

Kubernetes/Helm.

Дано:

Приложение запущенное в kubernetes.

Кластер развернут в yandex cloud.

Задание:

Сделать инфраструктуру декларативной и описать все в chart helm.

Запустить приложение через пакетный менеджер для Kubernetes - helm.

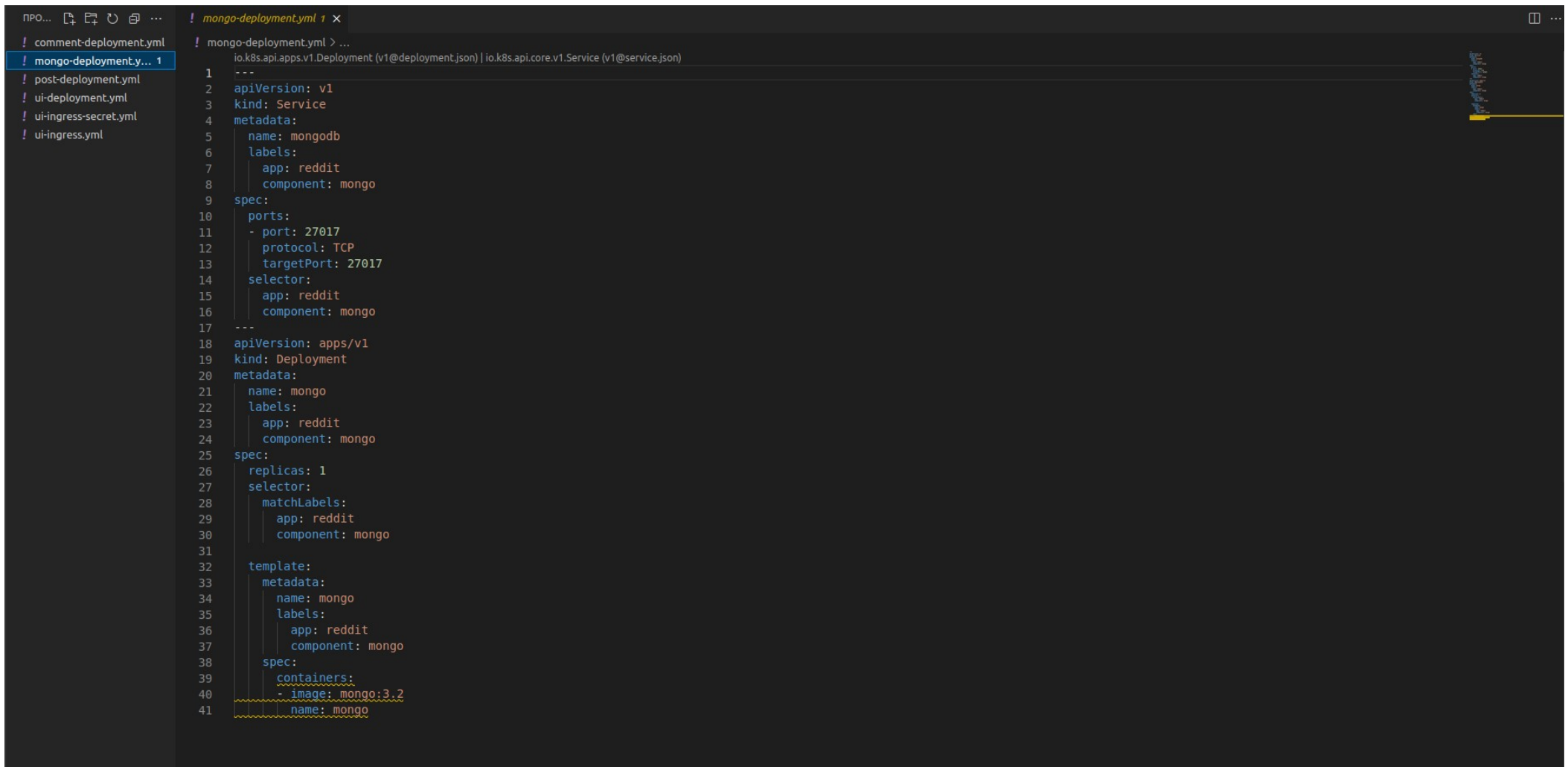
Манифест deployment и service для сервиса comment

```
! comment-deploymen... 1
! mongo-deployment.yml
! post-deployment.yml
! ui-deployment.yml
! ui-ingress-secret.yml
! ui-ingress.yml

! comment-deployment.yml > apiVersion
io.k8s.api.apps.v1.Deployment (v1@deployment.json) | io.k8s.api.core.v1.Service (v1@service.json)

1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: comment
6   labels:
7     app: reddit
8     component: comment
9 spec:
10   ports:
11   - port: 9292
12     protocol: TCP
13     targetPort: 9292
14   selector:
15     app: reddit
16     component: comment
17 ---
18 apiVersion: apps/v1
19 kind: Deployment
20 metadata:
21   name: comment
22   labels:
23     app: reddit
24     component: comment
25 spec:
26   replicas: 3
27   selector:
28     matchLabels:
29       app: reddit
30       component: comment
31   template:
32     metadata:
33       name: comment
34       labels:
35         app: reddit
36         component: comment
37     spec:
38       containers:
39       - image: r2d2k/comment
40         name: comment
41       env:
42       - name: COMMENT_DATABASE_HOST
43         value: mongodb
44
```

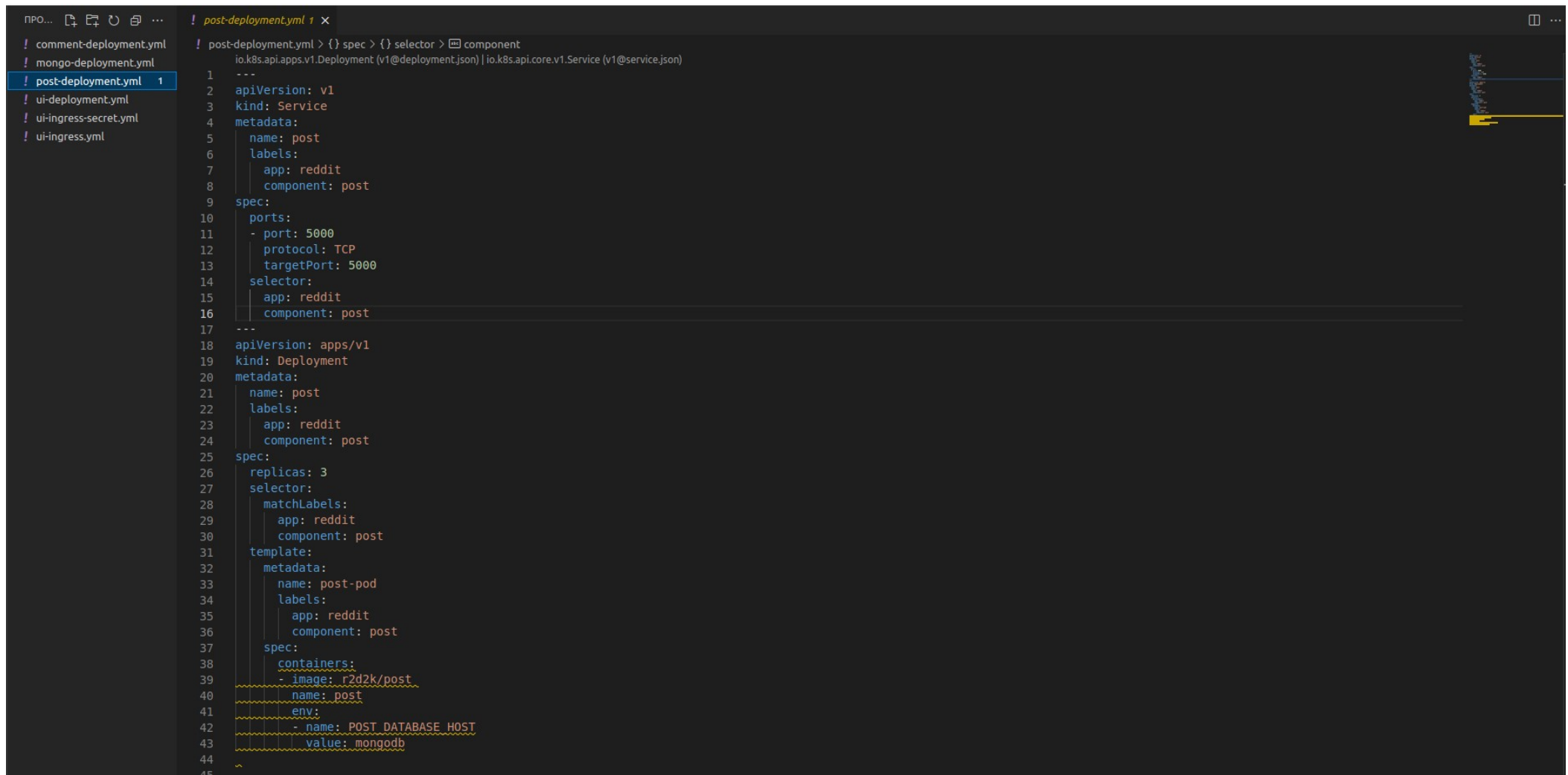
Манифест deployment и service для сервиса mongodb



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer lists several files: comment-deployment.yml, mongo-deployment.y... 1 (selected), post-deployment.yml, ui-deployment.yml, ui-ingress-secret.yml, and ui-ingress.yml. The code editor displays the content of mongo-deployment.y... 1, which is a Kubernetes manifest for a Service and a Deployment. The manifest is written in YAML and includes comments at the top indicating the source files for the Service and Deployment parts.

```
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  ---
18  apiVersion: apps/v1
19  kind: Deployment
20  metadata:
21    name: mongo
22    labels:
23      app: reddit
24      component: mongo
25  spec:
26    replicas: 1
27    selector:
28      matchLabels:
29        app: reddit
30        component: mongo
31    template:
32      metadata:
33        name: mongo
34        labels:
35          app: reddit
36          component: mongo
37      spec:
38        containers:
39          - image: mongo:3.2
40            name: mongo
```

Манифест deployment и service для сервиса post



```
! post-deployment.yaml 1 x
! comment-deployment.yaml
! mongo-deployment.yaml
! post-deployment.yaml 1
! ui-deployment.yaml
! ui-ingress-secret.yaml
! ui-ingress.yaml

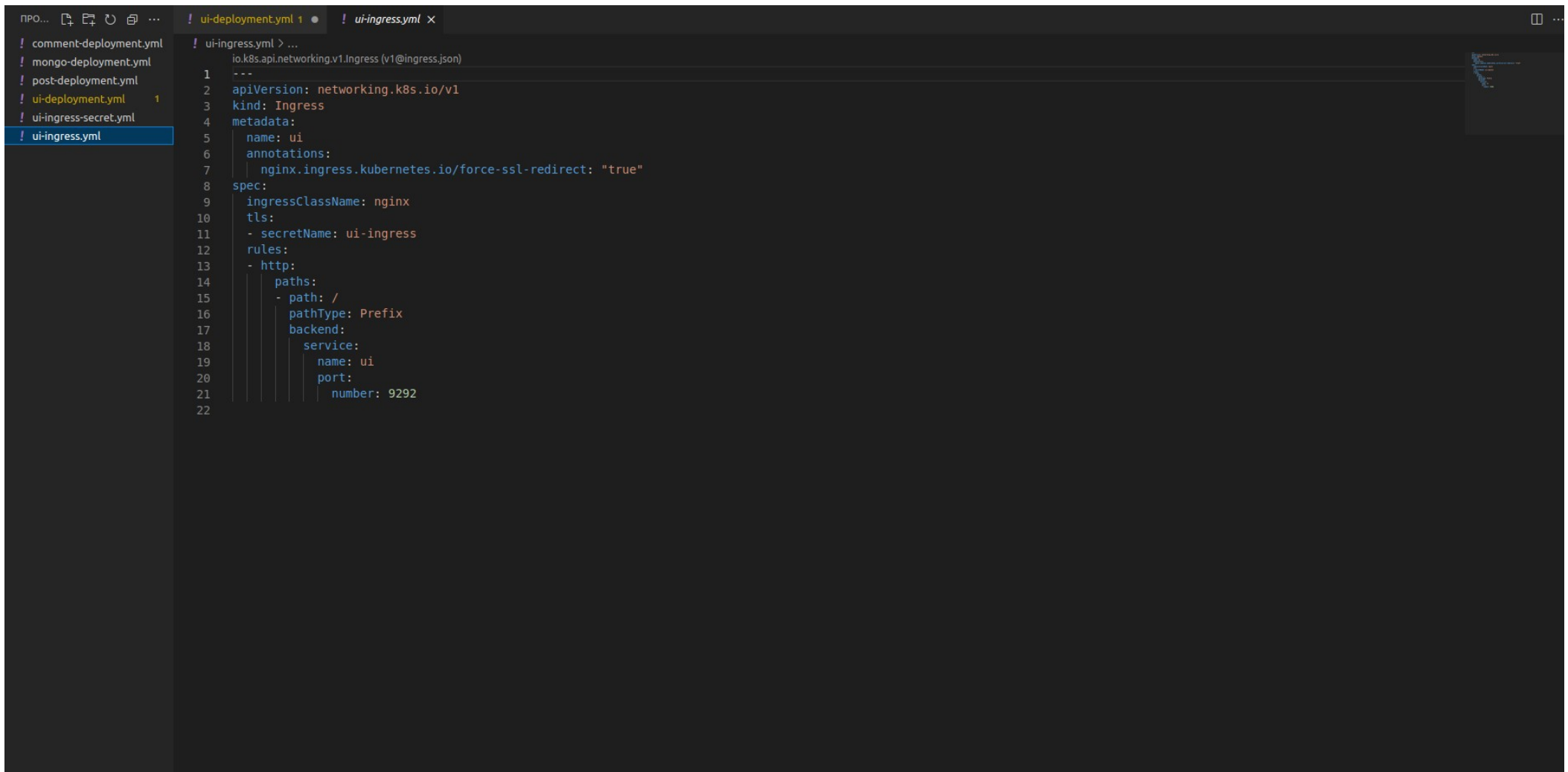
! post-deployment.yaml > {} spec > {} selector > component
io.k8s.api.apps.v1.Deployment (v1@deployment.json) | io.k8s.api.core.v1.Service (v1@service.json)

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  apiVersion: apps/v1
19  kind: Deployment
20  metadata:
21    name: post
22    labels:
23      app: reddit
24      component: post
25  spec:
26    replicas: 3
27    selector:
28      matchLabels:
29        app: reddit
30        component: post
31    template:
32      metadata:
33        name: post-pod
34        labels:
35          app: reddit
36          component: post
37      spec:
38        containers:
39          - image: r2d2k/post
40            name: post
41            env:
42              - name: POST_DATABASE_HOST
43                value: mongodb
44  ~
45  ~
```

