

Kubernetes

Дано:

- Приложение запущенное в docker

Задание:

- Запустить приложение в kubernetes.

Имеем 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  :::9292->9292/tcp                  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

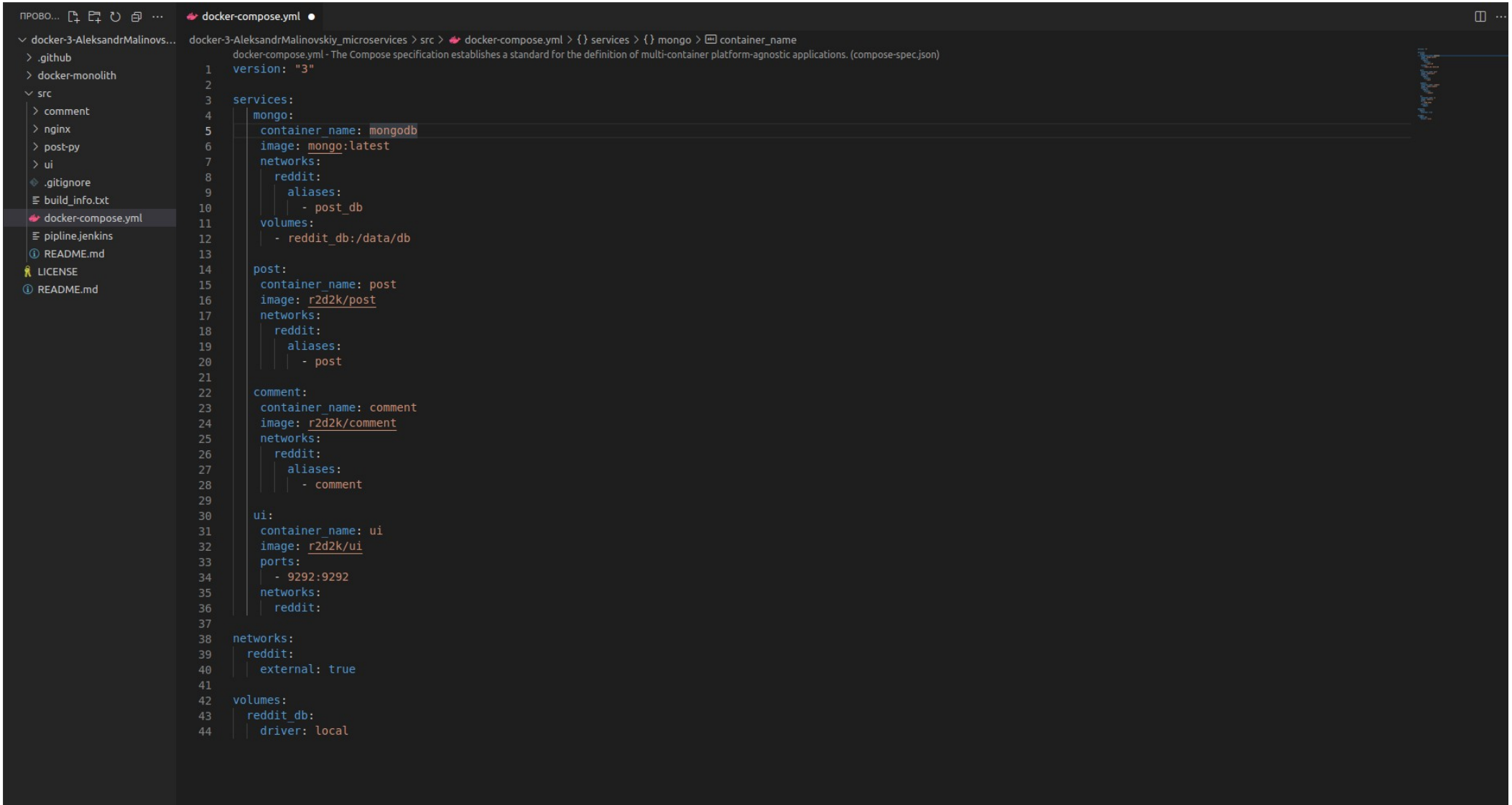
Microservices Reddit in 588814f79356 container

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 and 'reddit' network), 'post' (using 'r2d2k/post' image and 'reddit' network), 'comment' (using 'r2d2k/comment' image and 'reddit' network), and 'ui' (using 'r2d2k/ui' image, mapping port 9292, and 'reddit' network). A 'networks' section defines the 'reddit' network as external, and a 'volumes' section defines 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.

