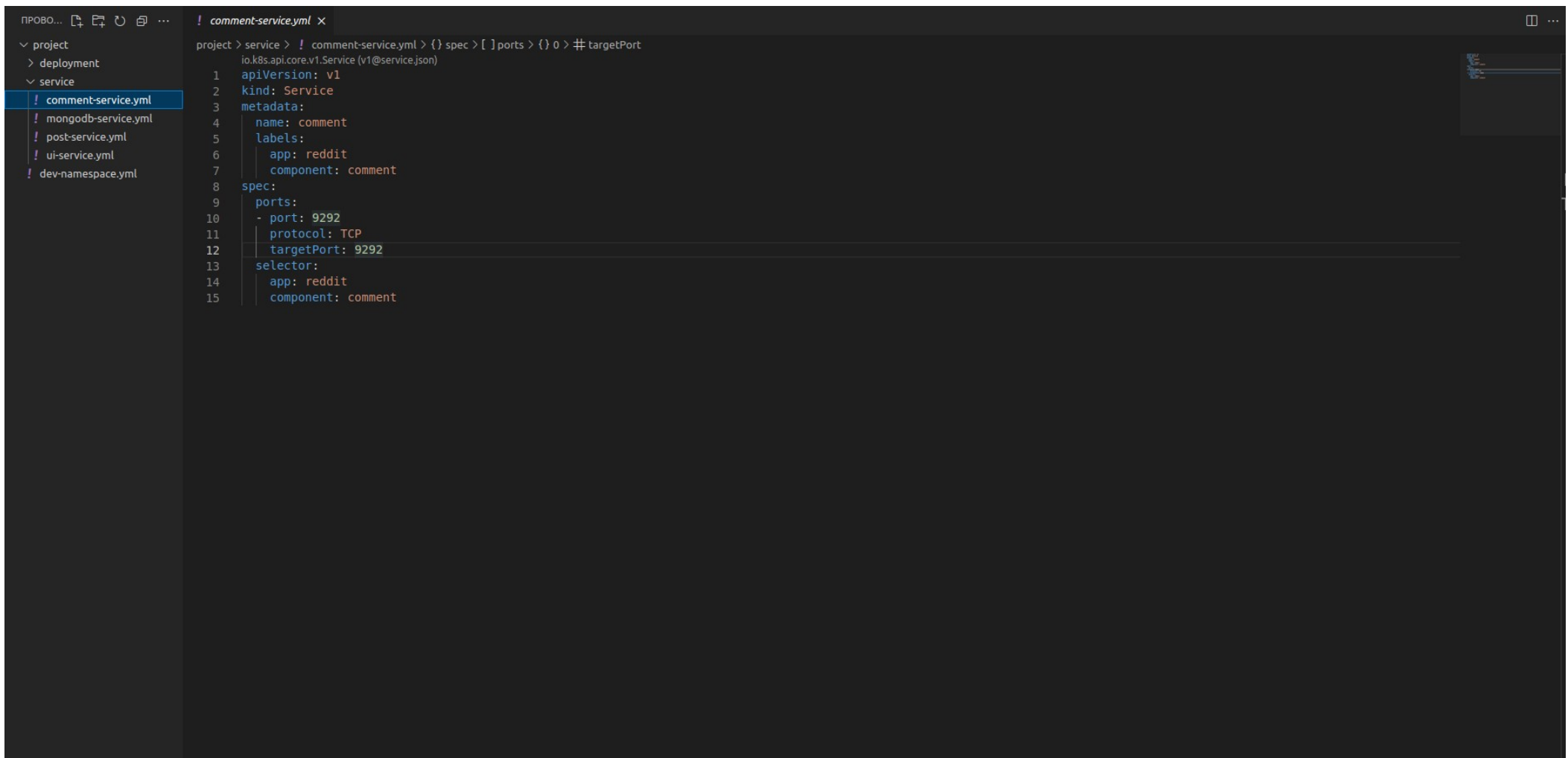
```
project > deployment > ! comment-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0
io.k8s.apl.apps.v1.Deployment (v1@deployment.json)

1 ---
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: comment
6   labels:
7     app: reddit
8     component: comment
9 spec:
10   replicas: 3
11   selector:
12     matchLabels:
13       app: reddit
14       component: comment
15   template:
16     metadata:
17       name: comment
18       labels:
19         app: reddit
20         component: comment
21     spec:
22       containers:
23       - image: r2d2k/comment
24         name: comment
25         env:
26         - name: COMMENT_DATABASE_HOST
27           value: mongodb
28
```

# Service для сервиса comment

В DNS появится запись для comment

При обращении на comment:9292 изнутри любого из POD-ов текущего namespace нас переправит на 9292 порт одного из POD-ов приложения comment, выбранных по label-ам.



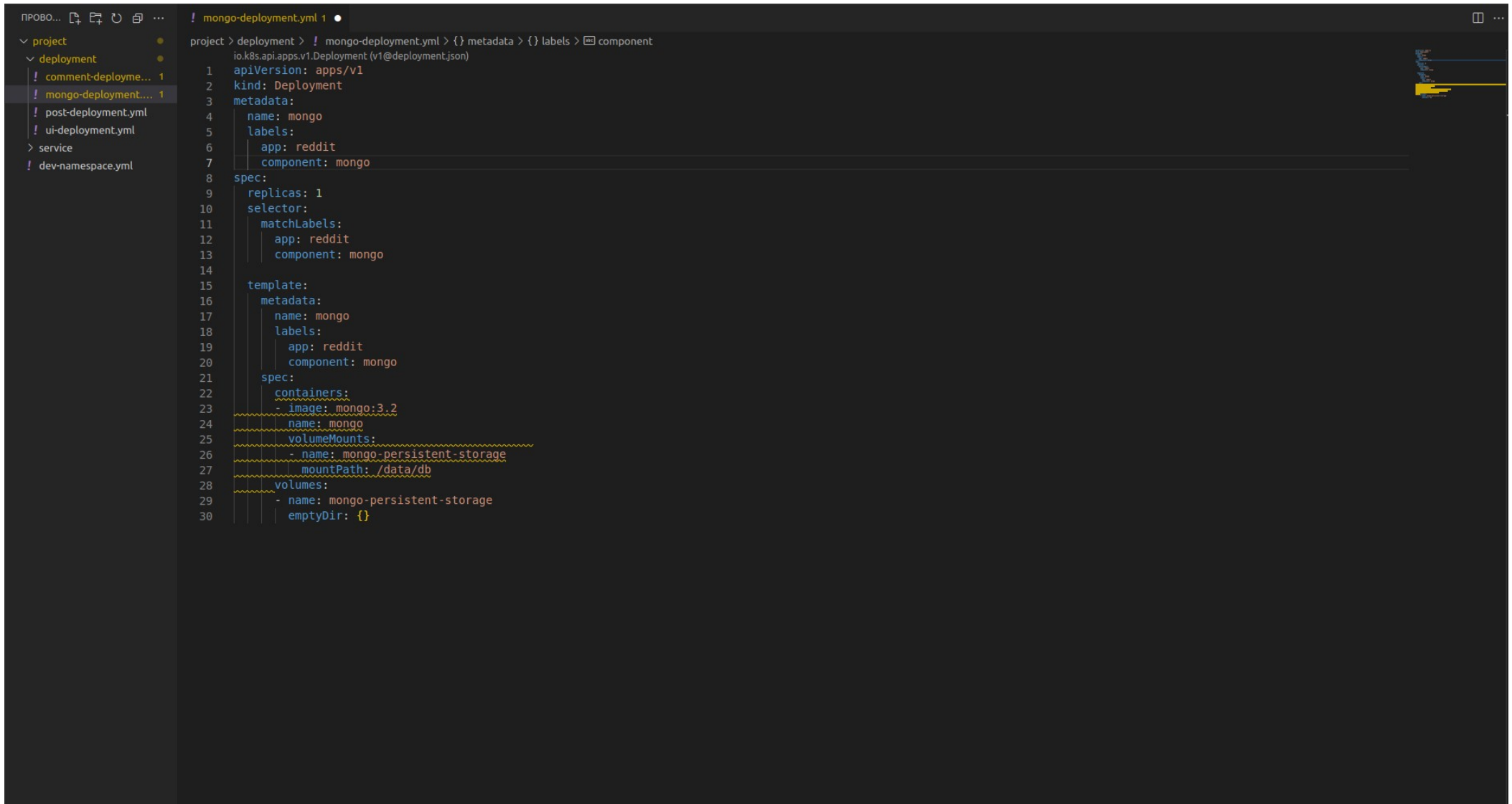
```
project > service > ! comment-service.yml > {} spec > [ ] ports > {} 0 > # targetPort
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: comment
5    labels:
6      app: reddit
7      component: comment
8  spec:
9    ports:
10     - port: 9292
11       protocol: TCP
12       targetPort: 9292
13    selector:
14      app: reddit
15      component: comment
```

# Deployment для сервиса mongodb

Количество реплик 1

VolumeMounts - Точка монтирования тома в контейнере (не в POD)