Манифест deployment и service для сервиса ui

```
! ui-deployment.yml 1 ●
! ui-deployment.yml > {} spec > [ ] ports > {} 0 > # targetPort
v1@deployment.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    - port: 9292
13      protocol: TCP
14      targetPort: 9292
15  selector:
16    app: reddit
17    component: ui
18 ---
19 apiVersion: apps/v1
20 kind: Deployment
21 metadata:
22   name: ui
23   labels:
24     app: reddit
25     component: ui
26 spec:
27   replicas: 3
28   selector:
29     matchLabels:
30       app: reddit
31       component: ui
32   template:
33     metadata:
34       name: ui-pod
35       labels:
36         app: reddit
37         component: ui
38     spec:
39       containers:
40         - image: r2d2k/ui
41           name: ui
42           env:
43             - name: ENV
44               valueFrom:
45                 fieldRef:
46                   fieldPath: metadata.namespace
47
```

Для организации единой точки входа в приложения
снаружи, установлен ingress-nginx.
Манифест ingress.
С работой Ingress Controller как классический веб.

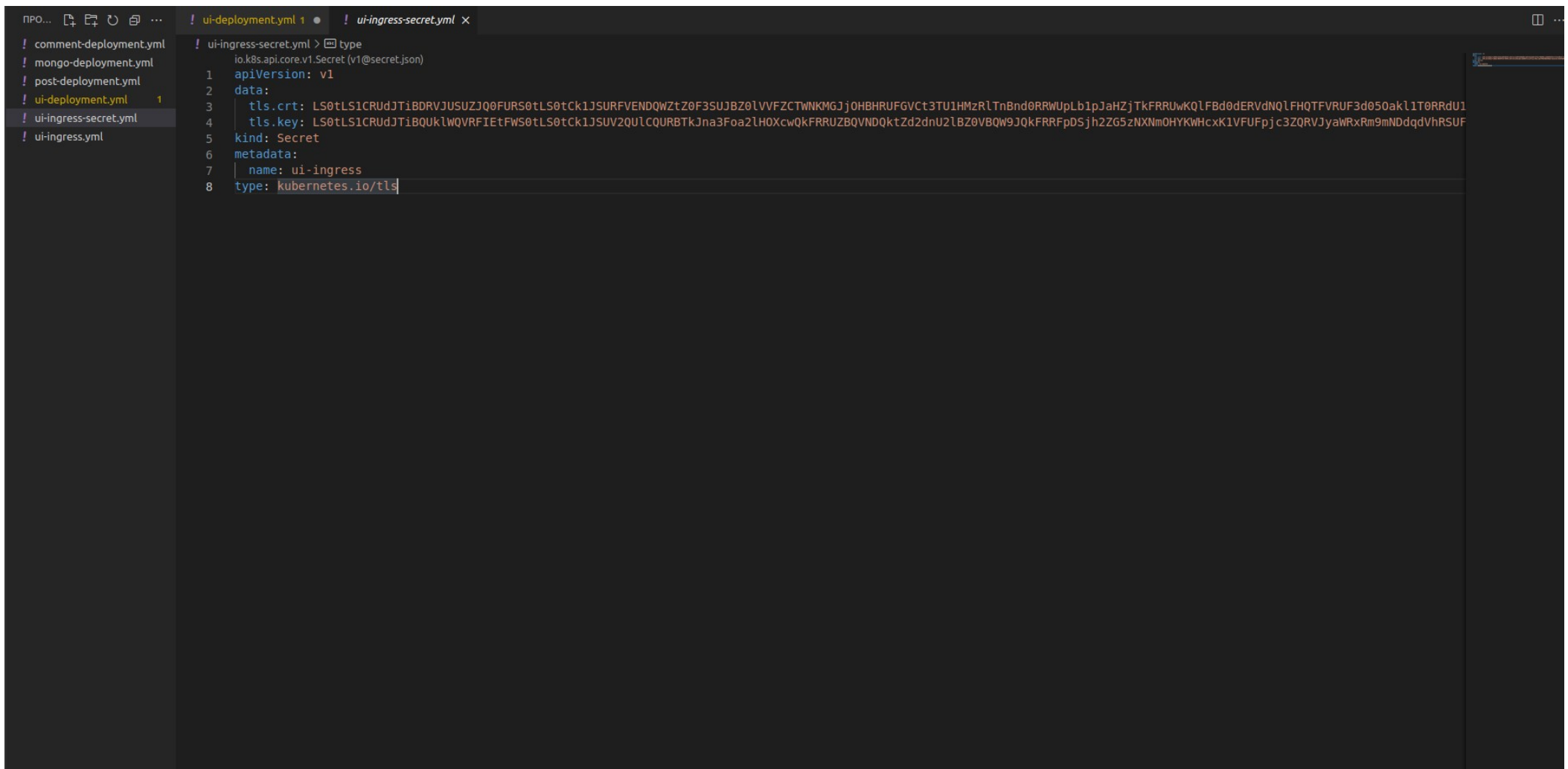


```
! comment-deployment.yml
! mongo-deployment.yml
! post-deployment.yml
! ui-deployment.yml 1
! ui-ingress-secret.yml
! ui-ingress.yml

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

Манифест secret для ingress

Самописные сертификаты ssl



The image shows a code editor with a dark theme. On the left, a file explorer lists several YAML files: comment-deployment.yml, mongo-deployment.yml, post-deployment.yml, ui-deployment.yml (highlighted with a yellow bar and the number 1), ui-ingress-secret.yml, and ui-ingress.yml. The main editor area displays the content of ui-ingress-secret.yml. The manifest is a Kubernetes Secret of type 'kubernetes.io/tls'. It includes a 'data' field with two entries: 'tls.crt' and 'tls.key', both containing long base64-encoded strings. The 'kind' is 'Secret', the 'name' is 'ui-ingress', and the 'type' is 'kubernetes.io/tls'.

```
! ui-deployment.yml 1  ! ui-ingress-secret.yml x
! comment-deployment.yml
! mongo-deployment.yml
! post-deployment.yml
! ui-deployment.yml 1
! ui-ingress-secret.yml
! ui-ingress.yml

! ui-ingress-secret.yml > type
io.k8s.api.core.v1.Secret (v1@secret.json)
1  apiVersion: v1
2  data:
3    |
4    |   tls.crt: LS0tLS1CRudJTlBDRVJUSUZJQ0FURSc0tLS0tCk1JSURFVENDQWZtZ0F3SUJBZ0lVVFZCTWNKMGIjOHBHRUFGVCT3TU1HMzRlTnBnd0RRWUpLb1pJaHZjTkFRRUwKQlFBd0dERVdNQlFHQTFVRUF3d050akl1T0RRdU1
5    |   tls.key: LS0tLS1CRudJTlBQUklwQVRFIEtFWS0tLS0tCk1JSUV2QUlCQURBTkJna3Foa2lHOXcwQkFRRUZBQVNDQktZd2dnU2lBZ0VBQW9JQkFRFRFpDSjh2ZG5zNXNmOHYKWHcxK1VFUFpj c3ZQRVJyaWRxRm9mNDdqdVhRSUF
6  kind: Secret
7  metadata:
8    |   name: ui-ingress
9  type: kubernetes.io/tls
```

Состояние kubernetes кластера

```
PROБЛЕМЫ 3 КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ВЫХОДНЫЕ ДАННЫЕ
! comment-deployment.yml
! mongo-deployment.yml
! post-deployment.yml
! ui-deployment.yml
! ui-ingress-secret.yml
! ui-ingress.yml

alex@alex:~/myDoc/project$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
comment-7755456d6c-c42cr  1/1     Running   0           66s
comment-7755456d6c-m5z6g  1/1     Running   0           67s
comment-7755456d6c-ndlzp  1/1     Running   0           66s
mongo-86f87568cd-zt7c9    1/1     Running   0           66s
post-b659fc678-5sks2     1/1     Running   0           66s
post-b659fc678-td74p     1/1     Running   0           66s
post-b659fc678-xgqps     1/1     Running   0           66s
ui-565b9d6499-b8n7j      1/1     Running   0           66s
ui-565b9d6499-lqlfs      1/1     Running   0           66s
ui-565b9d6499-snsnz      1/1     Running   0           66s

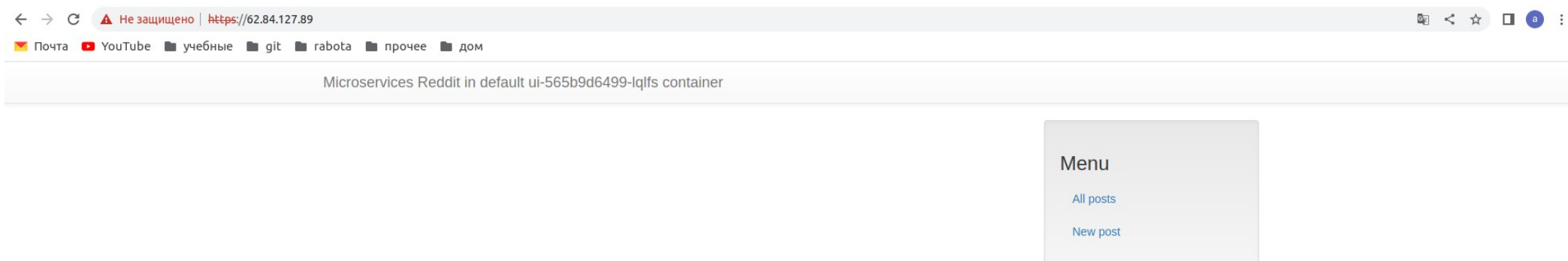
alex@alex:~/myDoc/project$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
comment 3/3      3             3           75s
mongo   1/1      1             1           74s
post    3/3      3             3           74s
ui      3/3      3             3           74s

alex@alex:~/myDoc/project$ kubectl get service
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
comment     ClusterIP   10.96.218.163 <none>         9292/TCP        80s
kubernetes  ClusterIP   10.96.128.1   <none>         443/TCP          131m
mongodb     ClusterIP   10.96.180.98  <none>         27017/TCP        79s
post        ClusterIP   10.96.222.16  <none>         5000/TCP        79s
ui          NodePort    10.96.172.27  <none>         9292:32416/TCP   79s

alex@alex:~/myDoc/project$ kubectl get ingress
NAME CLASS HOSTS          ADDRESS        PORTS    AGE
ui   nginx * 62.84.127.89  80, 443      85s

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


Приложение доступно по ip присвоенному в ingress



Порядок действий

Создаем директорию charts.

В ней создаем директории comment, post, ui, mongodb и application.

Директории comment, post, ui, mongodb будут содержать в себе файлы Chart.yaml (описание Chart-a) и values.yaml (переменные), а так же директорию templates в которой будут находиться манифесты deployment и service.

В директории application будет находиться единый Chart, который объединит все сервисы а так же глобальные переменные.