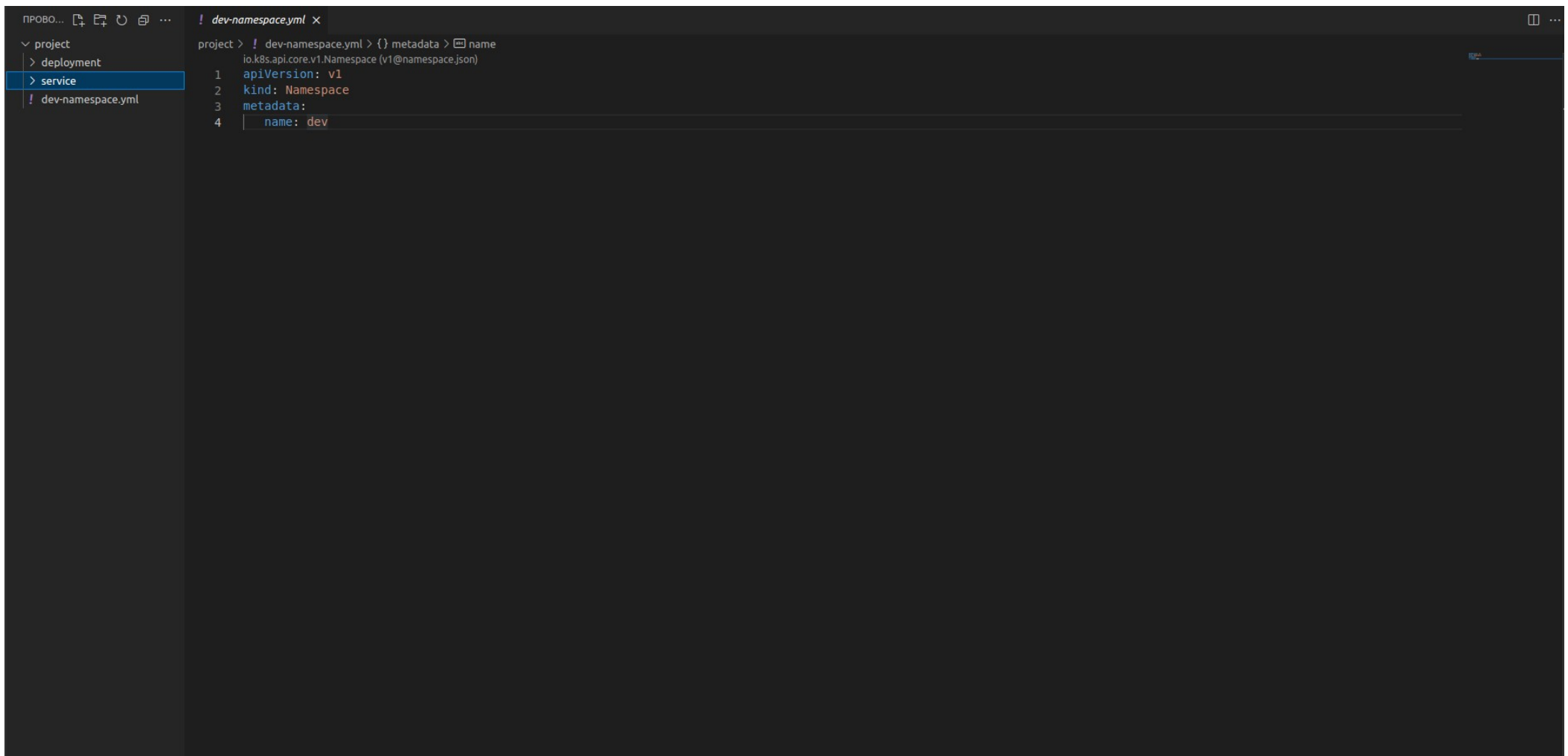
Что бы развернуть приложение необходимо создать для каждого сервиса deployment манифесты.

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.

Создадим 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)