Volumes - Ассоциированные с POD-ом Volume-ы



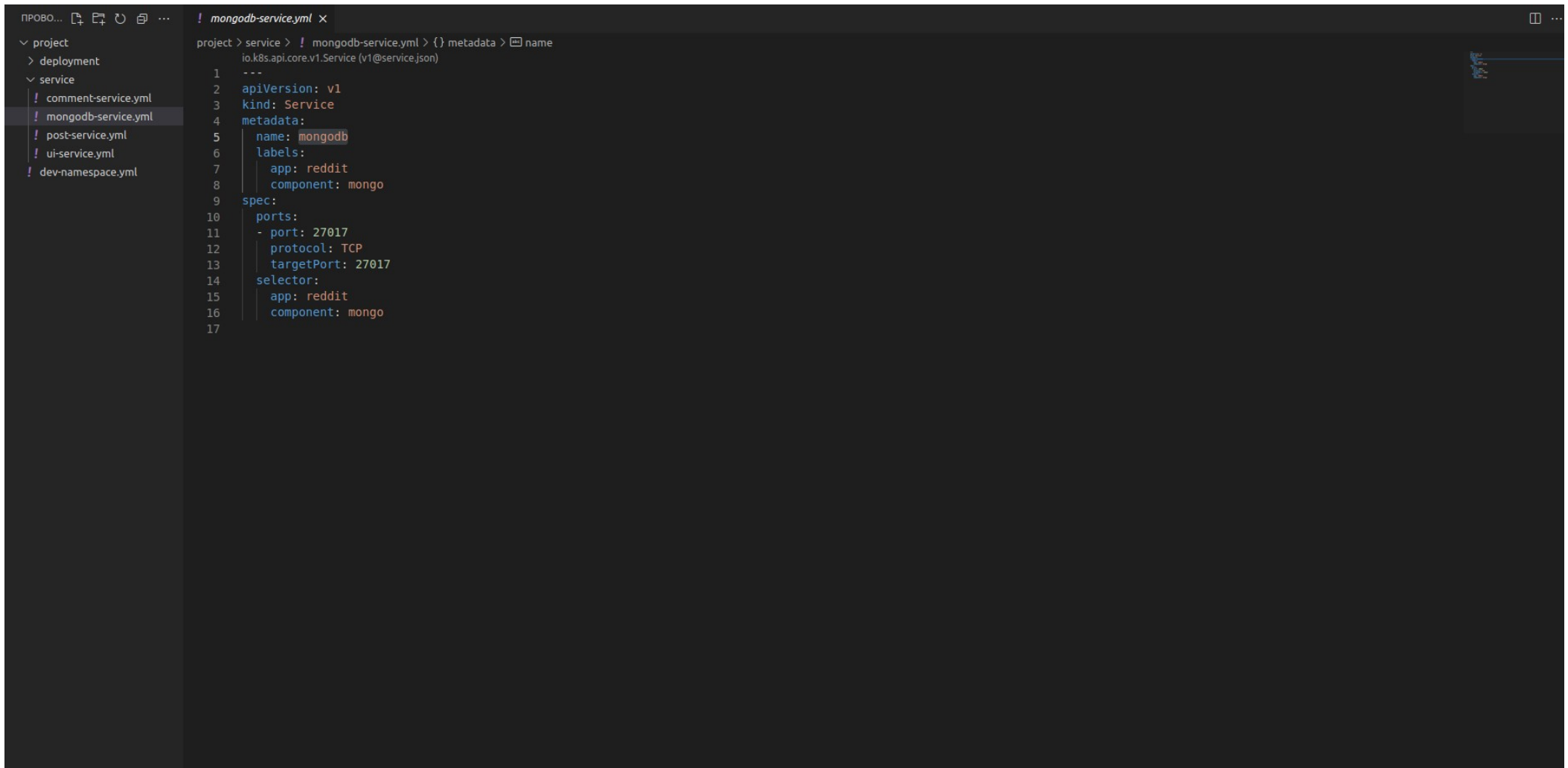
```
project > deployment > ! mongo-deployment.yml > {} metadata > {} labels > component
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mongo
5    labels:
6      app: reddit
7      component: mongo
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: reddit
13       component: mongo
14
15   template:
16     metadata:
17       name: mongo
18       labels:
19         app: reddit
20         component: mongo
21     spec:
22       containers:
23         - image: mongo:3.2
24           name: mongo
25           volumeMounts:
26             - name: mongo-persistent-storage
27               mountPath: /data/db
28       volumes:
29         - name: mongo-persistent-storage
30           emptyDir: {}
```



Service для сервиса mongodb

В DNS появится запись для mongodb

При обращении на mongodb:27017 изнутри любого из POD-ов текущего namespace нас переправит на 27017 порт приложения mongo, по выбранным label-ам.

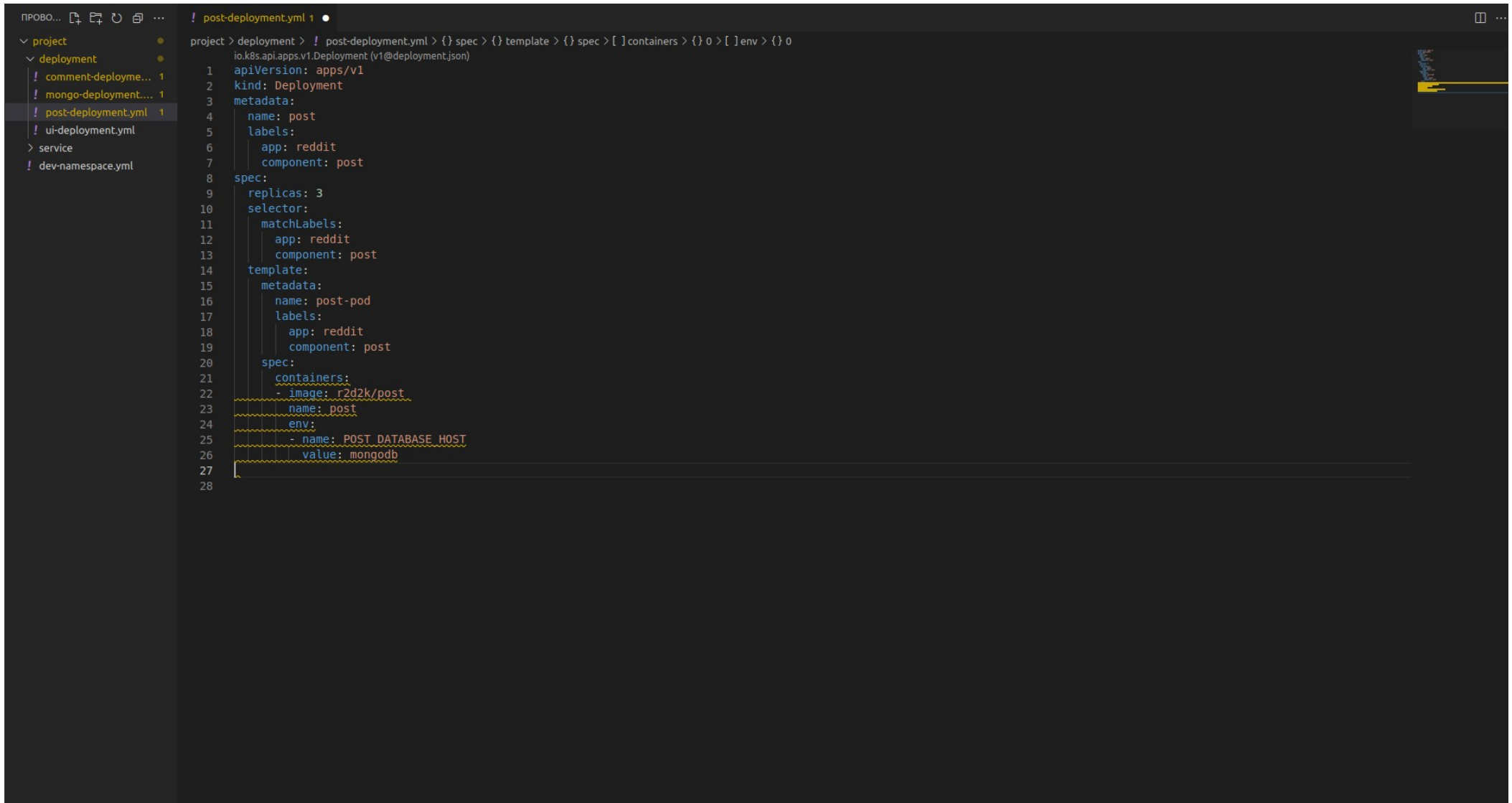


```
1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: mongodb
6   labels:
7     app: reddit
8     component: mongo
9 spec:
10   ports:
11     - port: 27017
12       protocol: TCP
13       targetPort: 27017
14   selector:
15     app: reddit
16     component: mongo
17
```

# Deployment для сервиса post

Количество реплик 3

В env указан POST\_DATABASE\_HOST mongodb — это переменная окружения для обращения к базе