Как по итогу должна выглядеть директория Charts.

```
ПРОБЛЕМЫ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ

alex@alex:~/myDoc/учеба/project/charts$ tree
.
├── application
│   ├── Chart.yaml
│   ├── requirements.yaml
│   └── values.yaml
├── comment
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   └── values.yaml
├── mongodb
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   └── values.yaml
├── post
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   └── values.yaml
└── ui
    ├── Chart.yaml
    ├── templates
    │   ├── deployment.yaml
    │   ├── ingress-secret.yaml
    │   ├── ingress.yaml
    │   └── service.yaml
    └── values.yaml

9 directories, 21 files
alex@alex:~/myDoc/учеба/project/charts$
```

application - директория с чартом которые объединит все чарты

comment - директория с чартом для сервиса comment

mongodb - директория с чартом для сервиса mongodb

post - директория с чартом для сервиса post

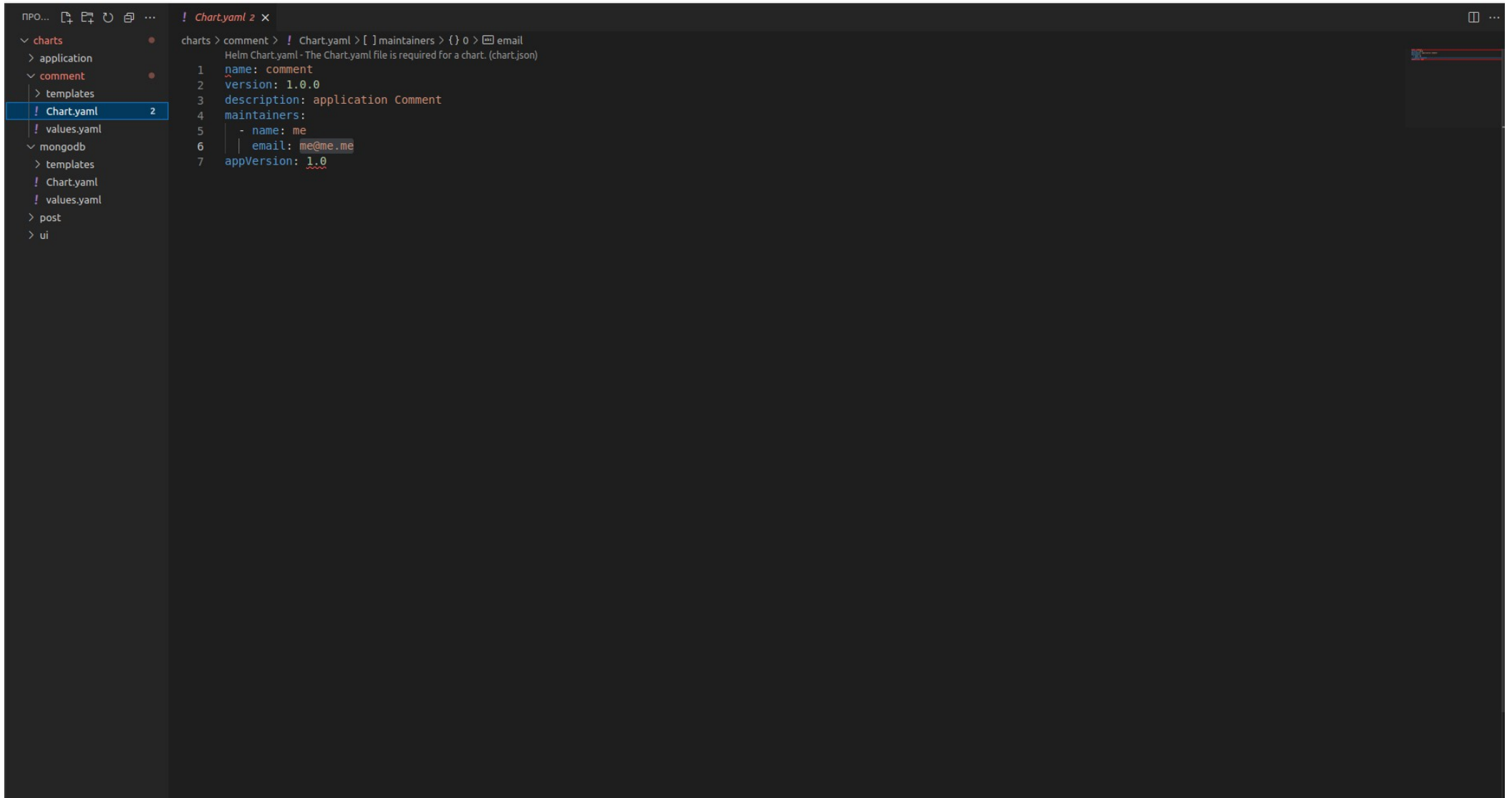
ui - директория с чартом для сервиса ui

Файл Chart.yaml описание чарта

Файл values.yaml переменные чарта

Файлы в директории templates все манифесты относящиеся к чарту

Создал Chart.yaml для сервиса comment

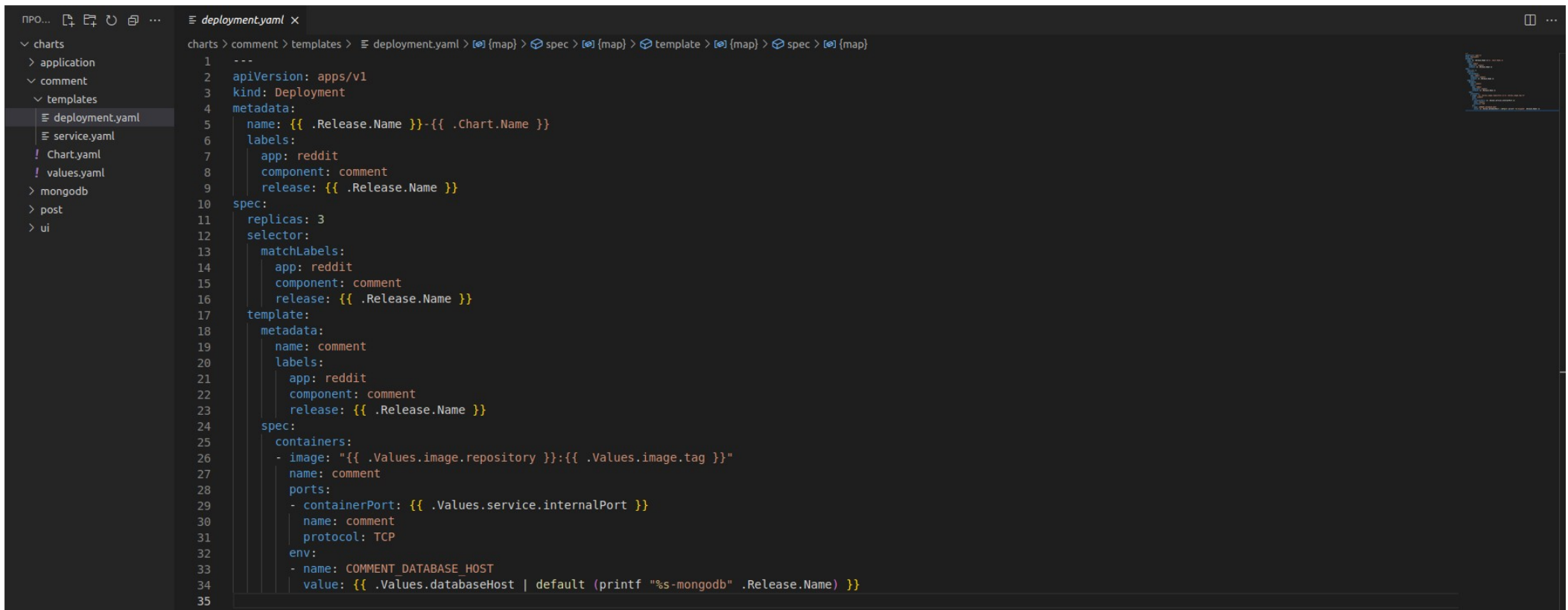


The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar displays a directory structure: 'charts' (expanded) contains 'application', 'comment' (expanded), and 'templates'. Under 'comment', there is a 'Chart.yaml' file (highlighted with a blue bar and the number '2') and a 'values.yaml' file. Other directories like 'mongodb', 'post', and 'ui' are also visible. The main editor area shows the content of 'Chart.yaml' with line numbers 1 through 7. The file contains the following YAML content:

```
1 name: comment
2 version: 1.0.0
3 description: application Comment
4 maintainers:
5   - name: me
6     email: me@me.me
7 appVersion: 1.0
```

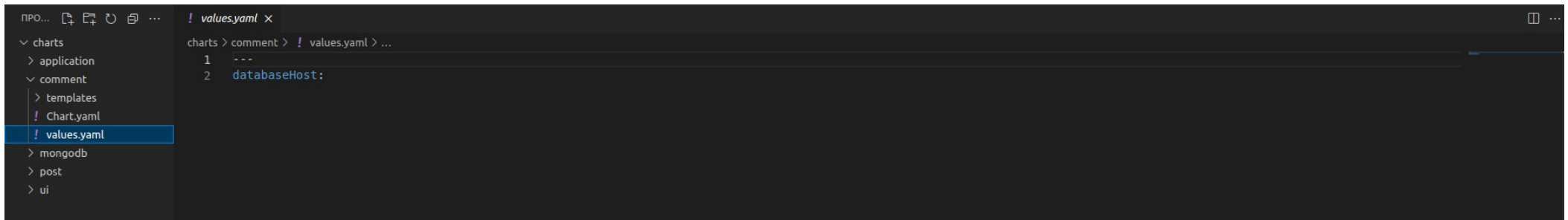
At the top of the editor, a breadcrumb trail reads: 'charts > comment > ! Chart.yaml > [] maintainers > { } 0 > email'. A status bar at the bottom indicates the file is 'Chart.yaml' and is 2 lines long.

Создал манифест deployment для сервиса comment
Переменные такие как Release.Name(имя релиза) и Chart.Name
(имя чарта) назначаются автоматически.
Переменная Values.service.internalPort будет определена в
директории в директории application.
Так как в директории application находится Chart, который
объединит все сервисы, переменные указанные в этой директории
будут перезаписывать значения переменных из зависимых чартов.



```
PRO... deployment.yaml X
charts > comment > templates > deployment.yaml > [map] > spec > [map] > template > [map] > spec > [map]
1 ---
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: {{ .Release.Name }}-{{ .Chart.Name }}
6   labels:
7     app: reddit
8     component: comment
9     release: {{ .Release.Name }}
10 spec:
11   replicas: 3
12   selector:
13     matchLabels:
14       app: reddit
15       component: comment
16       release: {{ .Release.Name }}
17   template:
18     metadata:
19       name: comment
20       labels:
21         app: reddit
22         component: comment
23         release: {{ .Release.Name }}
24     spec:
25       containers:
26       - image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
27         name: comment
28         ports:
29         - containerPort: {{ .Values.service.internalPort }}
30           name: comment
31           protocol: TCP
32       env:
33       - name: COMMENT_DATABASE_HOST
34         value: "{{ .Values.databaseHost | default (printf \"%s-mongodb\" .Release.Name) }}"
35
```

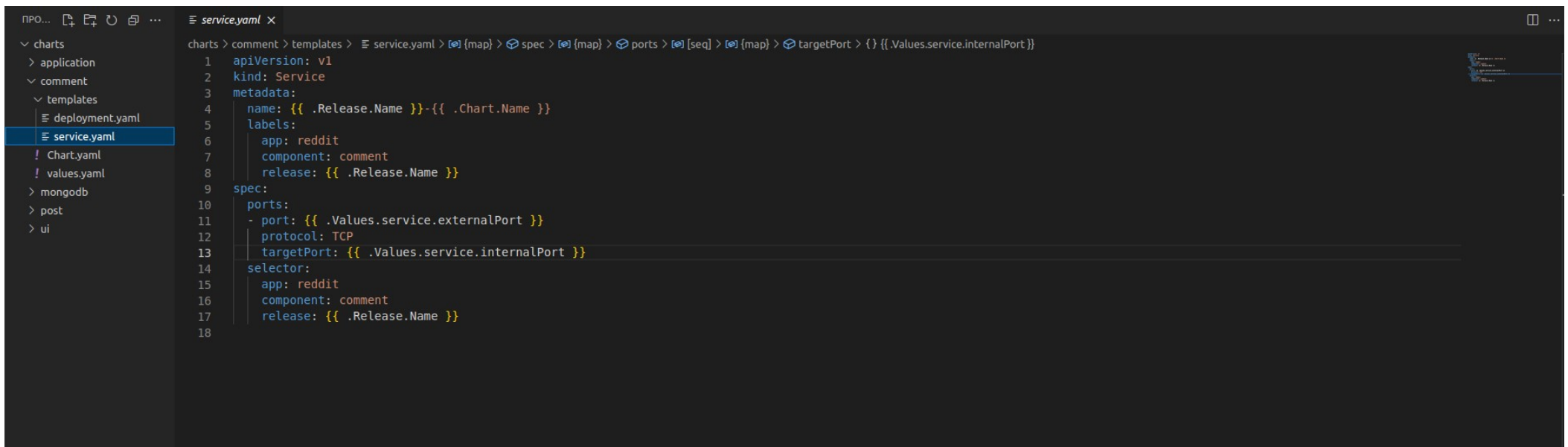
Создал values.yaml в нем определил единственную переменную Values.databaseHost



