

Урок 5. Сетевые абстракции Kubernetes

Разверните в кластере сервер базы данных PostgreSQL. Из предыдущего задания.

Добавьте к нему service с портом 5432 и именем database.

В этом же неймспэйсе создайте deployment с образом redmine:4.1.1

Для запуска нужно передать переменные окружения:

REDMINE_DB_POSTGRES = database REDMINE_DB_USERNAME = <postgres_user>

REDMINE_DB_PASSWORD = <postgres_password> (значение должно браться из секрета)

REDMINE_DB_DATABASE = <postgres_database>

REDMINE_SECRET_KEY_BASE = supersecretkey (значение должно браться из секрета)

Обратите внимание что имя пользователя, пароль и база данных должны соответствовать значениям которые указаны в переменных окружения деплоймента postgresql

В деплойменте приложения должен быть описан порт 3000

Создайте service для приложения с портом 3000

Создайте ingress для приложения, так чтобы запросы с любым доменом на белый IP вашего сервиса nginx-ingress-controller (тот что в нэймспэйсе ingress-nginx с типом LoadBalancer) шли на приложение

Проверьте что при обращении из браузера на белый IP вы видите открывшееся приложение Redmine (<https://www.redmine.org/>)

minikube start

```
Терминал - igor@ubuntu-server: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ minikube start
👉 minikube v1.23.2 on Ubuntu 20.04 (vbox/amd64)
👉 minikube 1.24.0 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.24.0
💡 To disable this notice, run: 'minikube config set WantUpdateNotification false'

🌟 Using the docker driver based on existing profile
👉 Starting control plane node minikube in cluster minikube
👉 Pulling base image ...
👉 Restarting existing docker container for "minikube" ...
👉 Preparing Kubernetes v1.22.2 on Docker 20.10.8 ...
👉 Verifying Kubernetes components...
   ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
👉 Enabled addons: default-storageclass, storage-provisioner
👉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
igor@ubuntu-server:~$
```

minikube status

```
Терминал - igor@ubuntu-server: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
igor@ubuntu-server:~$
```

kubectl cluster-info

```
Терминал - igor@ubuntu-server: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
igor@ubuntu-server:~$
```

kubectl create ns redmine

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ vi conf_map_pv.yaml
tolstikov@my3vm:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
tolstikov@my3vm:~$ kubectl create ns redmine
namespace/redmine created
tolstikov@my3vm:~$
```

kubectl config set-context --current --namespace=redmine

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
tolstikov@my3vm:~$ kubectl create ns redmine
namespace/redmine created
tolstikov@my3vm:~$ kubectl config set-context --current --namespace=redmine
Context "minikube" modified.
tolstikov@my3vm:~$
```

vi pvc.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pg-storage
  namespace: redmine
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: "standard"
:wq
```

kubectl apply -f pvc.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ vi pvc.yaml
tolstikov@my3vm:~$ kubectl apply -f pvc.yaml
persistentvolumeclaim/pg-storage created
tolstikov@my3vm:~$
```

kubectl get pvc

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pg-storage    Bound    pvc-9d7bee6d-d751-44aa-acce-36457f24be10  10Gi       RWX             standard       40s
tolstikov@my3vm:~$
```

kubectl create secret generic pg-secret --from-literal=PASS=rmdbpassword

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

pg-storage    Bound    pvc-9d7bee6d-d751-44aa-acce-36457f24be10  10Gi       RWX             standard       40s
tolstikov@my3vm:~$ kubectl create secret generic pg-secret --from-literal=PASS=rmdbpassword
secret/pg-secret created
tolstikov@my3vm:~$
```