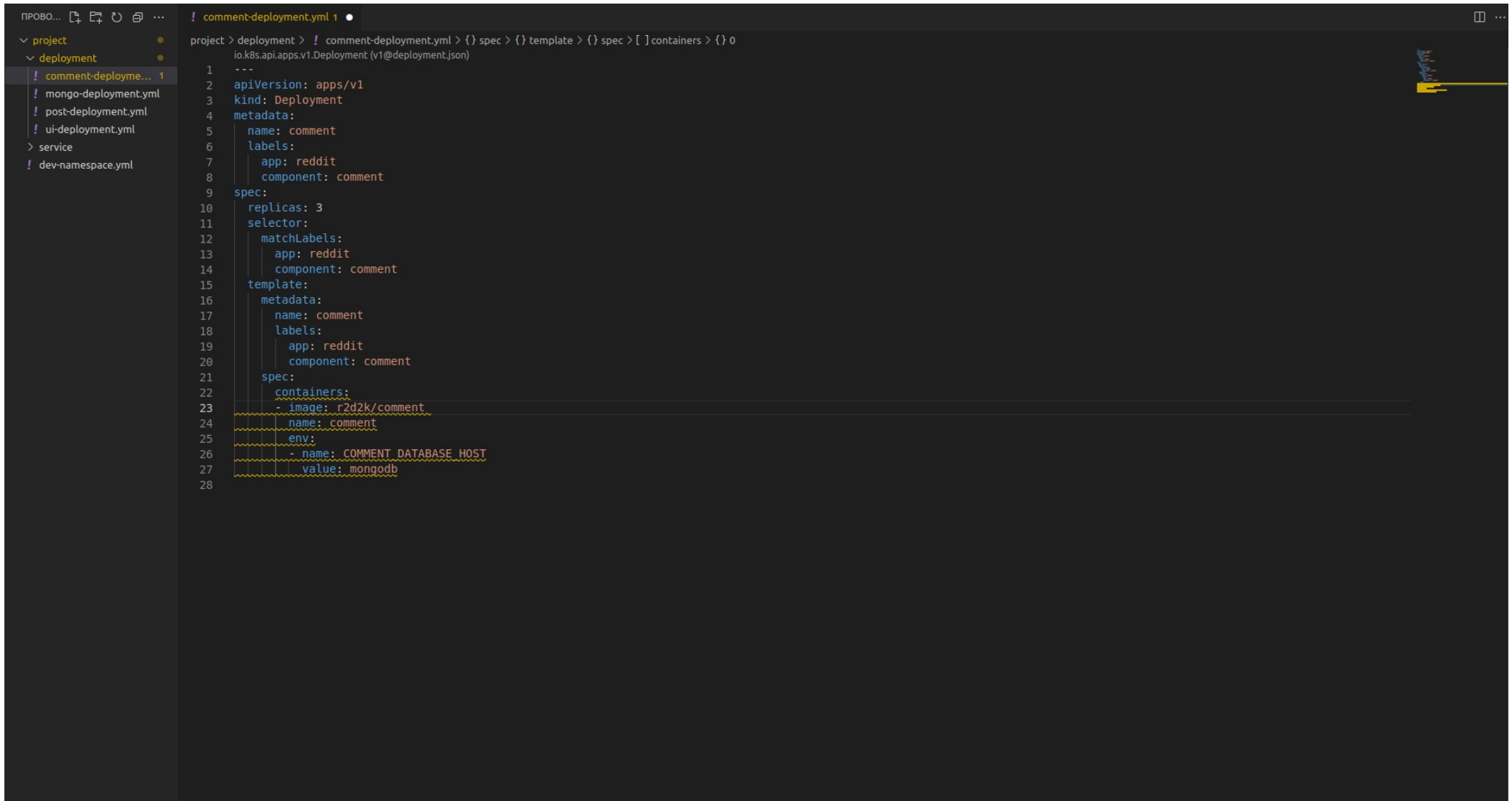
```
project > ! dev-namespace.yml > {} metadata > name
io.k8s.api.core.v1.Namespace (v1@namespace.json)
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: dev
```

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 snippet for creating a Namespace named 'dev'. The output is formatted with line numbers 1 through 4.

Deployment для сервиса comment

Количество реплик 3

В env указан COMMENT_DATABASE_HOST mongodb - это переменная окружения для обращения к базе