The screenshot shows the Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for a Helm chart named 'comment'. The 'values.yaml' file is selected. The code editor shows the content of 'values.yaml' with a single variable 'databaseHost' defined.

```
1 ---
2 databaseHost:
```

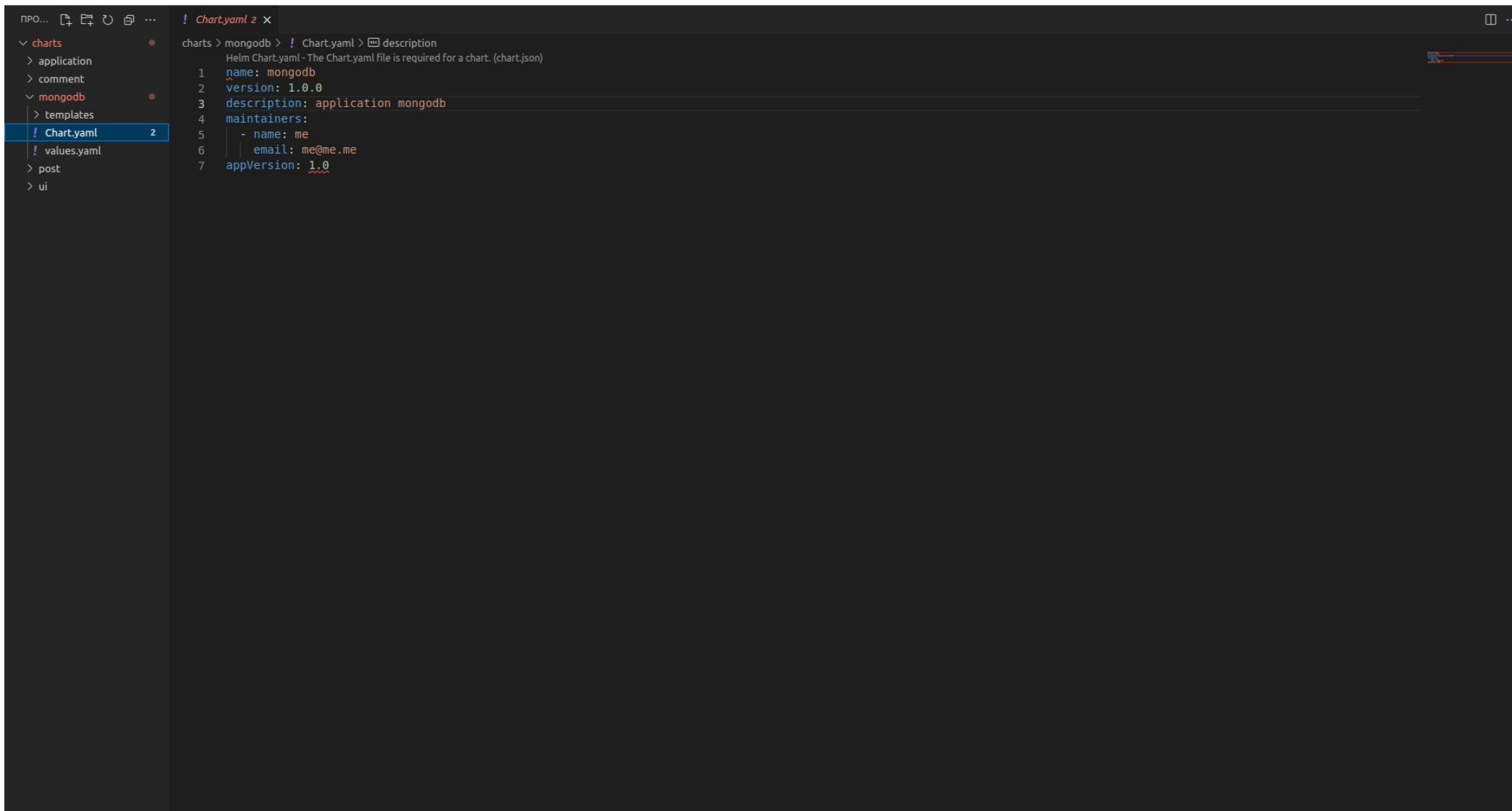
Создал манифест service для сервиса comment.
Переменные Release.Name(имя релиза) и Chart.Name (имя чарта) назначаются автоматически.
Переменная Values.service.internalPort и Values.service.externalPort будут определены в директории в директории application.



The screenshot shows the Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for a Helm chart named 'comment'. The 'service.yaml' file is selected. The code editor shows the content of 'service.yaml' with a service definition for 'comment'.

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: {{ .Release.Name }}-{{ .Chart.Name }}
5   labels:
6     app: reddit
7     component: comment
8     release: {{ .Release.Name }}
9 spec:
10   ports:
11     - port: {{ .Values.service.externalPort }}
12       protocol: TCP
13       targetPort: {{ .Values.service.internalPort }}
14   selector:
15     app: reddit
16     component: comment
17     release: {{ .Release.Name }}
```

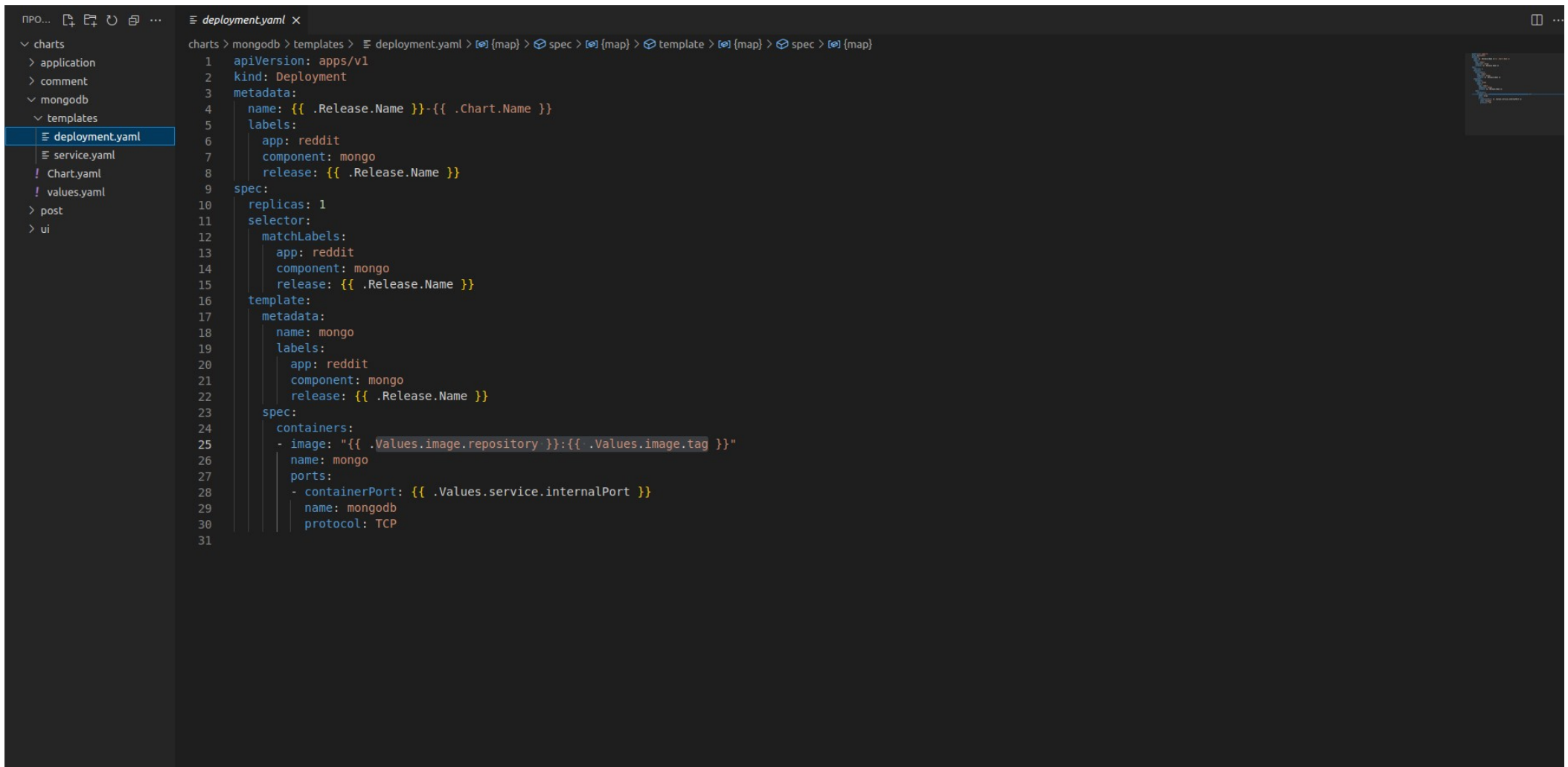
Создал Chart.yaml для сервиса mongodb



The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar displays a directory structure: 'charts' (expanded) contains 'application', 'comment', 'mongodb' (expanded), 'templates', 'Chart.yaml' (selected and highlighted in blue), 'values.yaml', 'post', and 'ui'. The main editor area shows the content of 'Chart.yaml' with line numbers 1 through 7. The file path at the top of the editor is 'charts > mongodb > ! Chart.yaml > description'. The content of the file is as follows:

```
1 name: mongodb
2 version: 1.0.0
3 description: application mongodb
4 maintainers:
5   - name: me
6     email: me@me.me
7 appVersion: 1.0
```

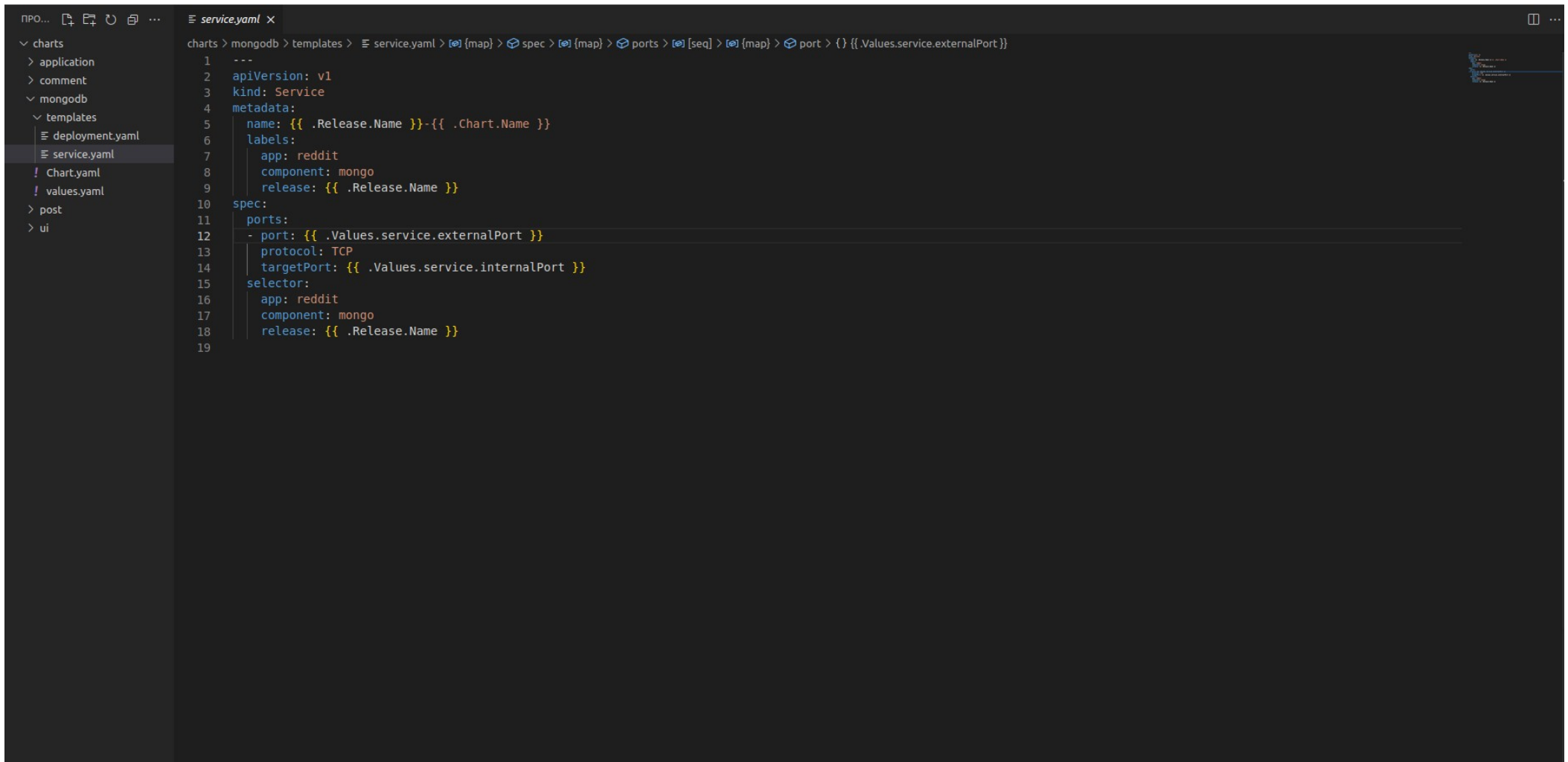
Создал манифест deployment для сервиса mongodb.
Переменные такие как Release.Name(имя релиза) и Chart.Name
(имя чарта) назначаются автоматически.
Переменные Values.service.internalPort, Values.image.repository,
Values.image.tag, будут определены в директории application.



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 directory structure with 'charts' containing 'application', 'comment', 'mongodb', and 'templates'. The 'templates' directory contains 'deployment.yaml', 'service.yaml', 'Chart.yaml', and 'values.yaml'. The 'deployment.yaml' file is selected and its content is displayed in the code editor. The code is a Kubernetes deployment manifest for MongoDB, using Helm templating syntax. It includes fields for apiVersion, kind, metadata (name, labels), spec (replicas, selector, matchLabels, template), and containers (image, name, ports). The manifest uses Helm's templating syntax to inject values from the chart's values file and the release's metadata.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: {{ .Release.Name }}-{{ .Chart.Name }}
5    labels:
6      app: reddit
7      component: mongo
8      release: {{ .Release.Name }}
9  spec:
10   replicas: 1
11   selector:
12     matchLabels:
13       app: reddit
14       component: mongo
15       release: {{ .Release.Name }}
16   template:
17     metadata:
18       name: mongo
19     labels:
20       app: reddit
21       component: mongo
22       release: {{ .Release.Name }}
23   spec:
24     containers:
25       - image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
26         name: mongo
27         ports:
28           - containerPort: {{ .Values.service.internalPort }}
29             name: mongodb
30             protocol: TCP
31
```