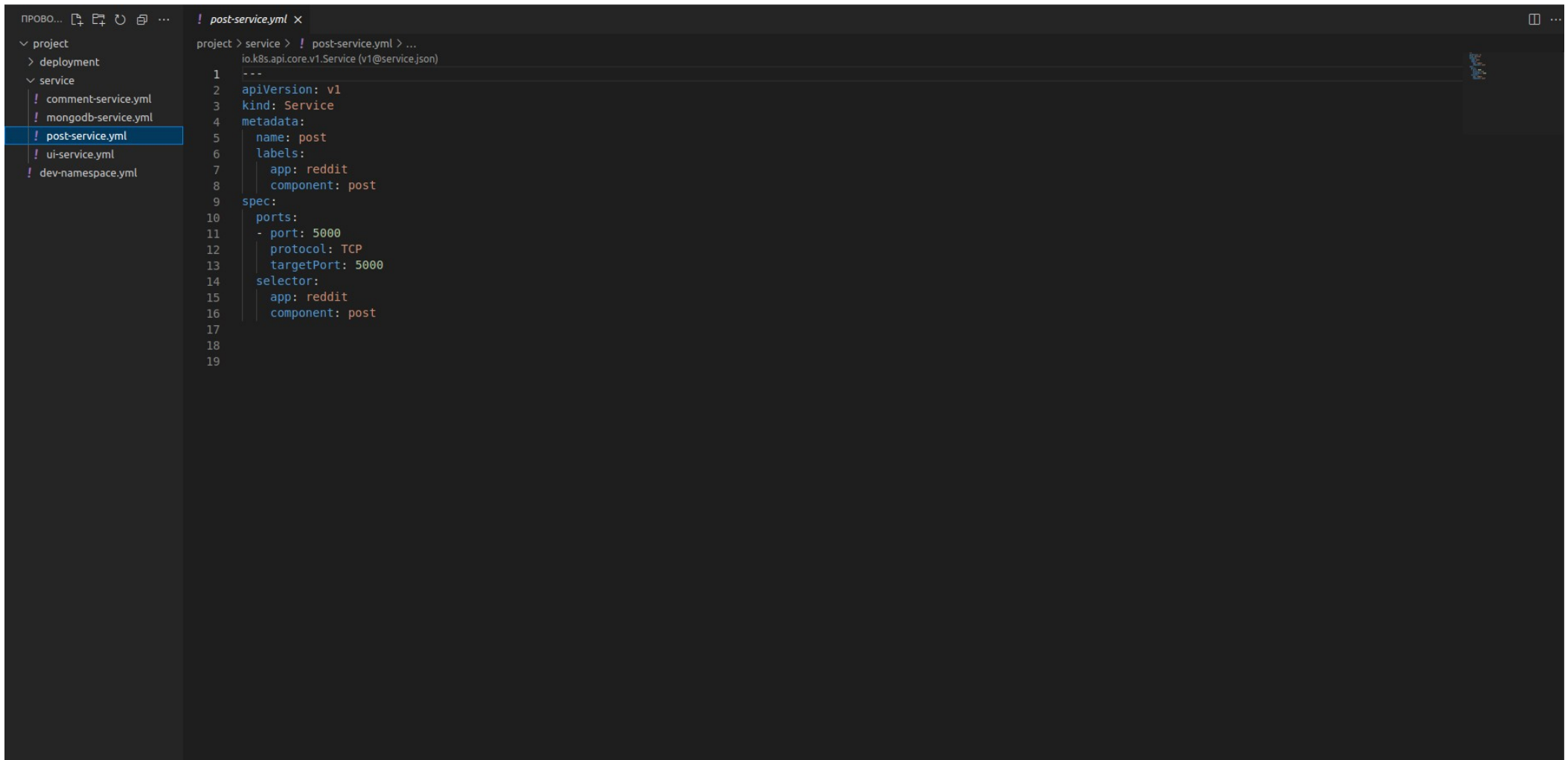


```
project > deployment > ! post-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] env > {} 0
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: post
5    labels:
6      app: reddit
7      component: post
8  spec:
9    replicas: 3
10   selector:
11     matchLabels:
12       app: reddit
13       component: post
14   template:
15     metadata:
16       name: post-pod
17     labels:
18       app: reddit
19       component: post
20     spec:
21       containers:
22         - image: r2d2k/post
23           name: post
24           env:
25             - name: POST_DATABASE_HOST
26               value: mongodb
27
28
```

# Service для сервиса post

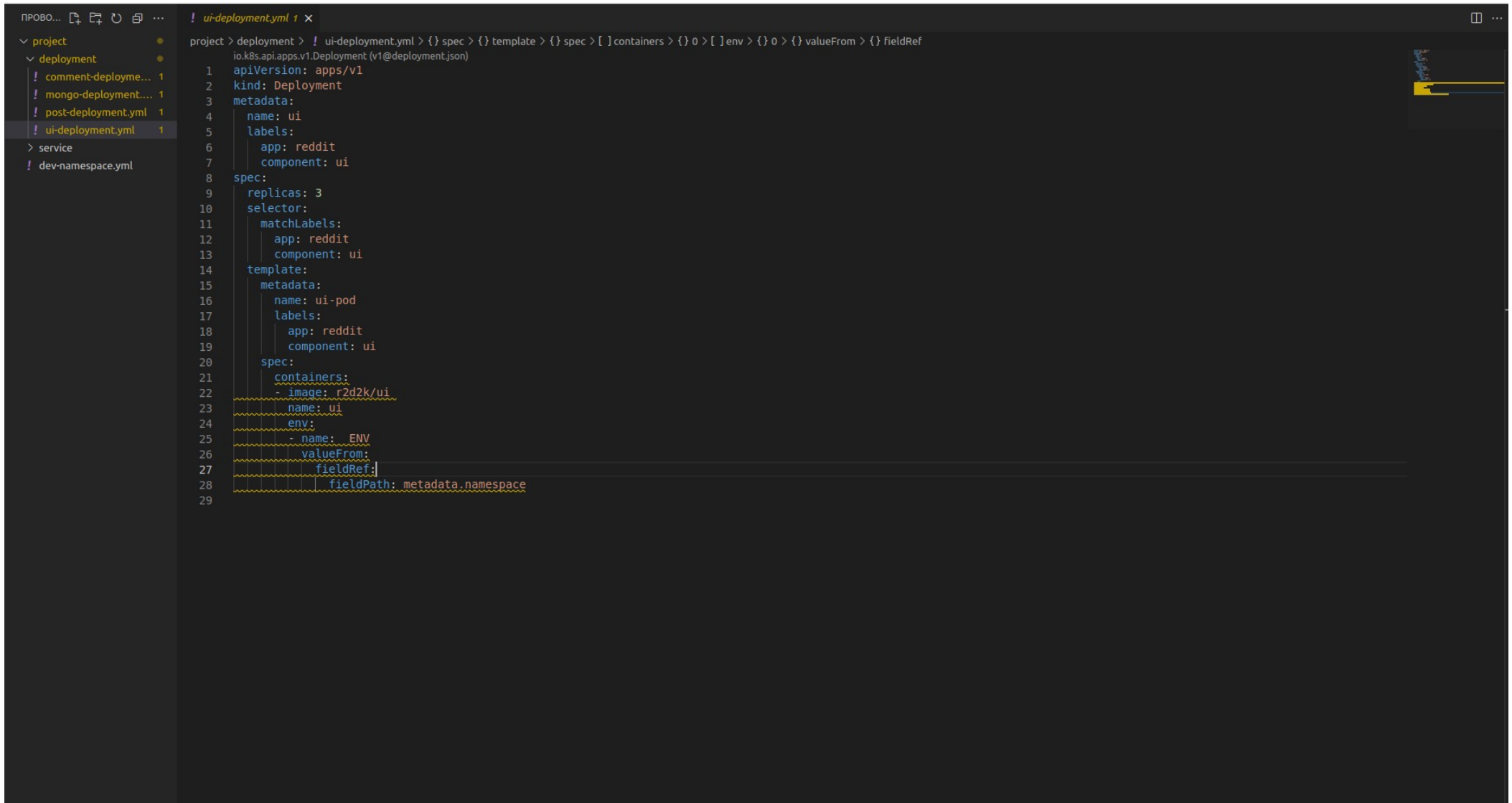
В DNS появится запись для post

При обращении на post:5000 изнутри любого из POD-ов текущего namespace нас переправит на 5000 порт одного из POD-ов приложения post, выбранных по label-ам.



```
project > service > ! post-service.yml > ...
io.k8s.api.core.v1.Service (v1@service.json)
1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: post
6   labels:
7     app: reddit
8     component: post
9 spec:
10  ports:
11    - port: 5000
12      protocol: TCP
13      targetPort: 5000
14  selector:
15    app: reddit
16    component: post
17
18
19
```