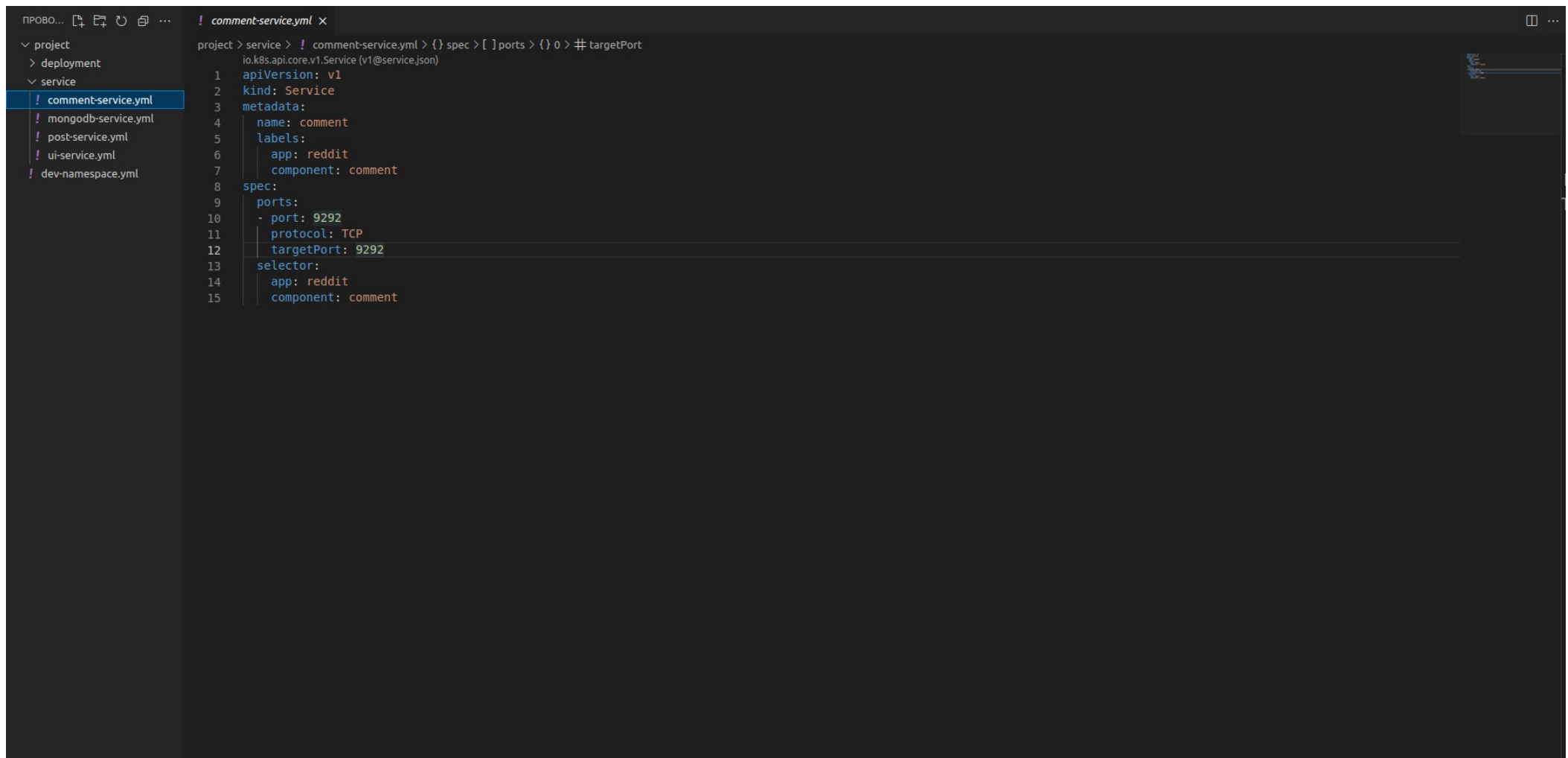
```
project > deployment > ! comment-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0
io.k8s.apl.apps.v1.Deployment (v1@deployment.json)

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.



The image shows a code editor with a file named `comment-service.yml` open. The left sidebar shows a project structure with folders for `project`, `deployment`, and `service`. Under `service`, several files are listed, including `comment-service.yml` which is currently selected. The main editor area displays the YAML content of this file, which defines a Kubernetes Service. The service is named `comment`, has a `kind` of `Service`, and is associated