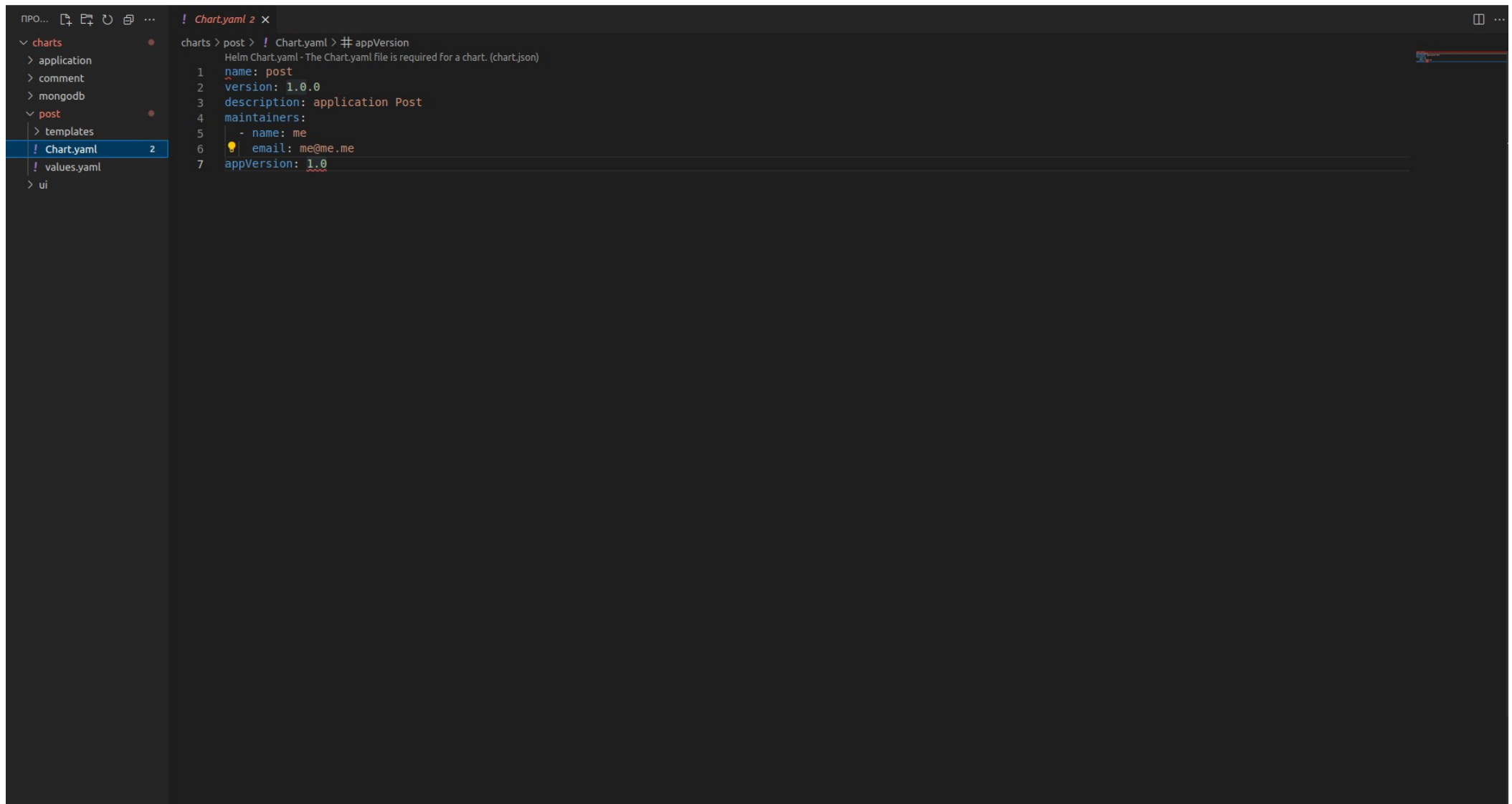

Создал манифест service для сервиса mongodb.
Переменные Release.Name(имя релиза) и Chart.Name (имя чарта)
назначаются автоматически.
Переменная Values.service.internalPort и Values.service.externalPort
будут определены в директории application.



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 directory structure with 'charts' containing 'application', 'comment', 'mongodb', and 'templates'. The 'templates' directory contains 'deployment.yaml', 'service.yaml', 'Chart.yaml', and 'values.yaml'. The 'service.yaml' file is selected and its content is displayed in the code editor. The code is a Kubernetes service manifest for MongoDB, using Helm templating syntax. It includes metadata for name and labels, and a spec for ports and selector. The ports section defines an external port and a target port, both using Helm template syntax to reference values. The selector section defines the app, component, and release labels.

```
1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: {{ .Release.Name }}-{{ .Chart.Name }}
6   labels:
7     app: reddit
8     component: mongo
9     release: {{ .Release.Name }}
10 spec:
11   ports:
12     - port: {{ .Values.service.externalPort }}
13       protocol: TCP
14       targetPort: {{ .Values.service.internalPort }}
15   selector:
16     app: reddit
17     component: mongo
18     release: {{ .Release.Name }}
```

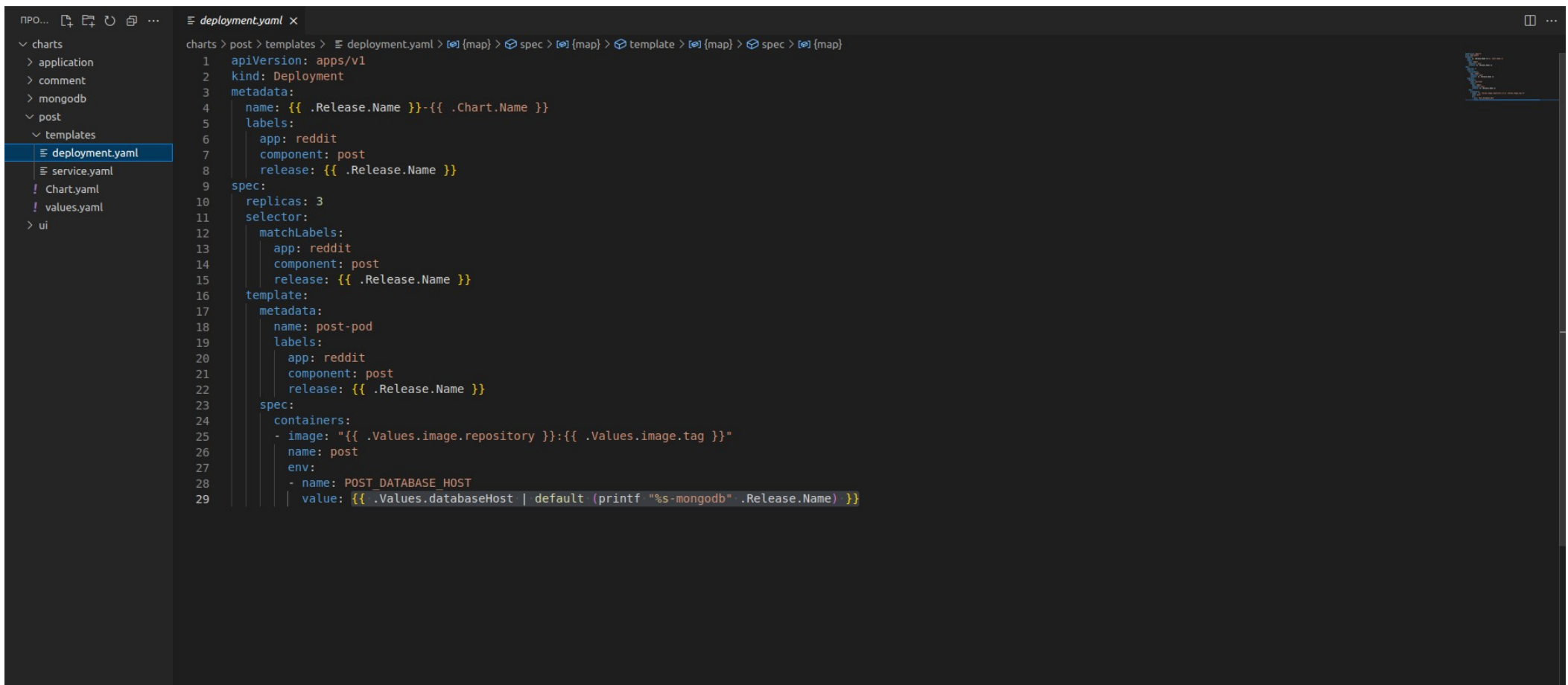
Создал Chart.yaml для сервиса post.



The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar displays a directory structure: 'charts' (expanded) contains 'application', 'comment', 'mongodb', 'post' (expanded), and 'ui'. Under 'post', there are 'templates', 'Chart.yaml' (selected and highlighted in blue), and 'values.yaml'. The main editor area shows the content of 'Chart.yaml' with line numbers 1 through 7. The text is as follows:

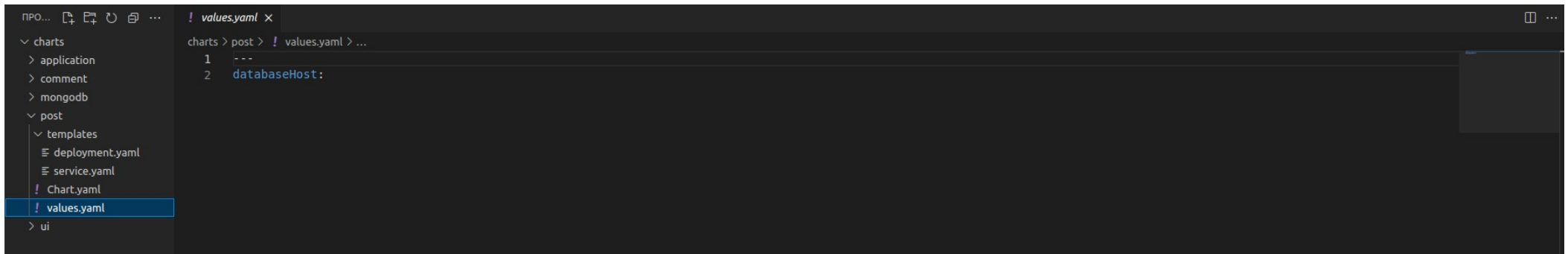
```
charts > post > ! Chart.yaml > # appVersion
Helm Chart.yaml - The Chart.yaml file is required for a chart. (chart.json)
1 name: post
2 version: 1.0.0
3 description: application Post
4 maintainers:
5   - name: me
6     email: me@me.me
7 appVersion: 1.0
```

Создал манифест deployment для сервиса post.
Переменные такие как Release.Name(имя релиза) и Chart.Name
(имя чарта) назначаются автоматически.
Переменные Values.service.internalPort, Values.image.repository,
Values.image.tag, будут определены в директории application.
Переменная Values.databaseHost будет определена в values.yaml
чарта post.



```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: {{ .Release.Name }}-{{ .Chart.Name }}
5    labels:
6      app: reddit
7      component: post
8      release: {{ .Release.Name }}
9  spec:
10   replicas: 3
11   selector:
12     matchLabels:
13       app: reddit
14       component: post
15       release: {{ .Release.Name }}
16   template:
17     metadata:
18       name: post-pod
19     labels:
20       app: reddit
21       component: post
22       release: {{ .Release.Name }}
23   spec:
24     containers:
25       - image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
26         name: post
27         env:
28           - name: POST DATABASE HOST
29             value: {{ .Values.databaseHost | default (printf "%s-mongodb" .Release.Name) }}
```