Deployment для сервиса ui  
Количество реплик 3  
В env указан том БД



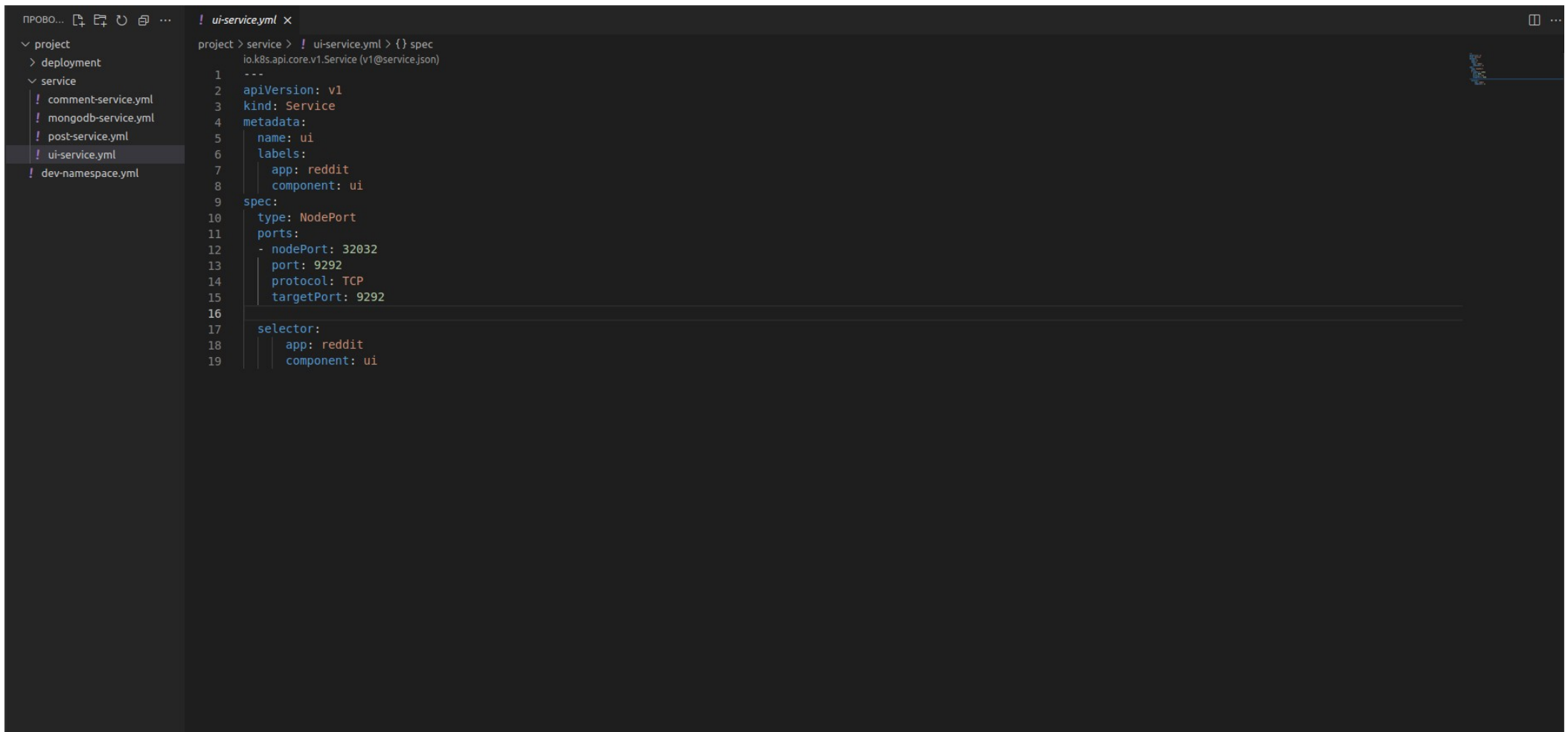
```
project > deployment > ! ui-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] env > {} 0 > {} valueFrom > {} fieldRef
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: ui
5    labels:
6      app: reddit
7      component: ui
8  spec:
9    replicas: 3
10   selector:
11     matchLabels:
12       app: reddit
13       component: ui
14   template:
15     metadata:
16       name: ui-pod
17       labels:
18         app: reddit
19         component: ui
20     spec:
21       containers:
22         - image: r2d2k/ui
23         name: ui
24         env:
25           - name: ENV
26             valueFrom:
27               fieldRef:
28                 fieldPath: metadata.namespace
29
```

Service для сервиса ui

В DNS появится запись для ui

При обращении на ui:9292 изнутри любого из POD-ов текущего namespace нас переправит на 9292 порт одного из POD-ов приложения ui, выбранных по label.

Так же мы открыли порт 32032 доступный вне kubernetes.



```
project > service > ! ui-service.yml > {} spec
io.k8s.api.core.v1.Service (v1@service.json)
1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: ui
6   labels:
7     app: reddit
8     component: ui
9 spec:
10  type: NodePort
11  ports:
12  - nodePort: 32032
13    port: 9292
14    protocol: TCP
15    targetPort: 9292
16
17 selector:
18   app: reddit
19   component: ui
```

# Kubernetes развернул в yandex cloud с двумя узлами. Создал namespace dev.

```

  ▢  📄  📄  🔄  🗑  ...
  ▾ project
    > deployment
    > service
    ! dev-namespace.yml

```

```

alex@alex:~/myDoc/учеба/project$ yc managed-kubernetes cluster get-credentials test --external
Context 'yc-test' was added as default to kubeconfig '/home/alex/.kube/config'.
Check connection to cluster using 'kubectl cluster-info --kubeconfig /home/alex/.kube/config'.

Note, that authentication depends on 'yc' and its config profile 'deploy'.
To access clusters using the Kubernetes API, please use Kubernetes Service Account.
alex@alex:~/myDoc/учеба/project$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
cl1jj74vhuf6h3etsjv1-abyz          Ready    <none>    4m43s v1.22.6
cl1jj74vhuf6h3etsjv1-utaq          Ready    <none>    4m36s v1.22.6
alex@alex:~/myDoc/учеба/project$ kubectl get pods
No resources found in default namespace.
alex@alex:~/myDoc/учеба/project$ kubectl get deployment
No resources found in default namespace.
alex@alex:~/myDoc/учеба/project$ kubectl get service
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes ClusterIP  10.96.128.1    <none>         443/TCP       81m
alex@alex:~/myDoc/учеба/project$ kubectl apply -f dev-namespace.yml
namespace/dev created
alex@alex:~/myDoc/учеба/project$ kubectl get namespace
NAME              STATUS    AGE
default           Active    82m
dev               Active    42s
kube-node-lease   Active    82m
kube-public       Active    82m
kube-system       Active    82m
yandex-system     Active    82m
alex@alex:~/myDoc/учеба/project$

```

Запустил манифесты с deployment и service.  
Проверил что все развернулось.  
Все pod -ы в статусе Running все прошло успешно.  
Все service созданы ошибок нет.

```

п  [иконки] ...
ПРОБЛЕМЫ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ  КОММЕНТАРИИ

▼ project
  > deployment
  > service
  ! dev-namespace.yml

alex@alex:~/myDoc/учеба/project$ kubectl apply -n dev -f deployment/
deployment.apps/comment created
deployment.apps/mongo created
deployment.apps/post created
deployment.apps/ui created
alex@alex:~/myDoc/учеба/project$ kubectl apply -n dev -f service/
service/comment created
service/mongodb created
service/post created
service/ui created
alex@alex:~/myDoc/учеба/project$ kubectl get pods -n dev
NAME                                READY   STATUS    RESTARTS   AGE
comment-7755456d6c-7dksk            1/1     Running   0           21s
comment-7755456d6c-qjw7g            1/1     Running   0           21s
comment-7755456d6c-w4wmv            1/1     Running   0           21s
mongo-77d8cfd6d6-hprxx              1/1     Running   0           20s
post-b659fc678-bt5ld                1/1     Running   0           20s
post-b659fc678-fmxkh                1/1     Running   0           20s
post-b659fc678-nnkd5                1/1     Running   0           20s
ui-565b9d6499-cb6hj                 1/1     Running   0           20s
ui-565b9d6499-t8hbx                 1/1     Running   0           19s
ui-565b9d6499-xbtqm                 1/1     Running   0           19s
alex@alex:~/myDoc/учеба/project$ kubectl get deployment -n dev
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
comment   3/3     3             3           30s
mongo     1/1     1             1           29s
post      3/3     3             3           29s
ui        3/3     3             3           29s
alex@alex:~/myDoc/учеба/project$ kubectl get service -n dev
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
comment   ClusterIP   10.96.252.215 <none>        9292/TCP         28s
mongodb   ClusterIP   10.96.176.196 <none>        27017/TCP        28s
post      ClusterIP   10.96.189.132 <none>        5000/TCP         27s
ui        NodePort    10.96.180.113 <none>        9292:32032/TCP   27s
alex@alex:~/myDoc/учеба/project$
```

Проверил один из созданных service.  
Все другие можно проверить аналогично.  
Для примера взяли service для comment.  
Получил внутренний ip адрес из описания .  
Далее сделал запрос из пода post по dns имени comment и  
получили в ответе тот же ip адрес.

```

  ▢  📄  🔍  🔄  🗑  ...
  ▾ project
    > deployment
    > service
    ! dev-namespace.yml
  ● alex@alex:~/myDoc/учеба/project$ kubectl get service -n dev
  NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
  comment    ClusterIP  10.96.252.215    <none>           9292/TCP      3m12s
  mongodb    ClusterIP  10.96.176.196    <none>           27017/TCP     3m12s
  post       ClusterIP  10.96.189.132    <none>           5000/TCP      3m11s
  ui         NodePort   10.96.180.113    <none>           9292:32032/TCP 3m11s
  ● alex@alex:~/myDoc/учеба/project$ kubectl describe service comment -n dev
  Name:      comment
  Namespace: dev
  Labels:    app=reddit
             component=comment
  Annotations: <none>
  Selector:   app=reddit,component=comment
  Type:       ClusterIP
  IP Family Policy: SingleStack
  IP Families: IPv4
  IP:         10.96.252.215
  IPs:        10.96.252.215
  Port:       <unset> 9292/TCP
  TargetPort: 9292/TCP
  Endpoints:  10.112.128.18:9292,10.112.128.19:9292,10.112.129.13:9292
  Session Affinity: None
  Events:     <none>
  ● alex@alex:~/myDoc/учеба/project$ kubectl get pods -n dev
  NAME                                READY   STATUS    RESTARTS   AGE
  comment-7755456d6c-7dksk             1/1     Running   0           3m28s
  comment-7755456d6c-qjw7g             1/1     Running   0           3m28s
  comment-7755456d6c-w4wmv             1/1     Running   0           3m28s
  mongo-77d8cf6d6d-hprxx               1/1     Running   0           3m27s
  post-b659fc678-bt5ld                 1/1     Running   0           3m27s
  post-b659fc678-fmxkh                 1/1     Running   0           3m27s
  post-b659fc678-nnkd5                 1/1     Running   0           3m27s
  ui-565b9d6499-cb6hj                 1/1     Running   0           3m27s
  ui-565b9d6499-t8hbx                 1/1     Running   0           3m26s
  ui-565b9d6499-xbtqm                 1/1     Running   0           3m26s
  ● alex@alex:~/myDoc/учеба/project$ kubectl exec post-b659fc678-bt5ld -n dev -- nslookup comment
  Name:      comment
  Address 1: 10.96.252.215 comment.dev.svc.cluster.local
  nslookup: can't resolve '(null)': Name does not resolve
  ○ alex@alex:~/myDoc/учеба/project$
```