with the `app: reddit` and `component: comment` labels. It is configured to listen on port `9292` using `TCP` protocol and targets pods with the same labels. The editor also shows a breadcrumb navigation path: `project > service > ! comment-service.yml > {} spec > [] ports > {} 0 > # targetPort`.

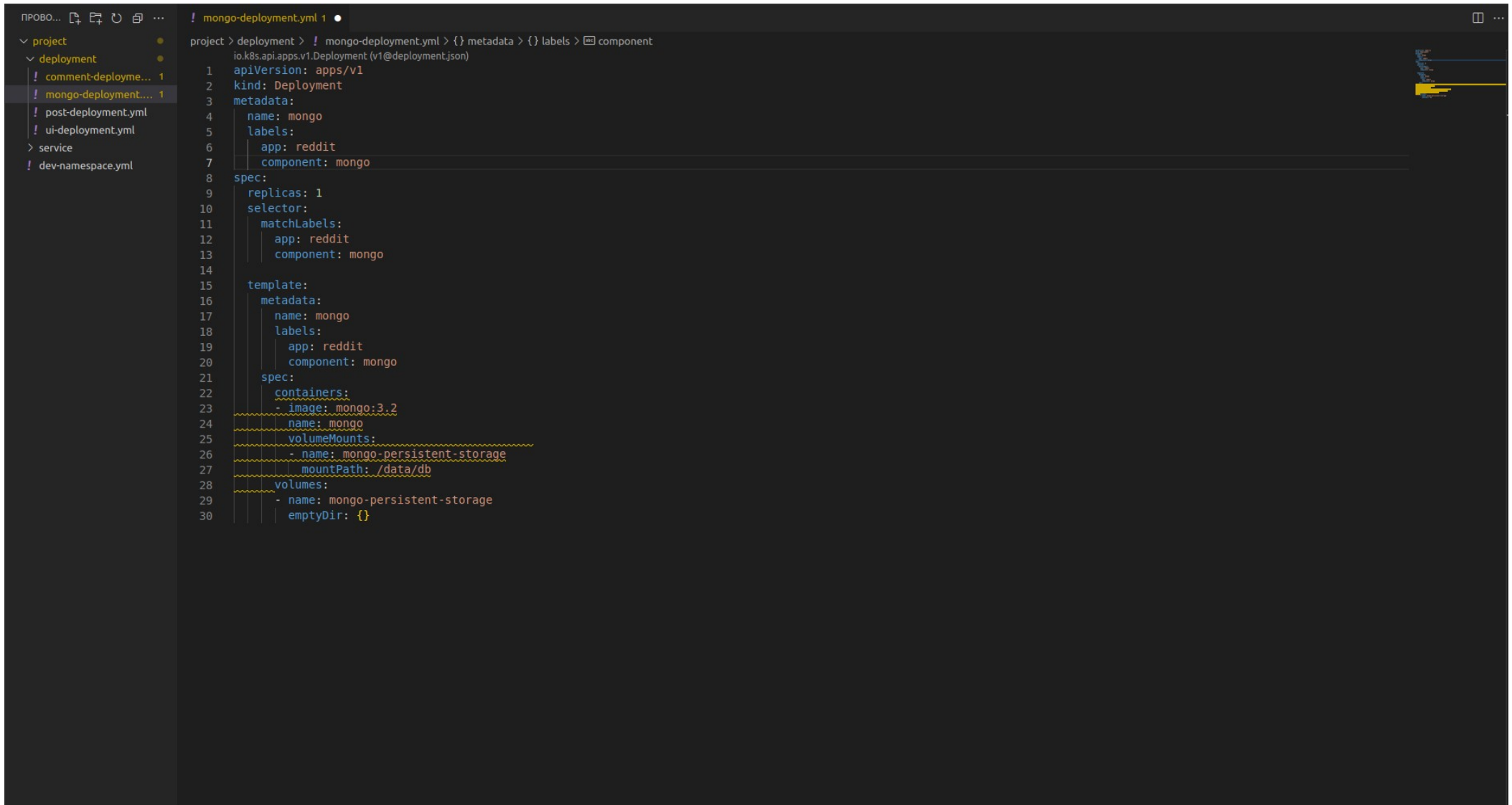
```
project > service > ! comment-service.yml > {} spec > [ ] ports > {} 0 > # targetPort
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: comment
5    labels:
6      app: reddit
7      component: comment
8  spec:
9    ports:
10     - port: 9292
11       protocol: TCP
12       targetPort: 9292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