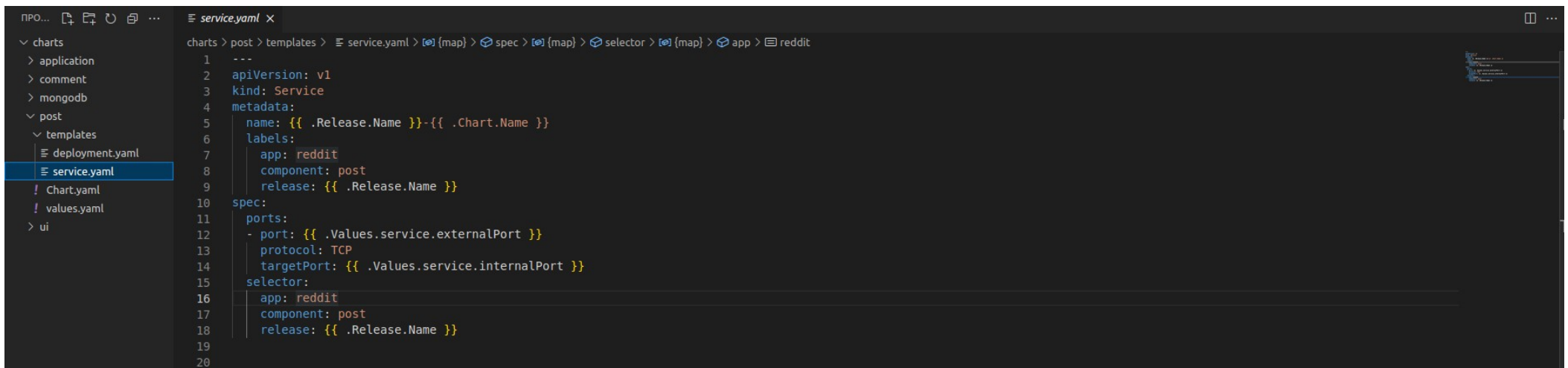
Создал values.yaml в нем определил единственную переменную Values.databaseHost



The screenshot shows a code editor with a sidebar on the left displaying a file tree. The tree is expanded to show the 'post' directory, which contains 'templates'. The 'values.yaml' file is selected and highlighted. The main editor area shows the content of 'values.yaml' with a single line: 'databaseHost:'. The breadcrumb at the top of the editor reads 'charts > post > ! values.yaml > ...'.

```
1 ---
2 databaseHost:
```

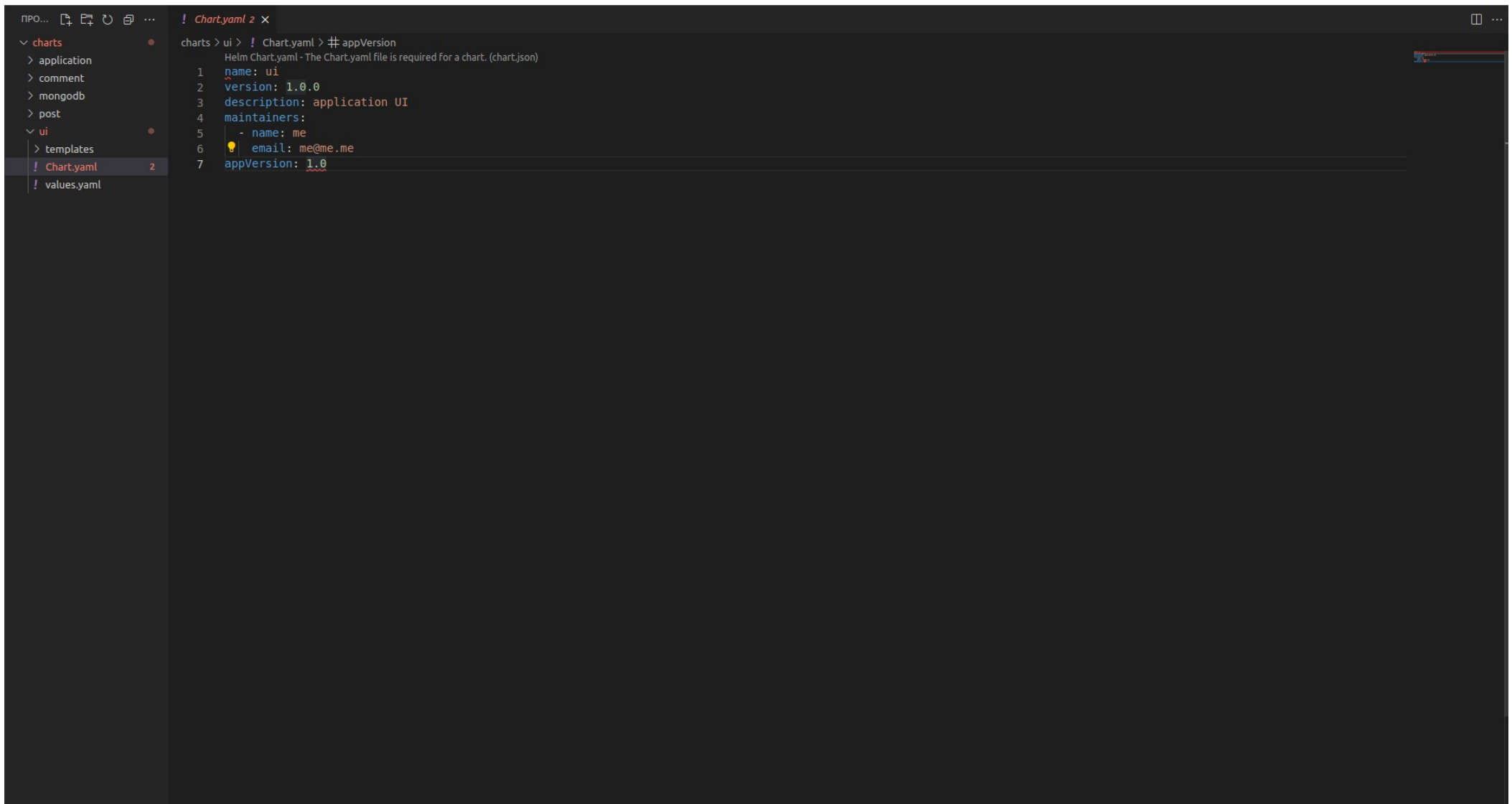
Создал манифест service для сервиса post.
Переменные Release.Name(имя релиза) и Chart.Name (имя чарта) назначаются автоматически.
Переменная Values.service.internalPort и Values.service.externalPort будут определены в директории application.



The screenshot shows a code editor with a sidebar on the left displaying a file tree. The tree is expanded to show the 'post' directory, which contains 'templates'. The 'service.yaml' file is selected and highlighted. The main editor area shows the content of 'service.yaml' with the following YAML structure: 'apiVersion: v1', 'kind: Service', 'metadata' with 'name' and 'labels', and 'spec' with 'ports' and 'selector'. The breadcrumb at the top of the editor reads 'charts > post > templates > service.yaml > [map] > spec > [map] > selector > [map] > app > reddit'.

```
1 ---
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: {{ .Release.Name }}-{{ .Chart.Name }}
6   labels:
7     app: reddit
8     component: post
9     release: {{ .Release.Name }}
10 spec:
11   ports:
12     - port: {{ .Values.service.externalPort }}
13       protocol: TCP
14       targetPort: {{ .Values.service.internalPort }}
15   selector:
16     app: reddit
17     component: post
18     release: {{ .Release.Name }}
```

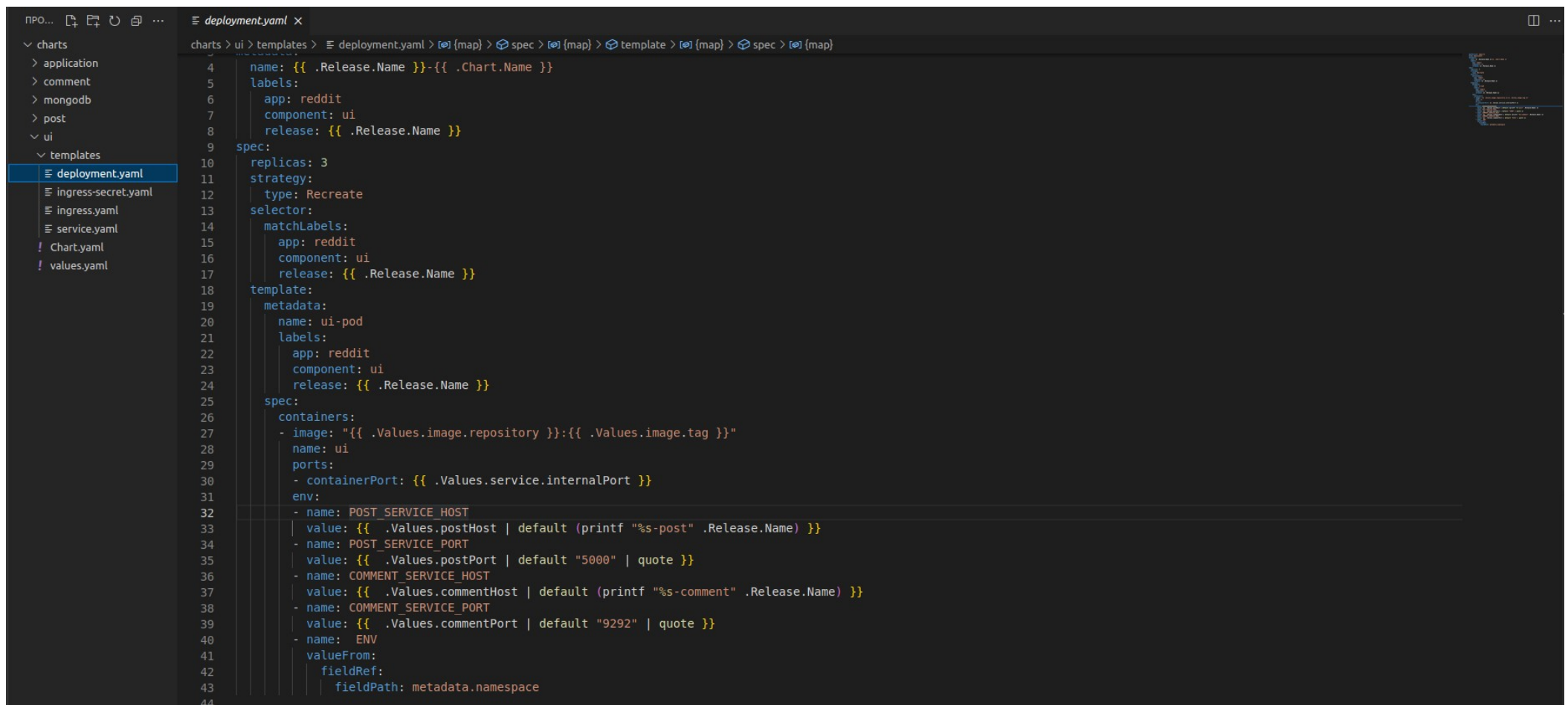
Создал Chart.yaml для сервиса ui



The screenshot shows a code editor with a dark theme. On the left, a file explorer shows a directory structure with 'charts' containing 'application', 'comment', 'mongodb', 'post', and 'ui'. The 'ui' directory contains 'templates', 'Chart.yaml', and 'values.yaml'. The 'Chart.yaml' file is selected and open in the main editor. The editor shows the following content:

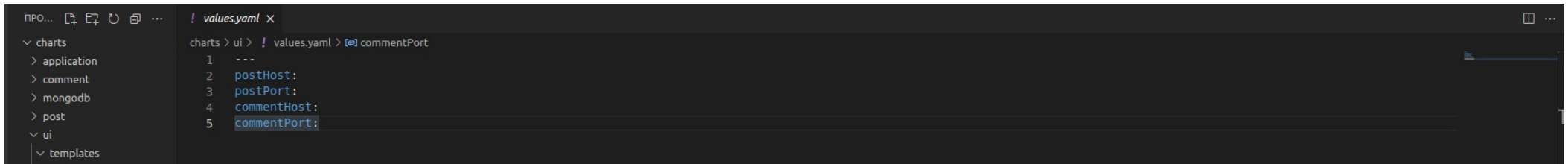
```
charts > ui > ! Chart.yaml > # appVersion
Helm Chart.yaml - The Chart.yaml file is required for a chart. (chart.json)
1  name: ui
2  version: 1.0.0
3  description: application UI
4  maintainers:
5    - name: me
6    email: me@me.me
7  appVersion: 1.0
```

Создал манифест deployment для сервиса ui. Переменные такие как Release.Name и Chart.Name назначаются автоматически. Переменные Values.service.internalPort, Values.image.repository, Values.image.tag, будут определены в директории application. Переменные Values.databaseHost, Values.postHost, Values.postPort, Values.commentHost, Values.commentPort будут определены в values чарта ui.