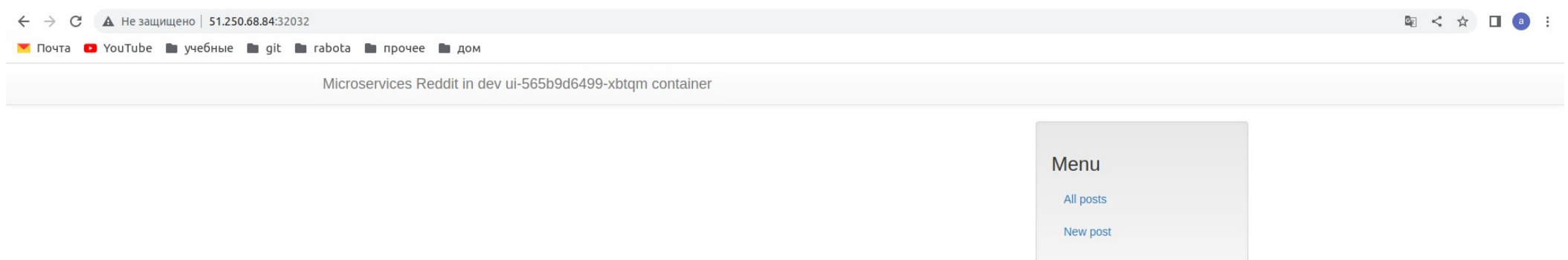


В service для ui мы указал порт доступный из вне 32032  
Убедился что это так посмотрев это уже в самом кластере.  
Выяснил ip адрес нод.

```

п  ↗  ↘  ↻  ⌂  ...
▼ project
  > deployment
  > service
  ! dev-namespace.yml
● alex@alex:~/myDoc/учеба/project$ kubectl get service -n dev
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
comment   ClusterIP    10.96.252.215 <none>         9292/TCP          5m18s
mongodb    ClusterIP    10.96.176.196 <none>         27017/TCP         5m18s
post       ClusterIP    10.96.189.132 <none>         5000/TCP          5m17s
ui         NodePort     10.96.180.113 <none>         9292:32032/TCP    5m17s
● alex@alex:~/myDoc/учеба/project$ kubectl get nodes -o wide
NAME                                STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
cl1jj74vhuf6h3etsjv1-abyz          Ready    <none>    45m   v1.22.6   10.128.0.15   51.250.68.84   Ubuntu 20.04.4 LTS   5.4.0-124-generic containerd://1.6.7
cl1jj74vhuf6h3etsjv1-utaq          Ready    <none>    45m   v1.22.6   10.128.0.24   130.193.36.198 Ubuntu 20.04.4 LTS   5.4.0-124-generic containerd://1.6.7
○ alex@alex:~/myDoc/учеба/project$
```

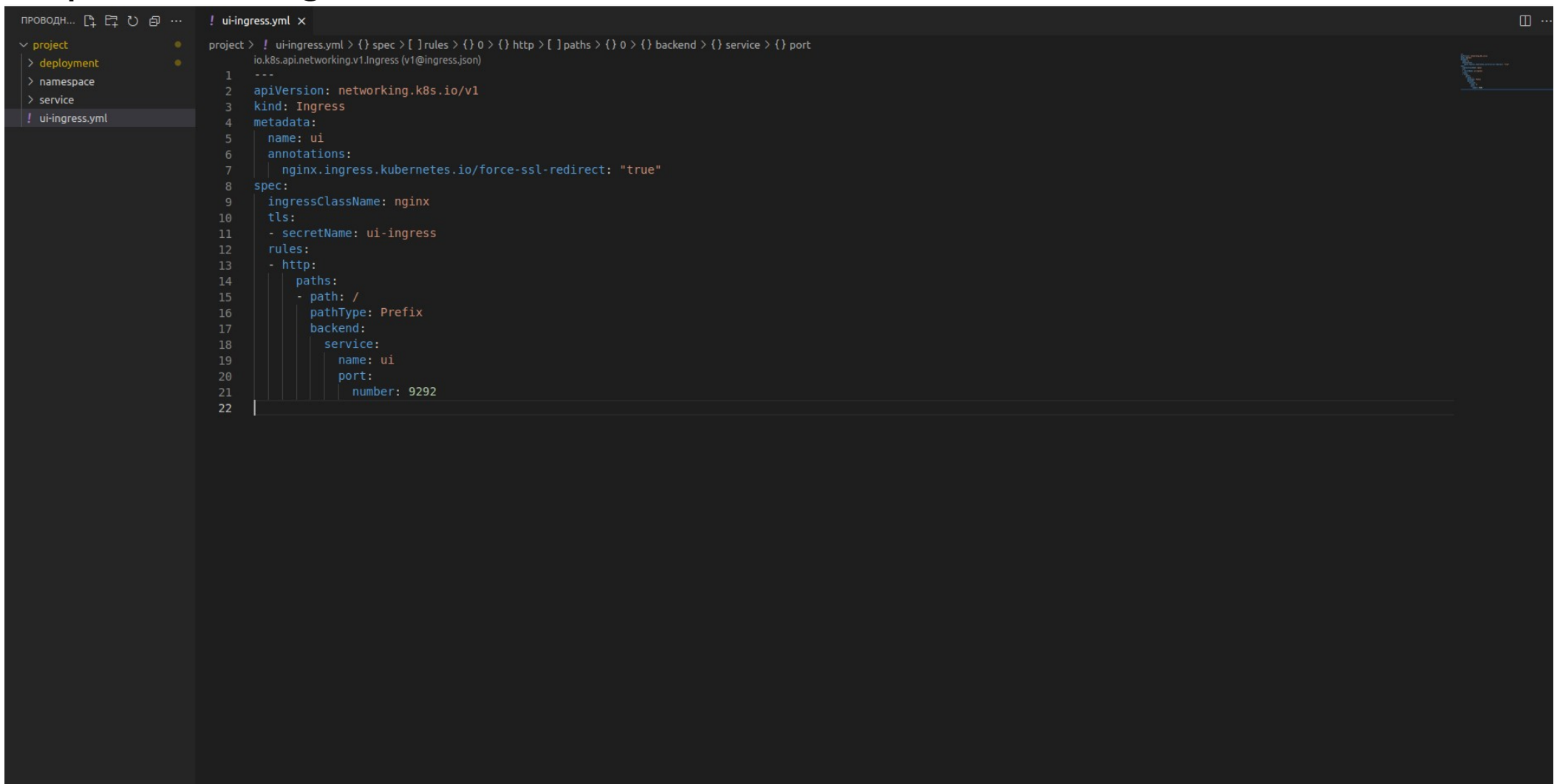
Перешел в браузере по ip адресс ноды порт 32032.  
В браузере видим в названии dev это namespace, а ui-565b9d6499-xbtqm это название одного из Pod -а ui.



Для организации единой точки входа в приложения снаружи установил ingress-nginx так как kubernetes из коробки не предоставляет ingress.

Создал манифест ingress.

С работой Ingress Controller как классический веб.