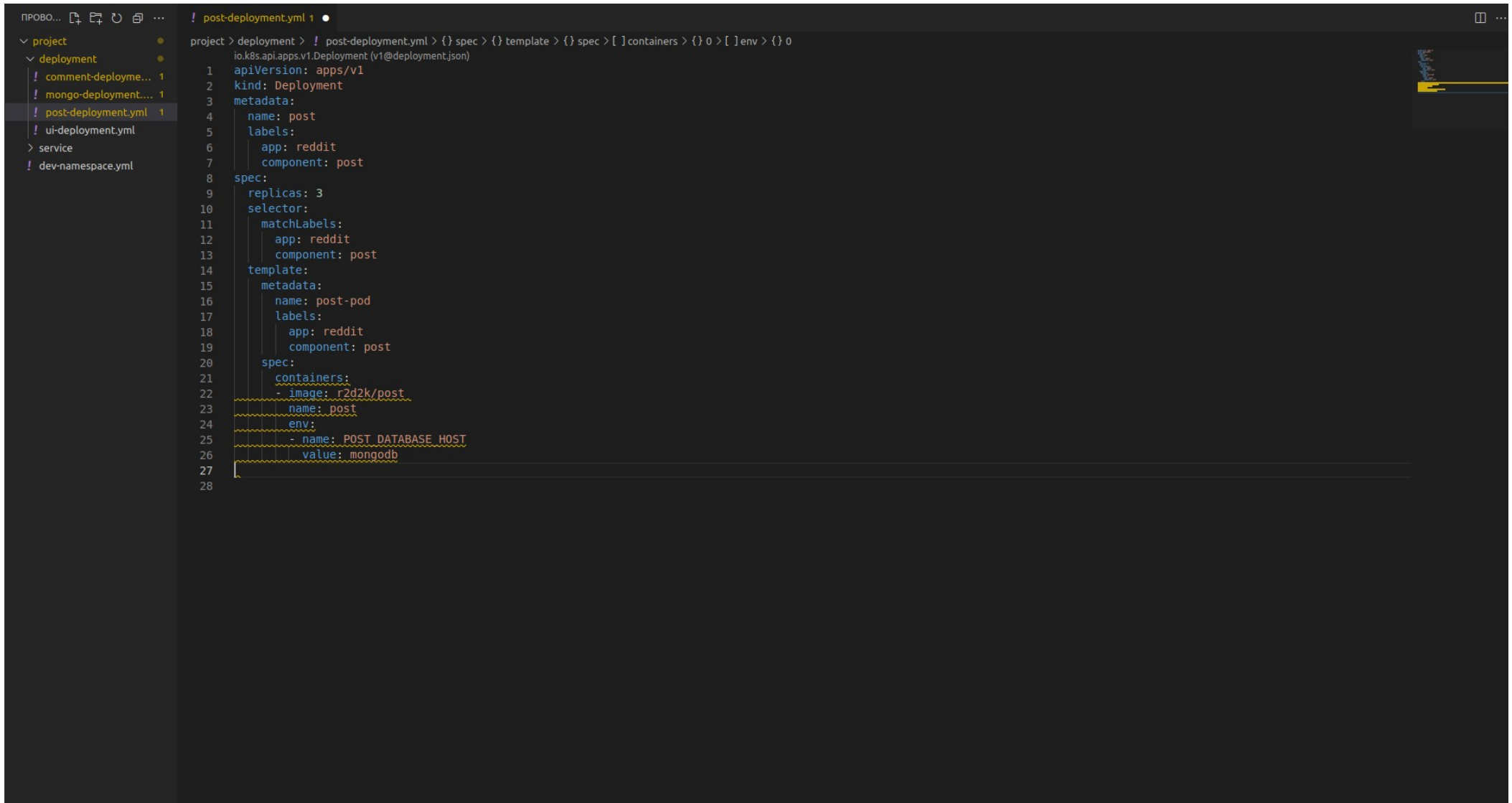
При обращении на mongodb:27017 изнутри любого из POD-ов текущего namespace нас переправит на 27017 порт приложения mongo, по выбранным label-ам.

```
project > service > ! mongodb-service.yml > {} metadata > name
io.k8s.api.core.v1.Service (v1@service.json)
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

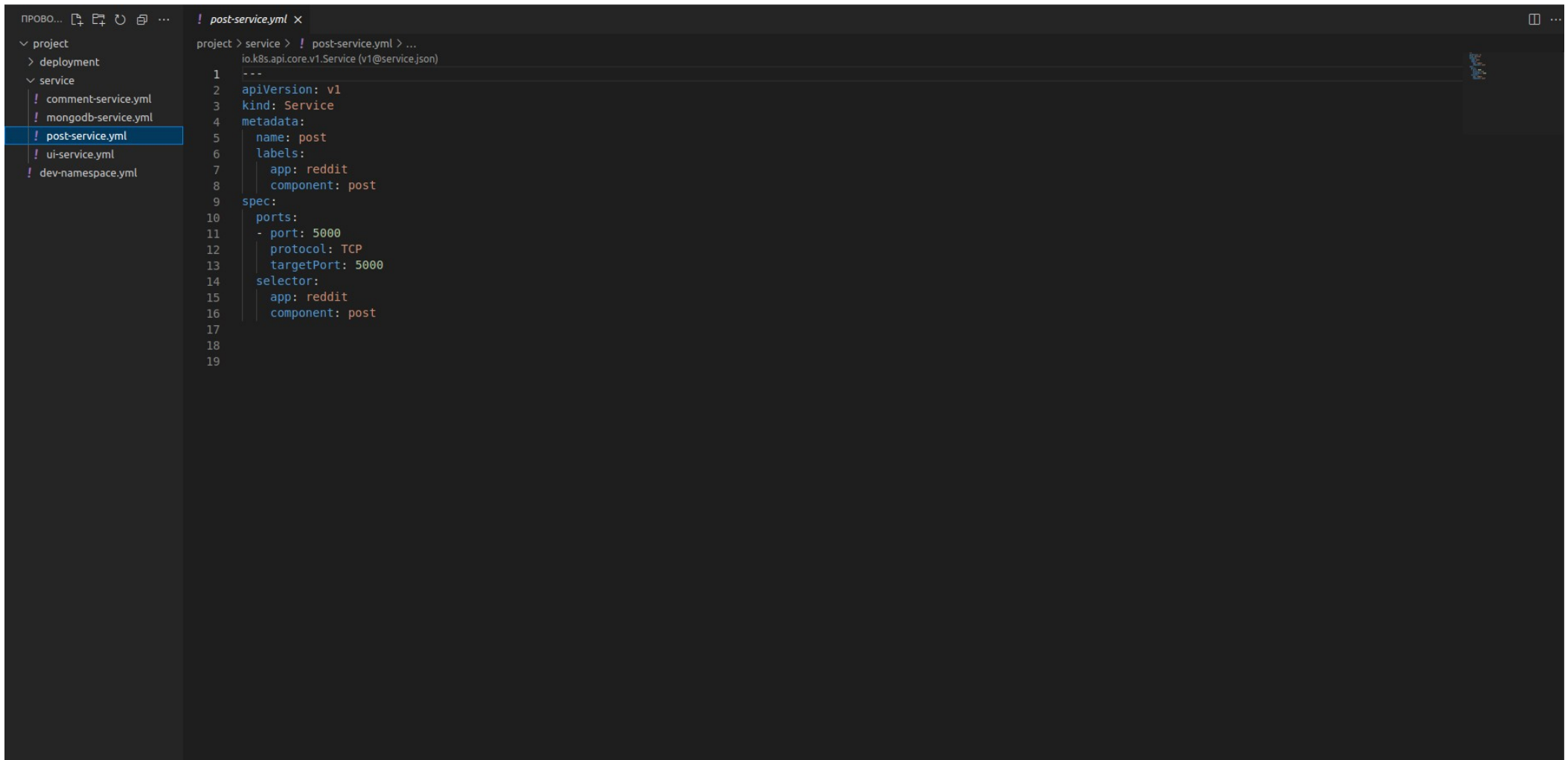