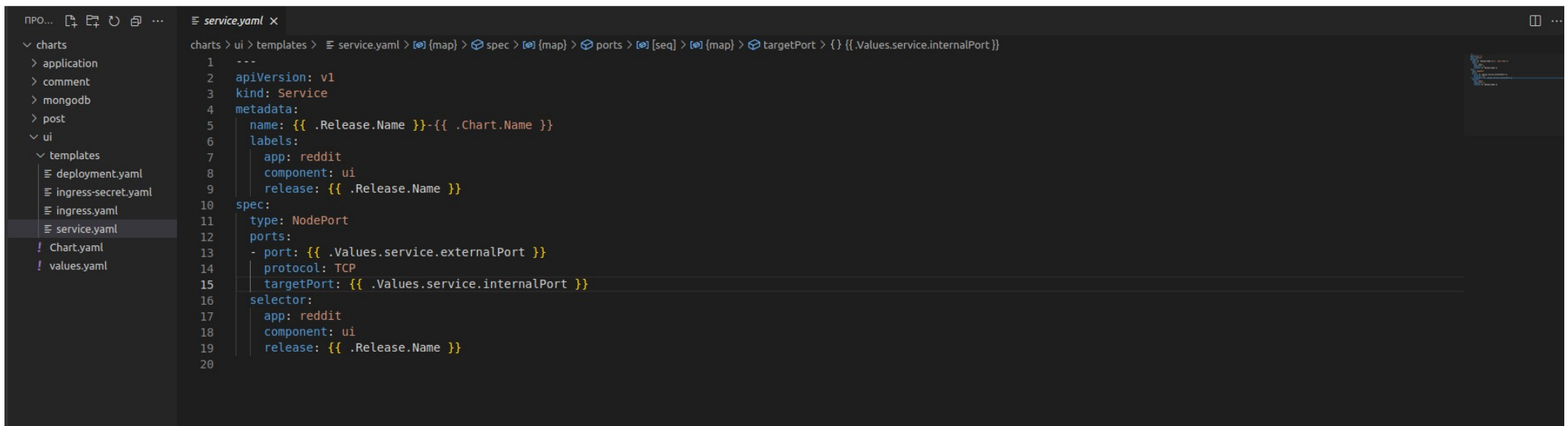


```
4  name: {{ .Release.Name }}-{{ .Chart.Name }}
5  labels:
6    app: reddit
7    component: ui
8    release: {{ .Release.Name }}
9  spec:
10   replicas: 3
11   strategy:
12     type: Recreate
13   selector:
14     matchLabels:
15       app: reddit
16       component: ui
17       release: {{ .Release.Name }}
18   template:
19     metadata:
20       name: ui-pod
21       labels:
22         app: reddit
23         component: ui
24         release: {{ .Release.Name }}
25     spec:
26       containers:
27       - image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
28         name: ui
29         ports:
30         - containerPort: {{ .Values.service.internalPort }}
31         env:
32         - name: POST_SERVICE_HOST
33           value: {{ .Values.postHost | default (printf "%s-post" .Release.Name) }}
34         - name: POST_SERVICE_PORT
35           value: {{ .Values.postPort | default "5000" | quote }}
36         - name: COMMENT_SERVICE_HOST
37           value: {{ .Values.commentHost | default (printf "%s-comment" .Release.Name) }}
38         - name: COMMENT_SERVICE_PORT
39           value: {{ .Values.commentPort | default "9292" | quote }}
40         - name: ENV
41           valueFrom:
42             fieldRef:
43               fieldPath: metadata.namespace
```

Создал values.yaml в нем определил переменные Values.databaseHost, Values.postHost, Values.postPort, Values.commentHost, Values.commentPort



Создал манифест service для сервиса ui.
Переменные Release.Name(имя релиза) и Chart.Name (имя чарта) назначаются автоматически.
Переменная Values.service.internalPort и Values.service.externalPort будут определены в директории application.



Создал манифест ingress. для организации единой точки входа в приложения снаружи, с работой как классический веб.

Переменные `Release.Name`(имя релиза) и `Chart.Name` (имя чарта) назначаются автоматически.

Переменная `Values.service.externalPort` будет определена в директории `application`.

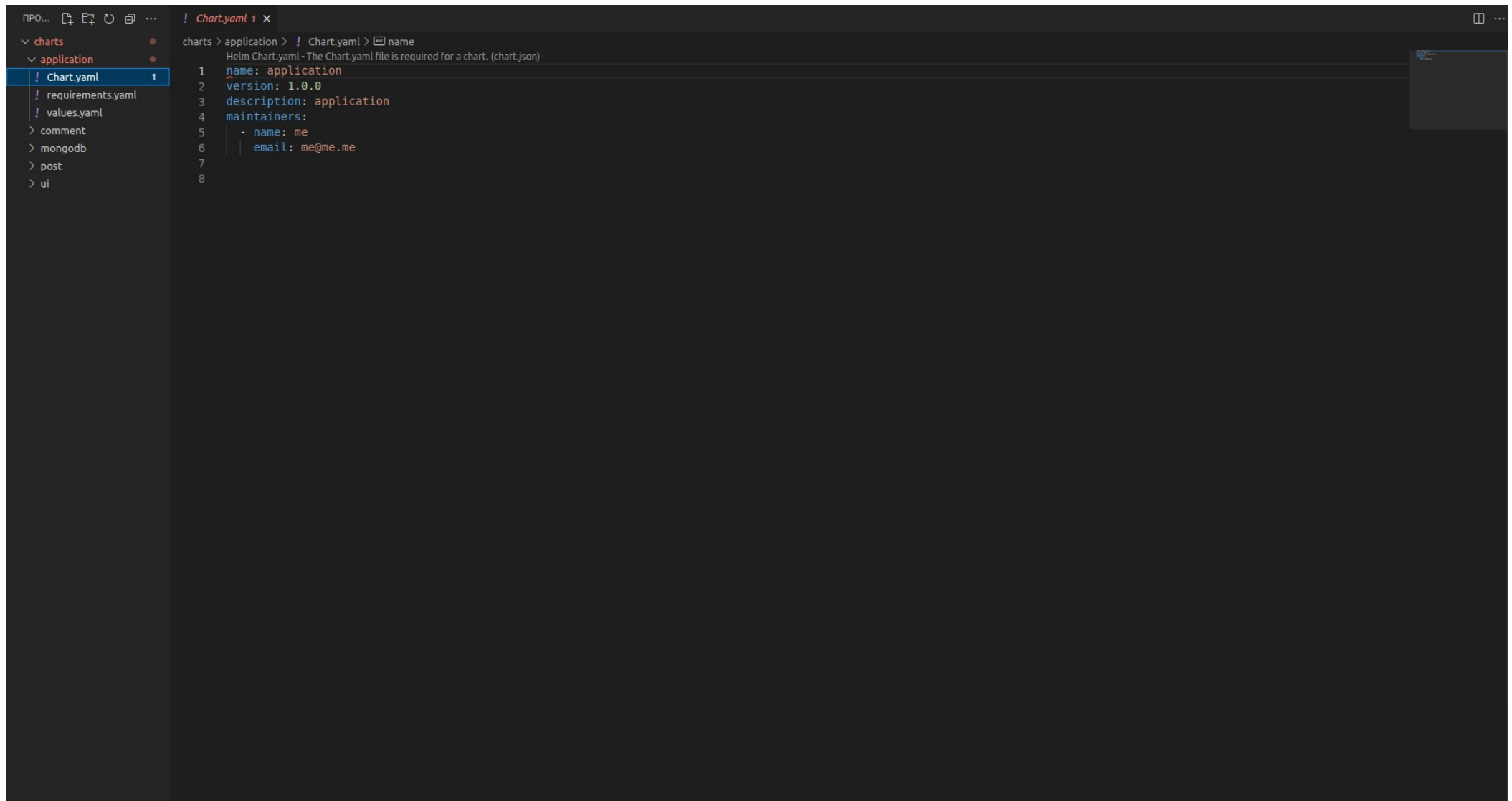
The screenshot shows a code editor with a sidebar on the left displaying a file tree. The tree is expanded to show the 'templates' directory under 'ui', with 'ingress.yaml' selected. The main editor area displays the content of 'ingress.yaml', which is a Kubernetes Ingress resource definition. The file is structured as follows:

```
1 ---
2 apiVersion: networking.k8s.io/v1
3 kind: Ingress
4 metadata:
5   name: {{ .Release.Name }}-{{ .Chart.Name }}
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: {{ .Release.Name }}-{{ .Chart.Name }}
20            port:
21              number: {{ .Values.service.externalPort }}
```

Создал манифест secret для ingress с самописным сертификатом

```
ingress-secret.yaml x
charts
  application
  comment
  mongodb
  post
  ui
    templates
      deployment.yaml
      ingress-secret.yaml x
        charts > templates > ingress-secret.yaml > [map] > type > [kubernetes.io/tls]
          1 apiVersion: v1
          2 data:
          3   |tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURFVENDQWZtZ0F3SUJBZ0lVVFZCTWNNMGJjOHBHRUFVGVct3TU1HMzRlTnBnd0RRWUpLb1pJaHZjTkFRRUwKQlFBd0dERVdNQLFHQTFVRUF3d050akl1T0RRdU1
          4   |tls.key: LS0tLS1CRUdJTiBQUklWQVRFIEtFWS0tLS0tCk1JSUV2QUlCQURBTk1Jna3Foa2l1OXMwQWcKFRRUZBQVNDQktZd2dnU2lBZ0VQVW9JQkFRFRpDSjh2ZG5zNXNmOHYKWHcxK1VFUFpjc3Q3RVJyaWRxRm9mNDDqdVhRSUF
          5 kind: Secret
          6 metadata:
          7   |name: ui-ingress
          8   |type: kubernetes.io/tls
```


В директории application с помощью механизма управления зависимостями создал единый Chart.yaml

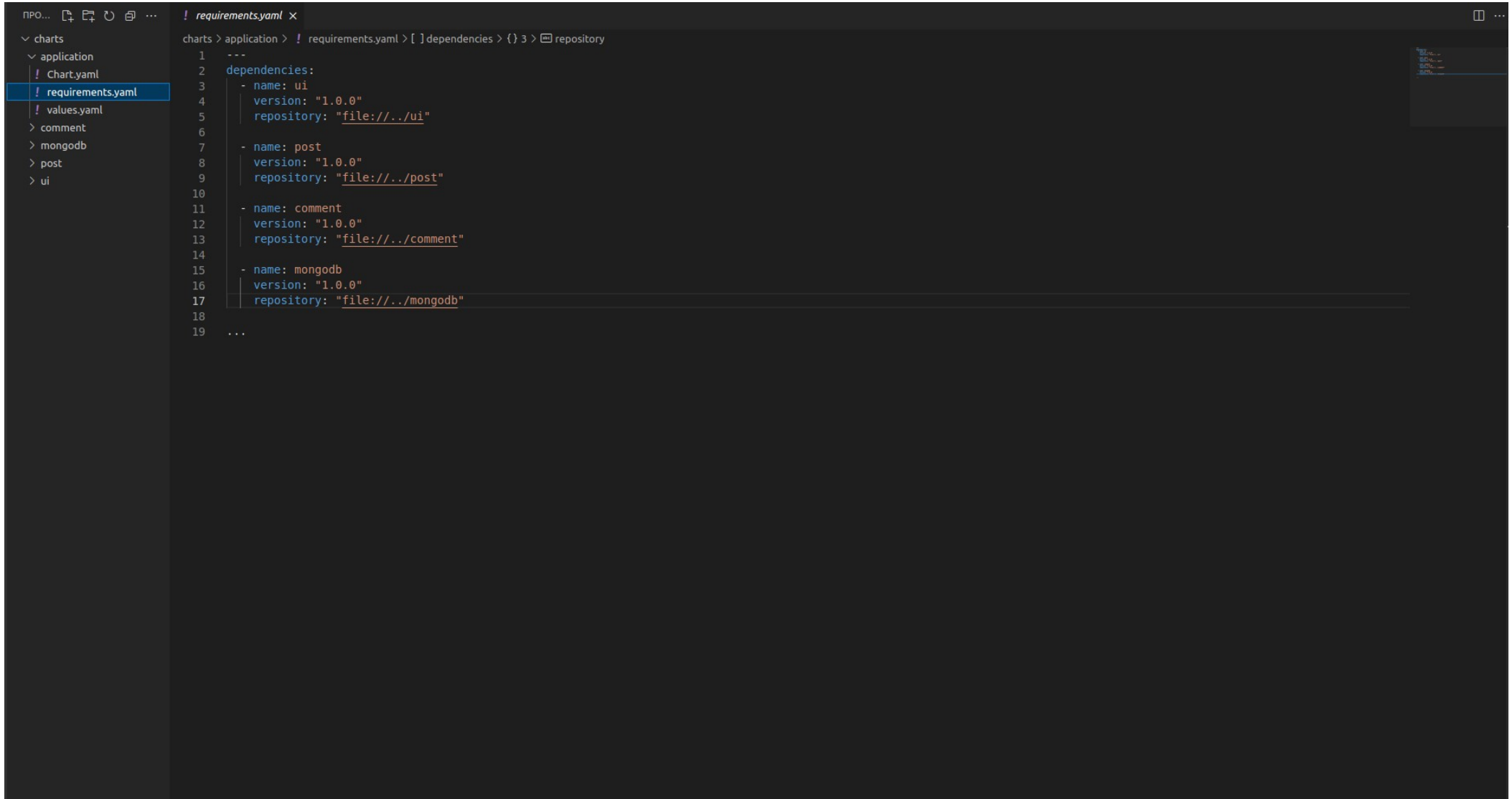


The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar displays a directory structure: 'charts' (expanded) contains 'application' (expanded), which contains 'Chart.yaml' (selected), 'requirements.yaml', and 'values.yaml'. Below these are sub-directories: 'comment', 'mongodb', 'post', and 'ui'. The main editor area shows the content of 'Chart.yaml' with line numbers 1 through 8. The file content is as follows:

```
1 name: application
2 version: 1.0.0
3 description: application
4 maintainers:
5   - name: me
6     email: me@me.me
7
8
```

At the top of the editor, a breadcrumb trail reads: 'charts > application > ! Chart.yaml > name'. A tooltip is visible above the 'name' part of the breadcrumb, displaying the text: 'Helm Chart.yaml - The Chart.yaml file is required for a chart. (chart.json)'.