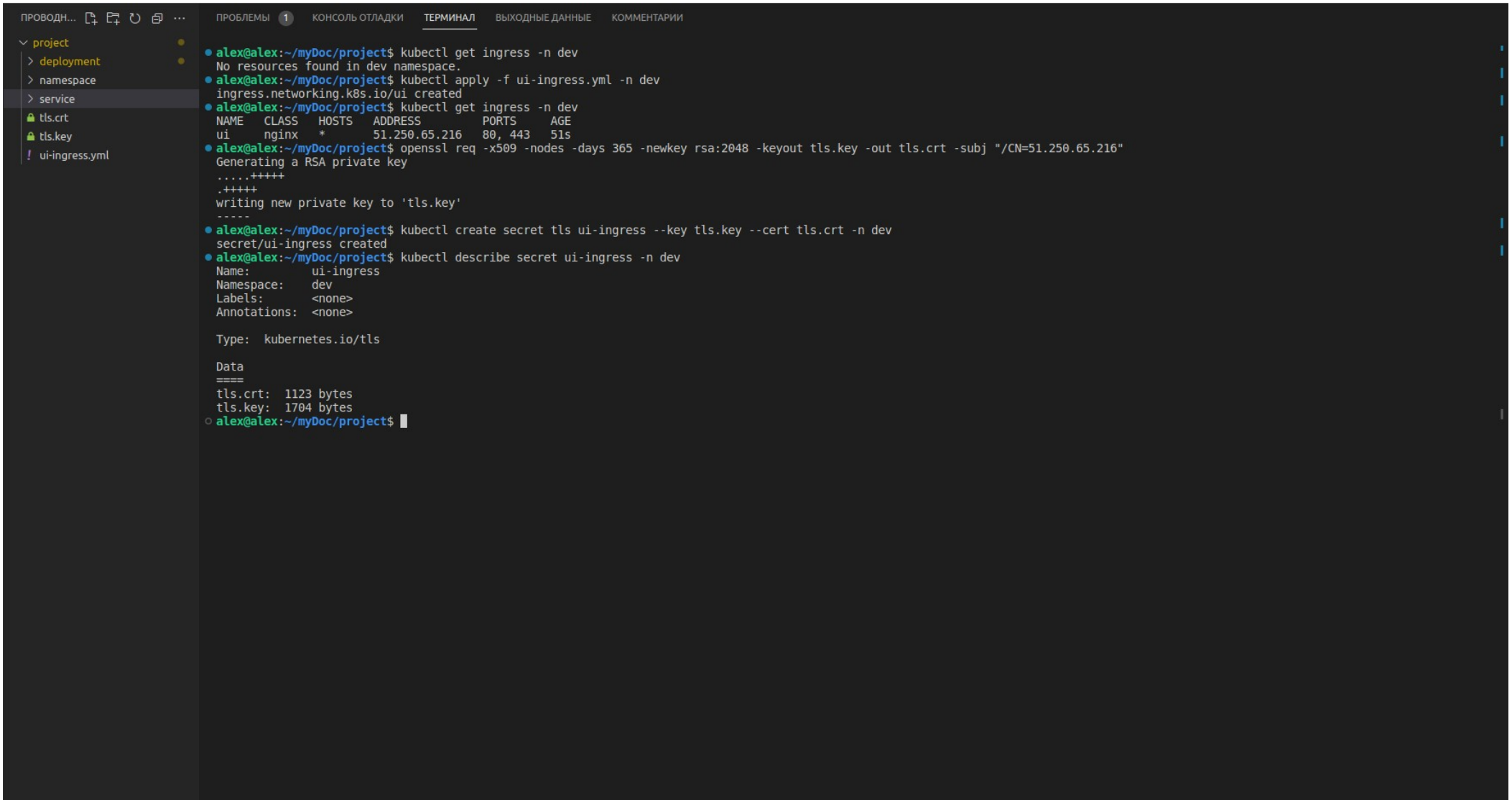
A screenshot of a code editor with a dark theme. On the left, a sidebar shows a file tree with 'project' expanded, containing 'deployment', 'namespace', 'service', and 'ui-ingress.yml'. The main editor area displays the content of 'ui-ingress.yml'. The file starts with a comment line 'project > ! ui-ingress.yml > {} spec > [ ] rules > {} 0 > {} http > [ ] paths > {} 0 > {} backend > {} service > {} port' followed by 'io.k8s.api.networking.v1.Ingress (v1@ingress.json)'. The manifest defines an Ingress resource named 'ui' with an annotation for force-ssl-redirect. It specifies the ingressClassName as 'nginx', uses a secret named 'ui-ingress' for TLS, and has a single rule for the root path '/' that routes traffic to the 'ui' service on port 9292.

```
project > ! ui-ingress.yml > {} spec > [ ] rules > {} 0 > {} http > [ ] paths > {} 0 > {} backend > {} service > {} port
io.k8s.api.networking.v1.Ingress (v1@ingress.json)
1 ---
2 apiVersion: networking.k8s.io/v1
3 kind: Ingress
4 metadata:
5   name: ui
6   annotations:
7     nginx.ingress.kubernetes.io/force-ssl-redirect: "true"
8 spec:
9   ingressClassName: nginx
10  tls:
11    - secretName: ui-ingress
12  rules:
13    - http:
14      paths:
15        - path: /
16          pathType: Prefix
17          backend:
18            service:
19              name: ui
20              port:
21                number: 9292
22
```

Создал ingress.

Подготовил тестовый сертификат используя IP как CN для проверки защиты нашего сервиса с помощью TLS.

Загрузил сертификат в кластер kubernetes.



```
ПРОВЕДН...  [?] [?] [?] [?] ...  ПРОБЛЕМЫ 1  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ  КОММЕНТАРИИ

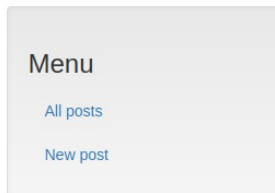
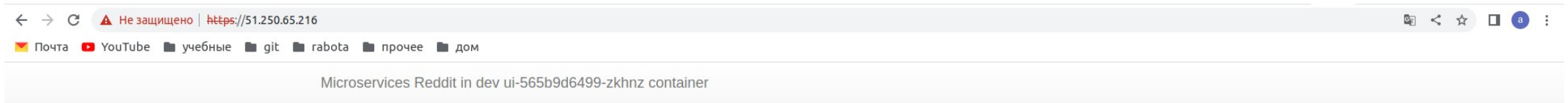
▼ project
  > deployment
  > namespace
  > service
  ▲ tls.crt
  ▲ tls.key
  ! ui-ingress.yml

alex@alex:~/myDoc/project$ kubectl get ingress -n dev
No resources found in dev namespace.
alex@alex:~/myDoc/project$ kubectl apply -f ui-ingress.yml -n dev
ingress.networking.k8s.io/ui created
alex@alex:~/myDoc/project$ kubectl get ingress -n dev
NAME      CLASS  HOSTS      ADDRESS      PORTS      AGE
ui        nginx  *          51.250.65.216  80, 443    51s
alex@alex:~/myDoc/project$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=51.250.65.216"
Generating a RSA private key
.....+++++
.+++++
writing new private key to 'tls.key'
-----
alex@alex:~/myDoc/project$ kubectl create secret tls ui-ingress --key tls.key --cert tls.crt -n dev
secret/ui-ingress created
alex@alex:~/myDoc/project$ kubectl describe secret ui-ingress -n dev
Name:         ui-ingress
Namespace:    dev
Labels:       <none>
Annotations:  <none>

Type: kubernetes.io/tls

Data
====
tls.crt: 1123 bytes
tls.key: 1704 bytes
alex@alex:~/myDoc/project$
```

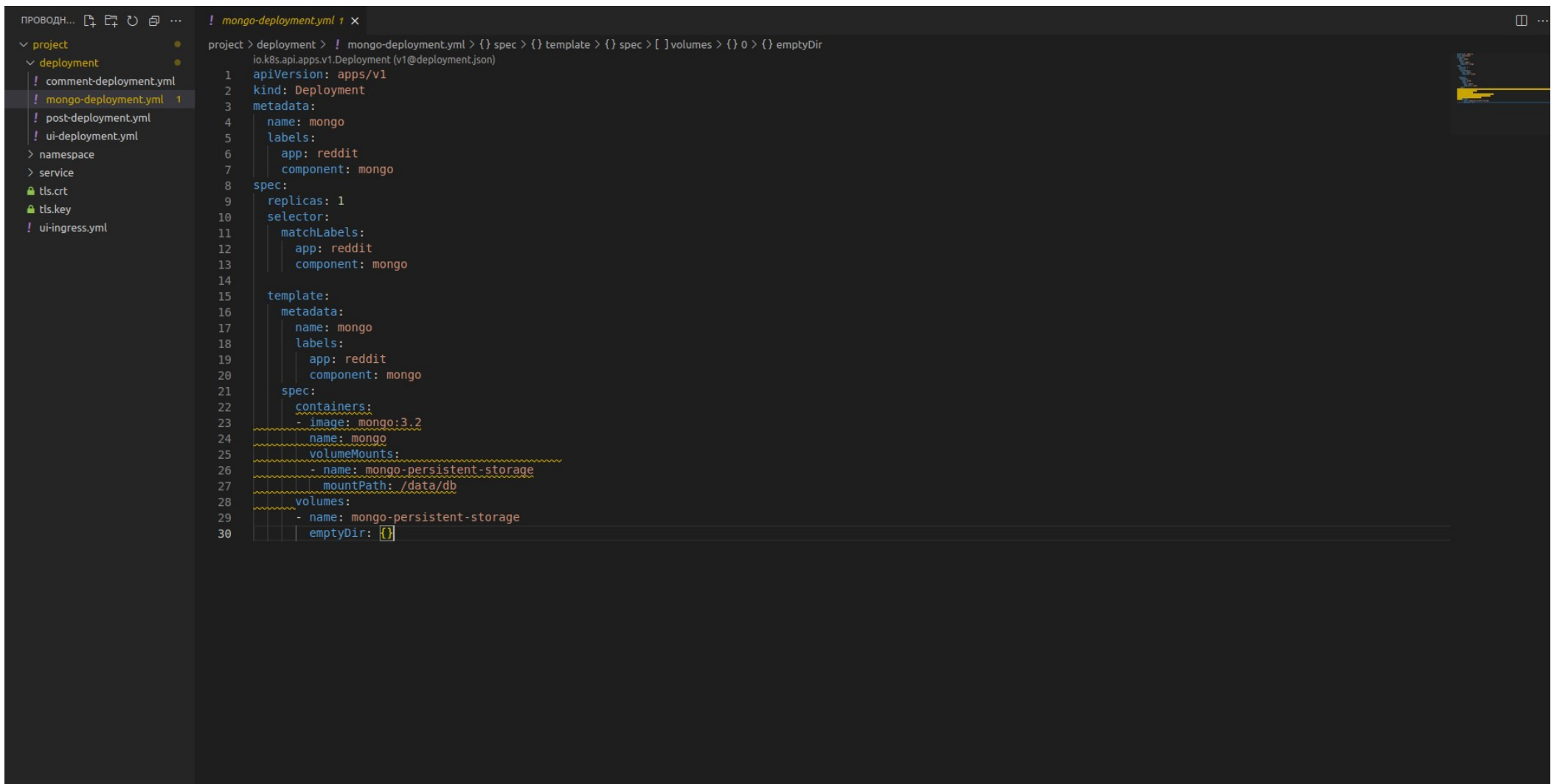
Перейдя в браузер по ip ingress видим приложение с редиректом на https.