```
project > deployment > ! post-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] env > {} 0
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: post
5    labels:
6      app: reddit
7      component: post
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

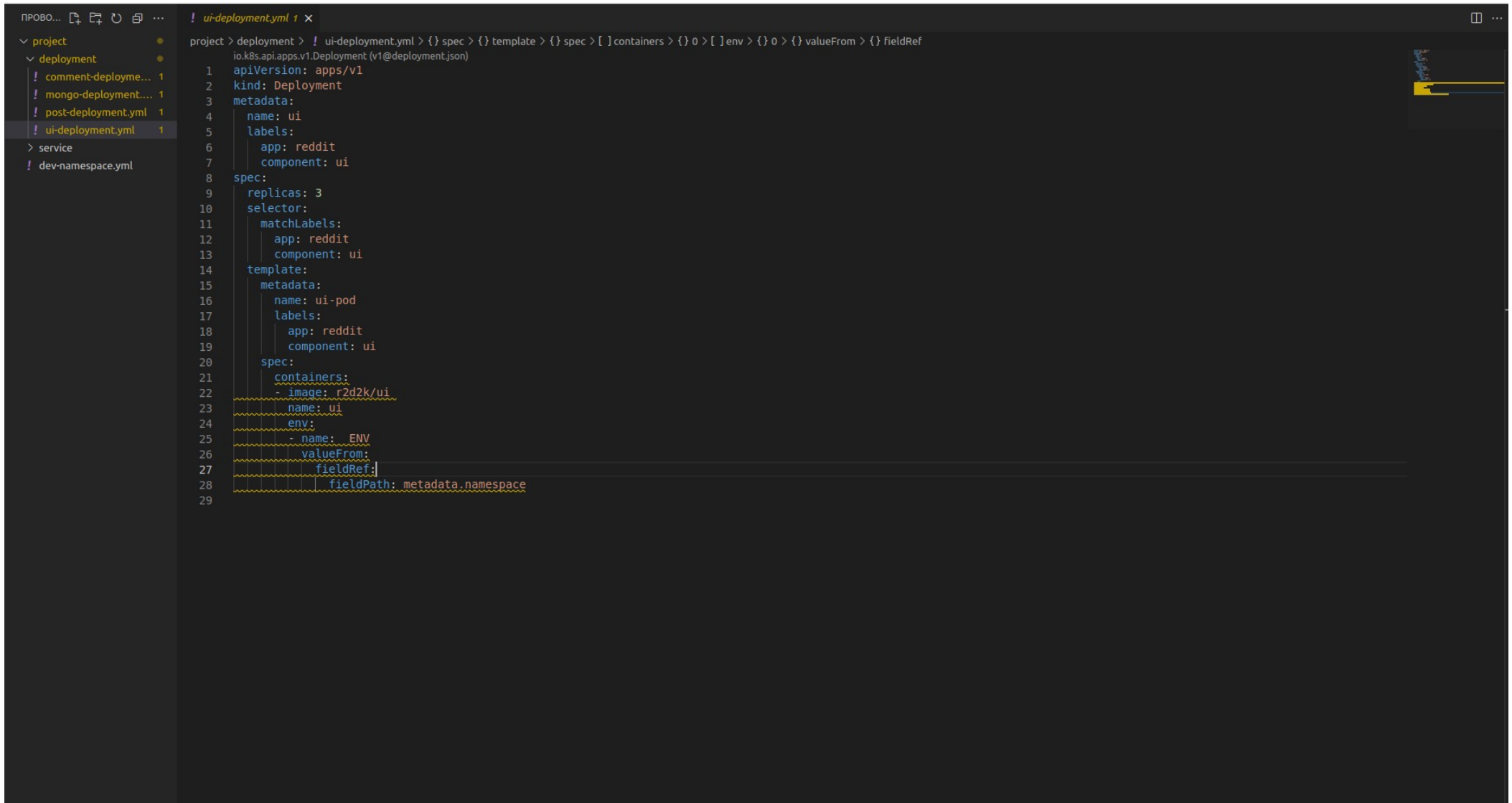
В DNS появится запись для post

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



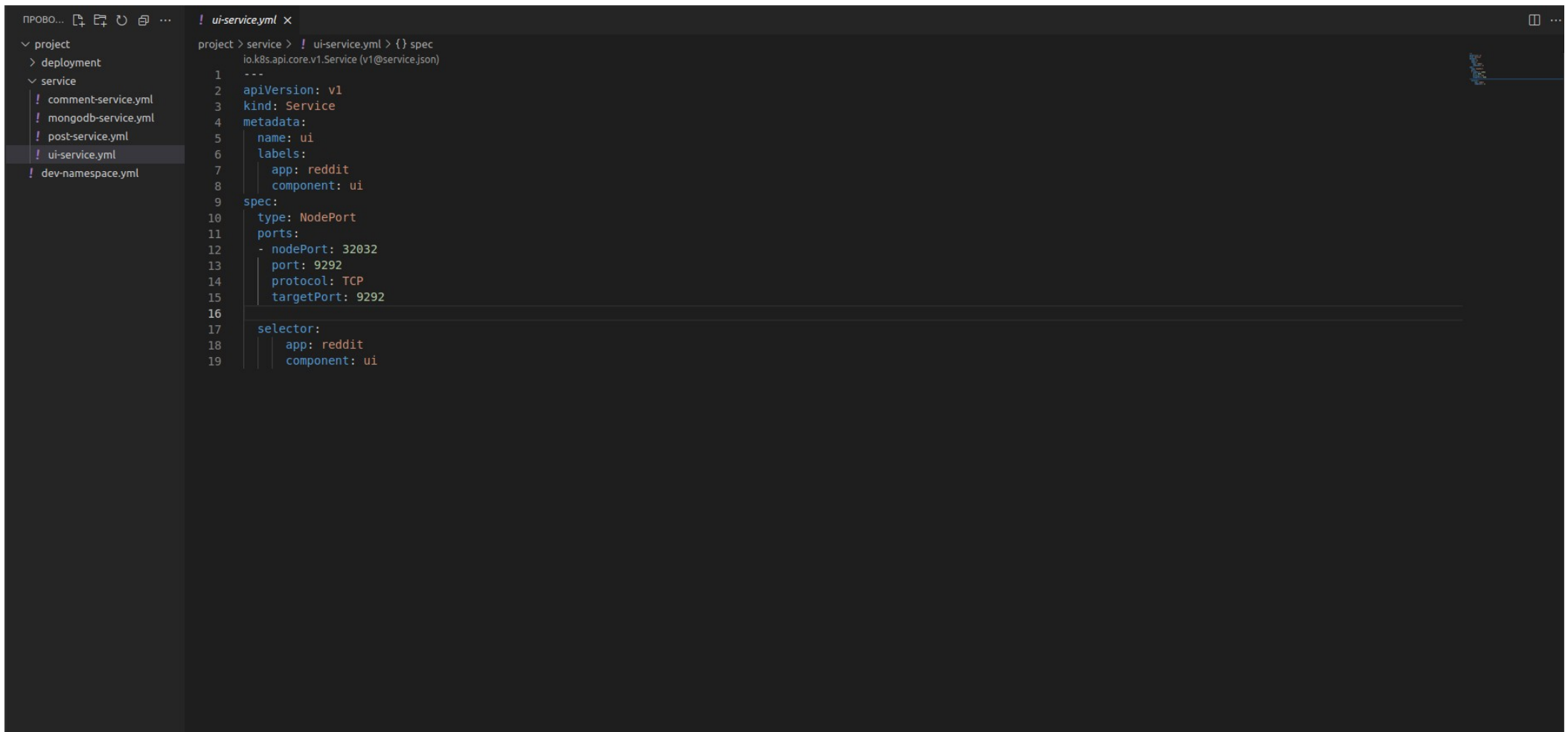
```
project > deployment > ! ui-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] env > {} 0 > {} valueFrom > {} fieldRef
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: ui
5    labels:
6      app: reddit
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 развернули в yandex cloud с двумя узлами. Создали 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$

```

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

```

  ▢  📄  🔄  🏠  ...
  ▾ 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$
```

Проверили один из созданных service.
Все другие можно проверить аналогично.
Для примера взяли service для comment.
Получили внутренний ip адрес из описания .
Далее сделали 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 указали постоянный порт доступный из вне 32032

Убедились что это так посмотрев это уже в самом кластере.

Выяснили 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$
```

Перешли в браузере по ip адресс ноды порт 32032.

Видим в названии dev это namespace, а ui-565b9d6499-xbtqm это название одного из Pod -а ui.