Создал requirements.yaml где описал все зависимости, а именно чарты ui, post, comment, mongodb.



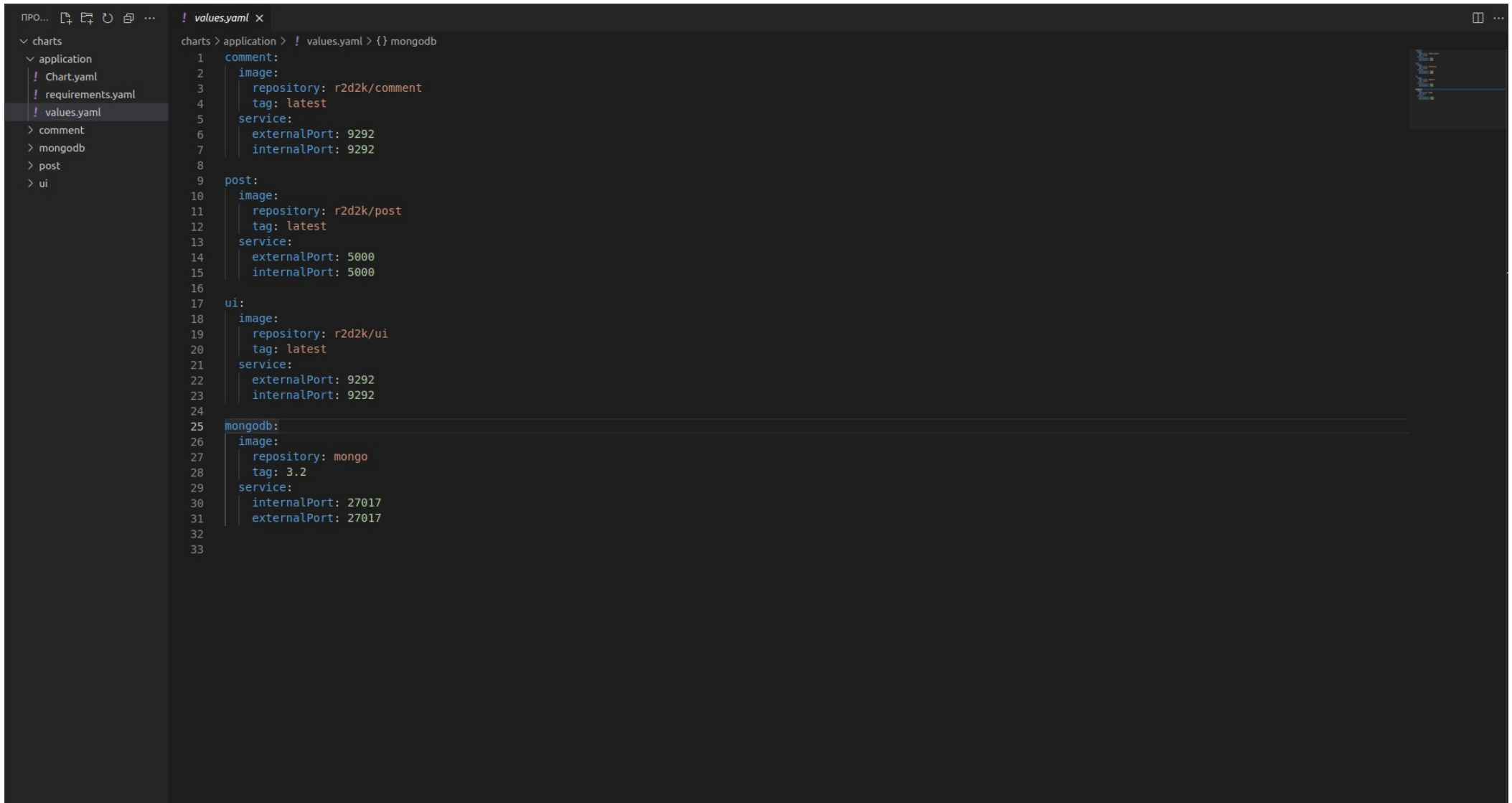
The screenshot shows a code editor with a sidebar on the left and a main editor area. The sidebar contains a tree view with the following structure:

- charts
 - application
 - Chart.yaml
 - requirements.yaml**
 - values.yaml
 - comment
 - mongodb
 - post
 - ui

The main editor area displays the content of the selected file, requirements.yaml. The text is as follows:

```
1 ---
2 dependencies:
3   - name: ui
4     version: "1.0.0"
5     repository: "file:///./ui"
6
7   - name: post
8     version: "1.0.0"
9     repository: "file:///./post"
10
11   - name: comment
12     version: "1.0.0"
13     repository: "file:///./comment"
14
15   - name: mongodb
16     version: "1.0.0"
17     repository: "file:///./mongodb"
18
19 ...
```

Создал values.yaml со всеми переменными из манифестов сервисов ui, post, comment, mongodb, кроме переменных определенных в values.yaml каждого чарта.



```
! values.yaml x
charts > application > ! values.yaml > {} mongodb
1  comment:
2    image:
3      repository: r2d2k/comment
4      tag: latest
5    service:
6      externalPort: 9292
7      internalPort: 9292
8
9  post:
10    image:
11      repository: r2d2k/post
12      tag: latest
13    service:
14      externalPort: 5000
15      internalPort: 5000
16
17  ui:
18    image:
19      repository: r2d2k/ui
20      tag: latest
21    service:
22      externalPort: 9292
23      internalPort: 9292
24
25  mongodb:
26    image:
27      repository: mongo
28      tag: 3.2
29    service:
30      internalPort: 27017
31      externalPort: 27017
32
33
```

Загрузил зависимости.

Появился файл requirements.lock с фиксацией зависимостей и была создана директория charts с зависимостями в виде архивов.

В итоге имеем вот такую структуру директории.

```
PROBLEMY  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ

charts
├── application
├── comment
├── mongodb
├── post
└── ui

alex@alex:~/myDoc/учеба/project/charts/application$ kubectl get pods
No resources found in default namespace.
alex@alex:~/myDoc/учеба/project/charts/application$ kubectl get deploy
No resources found in default namespace.
alex@alex:~/myDoc/учеба/project/charts/application$ kubectl get service
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.128.1   <none>        443/TCP    4h51m
alex@alex:~/myDoc/учеба/project/charts/application$ kubectl get ingress
No resources found in default namespace.
alex@alex:~/myDoc/учеба/project/charts/application$ helm dep update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "bitnami" chart repository
...Successfully got an update from the "google" chart repository
Update Complete. *Happy Helming!*
Saving 4 charts
Deleting outdated charts
alex@alex:~/myDoc/учеба/project/charts/application$ tree ../
../
├── application
│   ├── charts
│   │   ├── comment-1.0.0.tgz
│   │   ├── mongodb-1.0.0.tgz
│   │   ├── post-1.0.0.tgz
│   │   └── ui-1.0.0.tgz
│   ├── Chart.yaml
│   ├── requirements.lock
│   ├── requirements.yaml
│   └── values.yaml
├── comment
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   └── values.yaml
├── mongodb
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   ├── mongo-claim.yaml
│   │   ├── mongo-volume.yaml
│   │   └── service.yaml
│   └── values.yaml
├── post
│   ├── Chart.yaml
│   ├── templates
│   │   ├── deployment.yaml
│   │   └── service.yaml
│   └── values.yaml
└── ui
    ├── Chart.yaml
    ├── templates
    │   ├── deployment.yaml
    │   ├── ingress-secret.yaml
    │   ├── ingress.yaml
    │   └── service.yaml
    └── values.yaml

10 directories, 28 files
alex@alex:~/myDoc/учеба/project/charts/application$
```

Установил общий чарт из директории application.
По итогу вижу развернутое приложение в кластере kubernetes.
Получаю ip адрес присвоенный в ingress.

```
ПРОБЛЕМЫ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ

charts
├── application
├── comment
├── mongodb
├── post
└── ui

alex@alex:~/myDoc/учеба/project/charts$ helm install application ./application/
NAME: application
LAST DEPLOYED: Fri Jan 20 09:27:19 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None

alex@alex:~/myDoc/учеба/project/charts$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
application-comment-ffc95d468-8hkzx  1/1     Running   0           34s
application-comment-ffc95d468-k7jnz  1/1     Running   0           34s
application-comment-ffc95d468-r4csz  1/1     Running   0           34s
application-mongodb-84f95f6f4c-xncxg 1/1     Running   0           34s
application-post-65b54d798f-njhs2    1/1     Running   0           34s
application-post-65b54d798f-v52tg    1/1     Running   0           34s
application-post-65b54d798f-zfdqn    1/1     Running   0           34s
application-ui-65b8d5f946-5s9rz      1/1     Running   0           34s
application-ui-65b8d5f946-64nvw      1/1     Running   0           34s
application-ui-65b8d5f946-vqttj      1/1     Running   0           34s

alex@alex:~/myDoc/учеба/project/charts$ kubectl get deploy
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
application-comment                 3/3     3             3           40s
application-mongodb                 1/1     1             1           40s
application-post                     3/3     3             3           40s
application-ui                       3/3     3             3           40s

alex@alex:~/myDoc/учеба/project/charts$ kubectl get service
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
application-comment                 ClusterIP   10.96.189.199   <none>       9292/TCP          46s
application-mongodb                 ClusterIP   10.96.215.134   <none>       27017/TCP         46s
application-post                     ClusterIP   10.96.128.100   <none>       5000/TCP          46s
application-ui                       NodePort    10.96.168.223   <none>       9292:32564/TCP   46s
kubernetes                          ClusterIP   10.96.128.1     <none>       443/TCP           19h

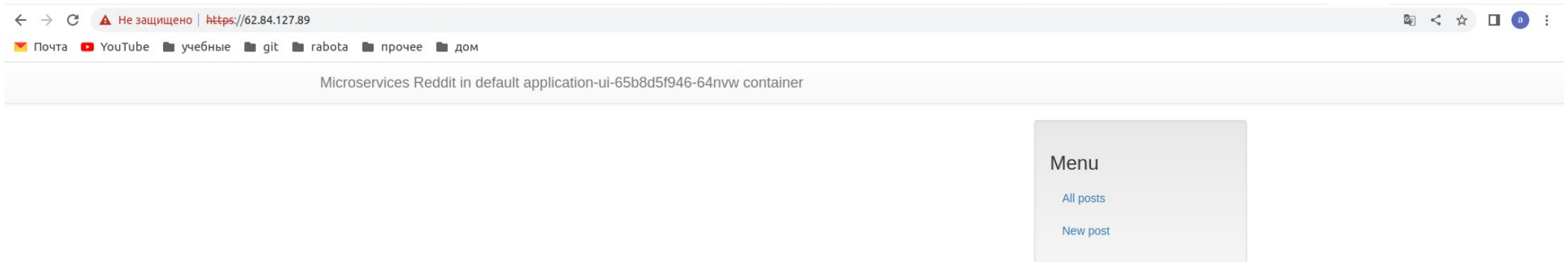
alex@alex:~/myDoc/учеба/project/charts$ kubectl get ingre
error: the server doesn't have a resource type "ingres"

alex@alex:~/myDoc/учеба/project/charts$ kubectl get ingress
NAME                                CLASS  HOSTS          ADDRESS      PORTS    AGE
application-ui                      nginx  *              62.84.127.89 80, 443  60s

alex@alex:~/myDoc/учеба/project/charts$ helm ls
NAME      NAMESPACE    REVISION    UPDATED              STATUS      CHART          APP VERSION
application  default      1           2023-01-20 09:27:19.403822268 +0300 MSK  deployed   application-1.0.0

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

В браузере по ip адресу присвоенному в ingress вижу развернутое приложение.



Теперь можно писать новую версию приложения и деплоить её под другим названием helm.