Сейчас используется тип Volume emptyDir.

При создании пода с таким типом просто создается пустой docker volume.

При остановке POD'а содержимое emptyDir удалится навсегда.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'deployment' folder containing several YAML files, including 'mongo-deployment.yml'. The code editor displays the content of 'mongo-deployment.yml', which is a Kubernetes Deployment manifest for MongoDB. The manifest includes metadata, labels, and a spec section with replicas, selector, and template. The template section includes a container named 'mongo' with a volume mount named 'mongo-persistent-storage' at the path '/data/db'. The volume is defined as an 'emptyDir' type.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mongo
5    labels:
6      app: reddit
7      component: mongo
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: reddit
13       component: mongo
14
15   template:
16     metadata:
17       name: mongo
18       labels:
19         app: reddit
20         component: mongo
21     spec:
22       containers:
23         - image: mongo:3.2
24           name: mongo
25           volumeMounts:
26             - name: mongo-persistent-storage
27               mountPath: /data/db
28       volumes:
29         - name: mongo-persistent-storage
30           emptyDir: {}
```

Для сохранения своих данных необходимо использовать механизм PersistentVolume.

Сначала создал диск в ya.cloud на 4 Gi.

```
ПРОВЕДН...  [иконки] ...
ПРОБЛЕМЫ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ  КОММЕНТАРИИ

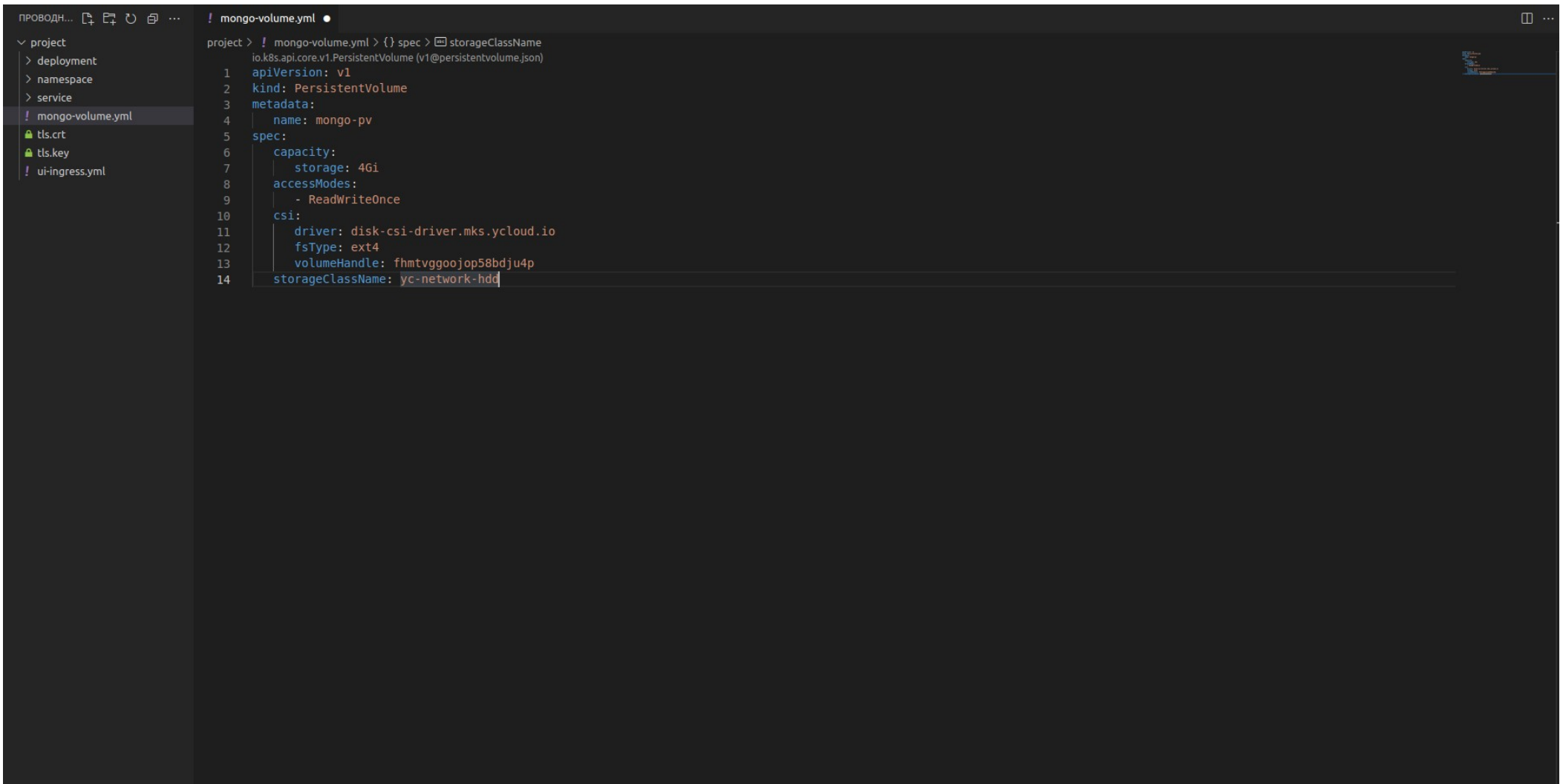
▼ project
  > deployment
  > namespace
  > service
  🟢 tls.crt
  🟢 tls.key
  ! ui-ingress.yml

alex@alex:~/myDoc/project$ yc compute disk create \
> --name k8s \
> --size 4 \
> --description "disk for k8s"
done (11s)
id: fhmtvggoojop58bdju4p
folder_id: blgv4l14uqqaafgtpcpr
created_at: "2023-01-17T14:36:45Z"
name: k8s
description: disk for k8s
type_id: network-hdd
zone_id: ru-central1-a
size: "4294967296"
block_size: "4096"
status: READY
disk_placement_policy: {}

alex@alex:~/myDoc/project$ yc compute disk list
+-----+-----+-----+-----+-----+-----+-----+
| ID | NAME | SIZE | ZONE | STATUS | INSTANCE IDS | DESCRIPTION |
+-----+-----+-----+-----+-----+-----+-----+
| fhm96hq4l0iciim7vkrr | | 103079215104 | ru-central1-a | READY | fhmgstcbpflpjdk4v39c | |
| fhmifg0p87fpfbula1k | | 103079215104 | ru-central1-a | READY | fhmqovd437e5dotp7k9s | |
| fhmtvggoojop58bdju4p | k8s | 4294967296 | ru-central1-a | READY | | disk for k8s |
+-----+-----+-----+-----+-----+-----+-----+

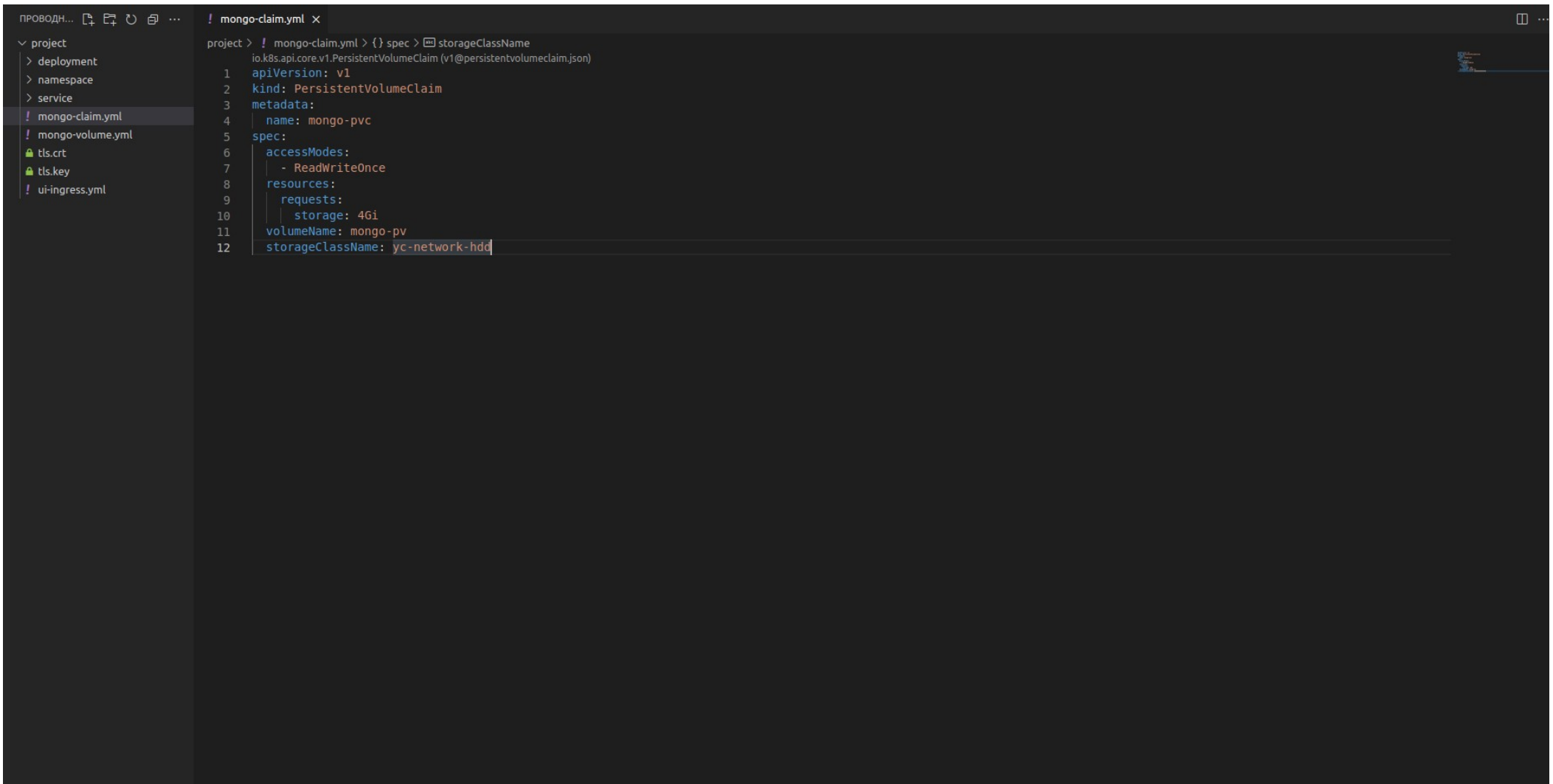
alex@alex:~/myDoc/project$
```

В манифесте PersistentVolume прописал создание ресурса дискового хранилища, распространенного на весь кластер



```
project > ! mongo-volume.yml > {} spec > storageClassName
io.k8s.api.core.v1.PersistentVolume (v1@persistentvolume.json)
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: mongo-pv
5  spec:
6    capacity:
7      storage: 4Gi
8    accessModes:
9      - ReadWriteOnce
10   csi:
11     driver: disk-csi-driver.mks.ycloud.io
12     fsType: ext4
13     volumeHandle: fhmtvggoojop58bdju4p
14   storageClassName: yc-network-hdd
```

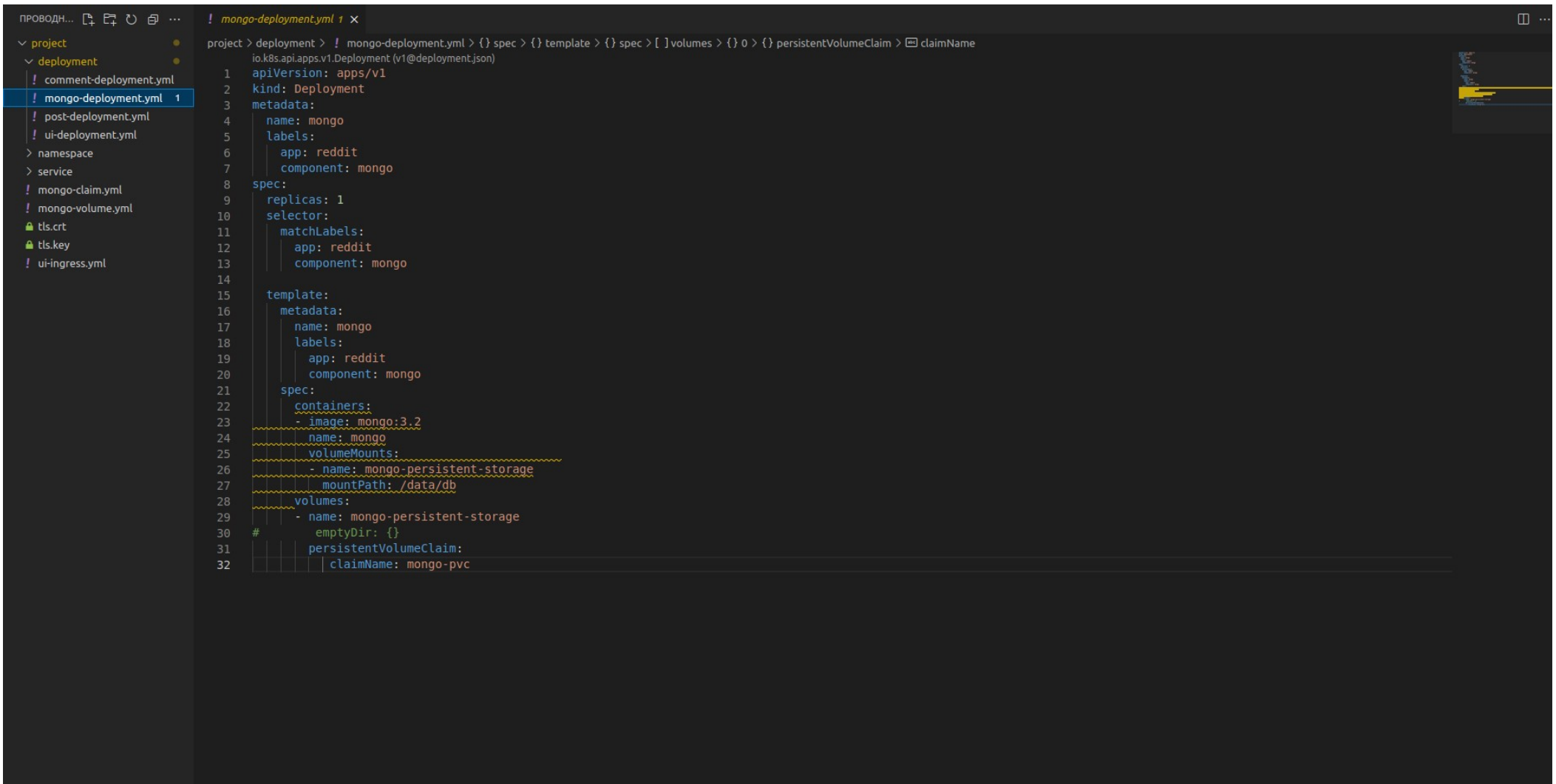
Создал манифест PersistentVolumeClaim чтобы выделить приложению часть дискового хранилища PersistentVolume. Claim - это именно запрос, а не само хранилище.

A screenshot of a code editor with a dark theme. On the left is a sidebar with a file explorer showing a project structure with folders 'deployment', 'namespace', and 'service', and files 'mongo-claim.yml', 'mongo-volume.yml', 'tls.crt', 'tls.key', and 'ui-ingress.yml'. The 'mongo-claim.yml' file is selected. The main editor area shows the content of 'mongo-claim.yml' with line numbers 1 through 12. The YAML content is as follows:

```
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mongo-pvc
5  spec:
6    accessModes:
7      - ReadWriteOnce
8    resources:
9      requests:
10       storage: 4Gi
11    volumeName: mongo-pv
12    storageClassName: yc-network-hdd
```



Прописал в манифесте, deployment для сервиса mongo, использование PersistentVolumeClaim



```
project > deployment > ! mongo-deployment.yml > {} spec > {} template > {} spec > [ ] volumes > {} 0 > {} persistentVolumeClaim > claimName
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mongo
5    labels:
6      app: reddit
7      component: mongo
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: reddit
13       component: mongo
14
15   template:
16     metadata:
17       name: mongo
18       labels:
19         app: reddit
20         component: mongo
21     spec:
22       containers:
23         - image: mongo:3.2
24           name: mongo
25           volumeMounts:
26             - name: mongo-persistent-storage
27               mountPath: /data/db
28           volumes:
29             - name: mongo-persistent-storage
30               emptyDir: {}
31             persistentVolumeClaim:
32               claimName: mongo-pvc
```

# Создал PersistentVolume и PersistentVolumeClaim Создал заново deployment для mongo.

```
alex@alex:~/myDoc/project$ kubectl delete -f deployment/mongo-deployment.yml -n dev
deployment.apps "mongo" deleted
alex@alex:~/myDoc/project$ kubectl apply -f mongo-volume.yml -n dev
persistentvolume/mongo-pv created
alex@alex:~/myDoc/project$ kubectl apply -f mongo-claim.yml -n dev
persistentvolumeclaim/mongo-pvc created
alex@alex:~/myDoc/project$ kubectl apply -f deployment/mongo-deployment.yml -n dev
deployment.apps/mongo created
alex@alex:~/myDoc/project$ kubectl get pv -n dev
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  REASON  AGE
mongo-pv      4Gi       RWO           Retain          Bound   dev/mongo-pvc  yc-network-hdd  28s
alex@alex:~/myDoc/project$ kubectl get pvc -n dev
NAME          STATUS  VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
mongo-pvc     Bound   mongo-pv        4Gi       RWO           yc-network-hdd  25s
alex@alex:~/myDoc/project$
```

The screenshot shows a terminal window with a file explorer on the left and a terminal output on the right. The file explorer shows a directory structure with files like `comment-deployment.yml`, `mongo-deployment.yml`, `post-deployment.yml`, `ui-deployment.yml`, `namespace`, `service`, `mongo-claim.yml`, `mongo-volume.yml`, `tls.crt`, `tls.key`, and `ui-ingress.yml`. The terminal output shows the following commands and their results:

- `kubectl delete -f deployment/mongo-deployment.yml -n dev`: `deployment.apps "mongo" deleted`
- `kubectl apply -f mongo-volume.yml -n dev`: `persistentvolume/mongo-pv created`
- `kubectl apply -f mongo-claim.yml -n dev`: `persistentvolumeclaim/mongo-pvc created`
- `kubectl apply -f deployment/mongo-deployment.yml -n dev`: `deployment.apps/mongo created`
- `kubectl get pv -n dev`:

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
mongo-pv	4Gi	RWO	Retain	Bound	dev/mongo-pvc	yc-network-hdd		28s
- `kubectl get pvc -n dev`:

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
mongo-pvc	Bound	mongo-pv	4Gi	RWO	yc-network-hdd	25s

Создал тестовую запись в приложении.  
Удалили и заново создал deployment для mongodb.  
По итогу видим что запись сохранилась значит все настроено верно  
и вся информация хранится в дисковом хранилище.