vi deployment-pg.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: pg-db
  namespace: redmine
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pg-db
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: pg-db
    spec:
      initContainers:
        - image: busybox
          name: mount-permissions-fix
          command: ["sh", "-c", "chmod 777 /var/lib/postgresql/data"]
          volumeMounts:
            - name: data
              mountPath: /var/lib/postgresql/data
      containers:
        - image: postgres:10.13
          name: postgres
          env:
            - name: POSTGRES_USER
              value: "rm_db_user"
            - name: POSTGRES_DB
              value: "rm_db_database"
            - name: PGDATA
              value: "/var/lib/postgresql/data/pgdata"
            - name: POSTGRES_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: pg-secret
                  key: PASS
          ports:
            - containerPort: 5432
              protocol: TCP
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
            limits:
              cpu: 100m
              memory: 100Mi
          volumeMounts:
            - name: data
              mountPath: /var/lib/postgresql/data
      volumes:
        - name: data
          persistentVolumeClaim:
            claimName: pg-storage
:wq
```

kubectl apply -f deployment-pg.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl get pvc
NAME          STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
pg-storage    Bound   pvc-9d7bee6d-d751-44aa-acce-36457f24be10  10Gi      RWX           standard      40s
tolstikov@my3vm:~$ kubectl create secret generic pg-secret --from-literal=PASS=rmdbpassword
secret/pg-secret created
tolstikov@my3vm:~$ vi deployment-pg.yaml
tolstikov@my3vm:~$ kubectl apply -f deployment-pg.yaml
deployment.apps/pg-db created
tolstikov@my3vm:~$
```

vi service-pg.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: v1
kind: Service
metadata:
  name: pg-service
  namespace: redmine
spec:
  ports:
  - port: 5432
    targetPort: 5432
  selector:
    app: pg-db
  type: ClusterIP
~
:wq
```

kubectl apply -f service-pg.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ vi service-pg.yaml
tolstikov@my3vm:~$ kubectl apply -f service-pg.yaml
service/pg-service created
tolstikov@my3vm:~$
```

kubectl create secret generic redmine-secret --from-literal=KEY=supersecretkey

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl apply -f service-pg.yaml
service/pg-service created
tolstikov@my3vm:~$ kubectl create secret generic redmine-secret --from-literal=KEY=supersecretkey
secret/redmine-secret created
tolstikov@my3vm:~$
```

vi deployment-redmine.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redmine-app
  namespace: redmine
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redmine-app
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: redmine-app
    spec:
      containers:
      - image: redmine:4.1.1
        name: redmine
        env:
          - name: REDMINE_DB_POSTGRES
            value: "pg-service"
          - name: REDMINE_DB_USERNAME
            value: "rm_db_user"
          - name: REDMINE_DB_DATABASE
            value: "rm_db_database"
          - name: REDMINE_DB_PASSWORD
            valueFrom:
              secretKeyRef:
                name: pg-secret
                key: PASS
          - name: REDMINE_SECRET_KEY_BASE
            valueFrom:
              secretKeyRef:
                name: redmine-secret
                key: KEY
        ports:
          - containerPort: 3000

:wq
```

kubectl apply -f deployment-redmine.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

secret/redmine-secret created
tolstikov@my3vm:~$ vi deployment-redmine.yaml
tolstikov@my3vm:~$ kubectl apply -f deployment-redmine.yaml
deployment.apps/redmine-app created
tolstikov@my3vm:~$
```

vi service-redmine.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: v1
kind: Service
metadata:
  name: redmine-service
  namespace: redmine
spec:
  ports:
  - port: 80
    targetPort: 3000
  selector:
    app: redmine-app
  type: ClusterIP
:wd
```

kubectl apply -f service-redmine.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl apply -f deployment-redmine.yaml
deployment.apps/redmine-app created
tolstikov@my3vm:~$ vi service-redmine.yaml
tolstikov@my3vm:~$ kubectl apply -f service-redmine.yaml
service/redmine-service created
tolstikov@my3vm:~$
```

vi ingress.yaml

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: redmine-ingress
  namespace: redmine
spec:
  rules:
  - http:
      paths:
      - backend:
          serviceName: redmine-service
          servicePort: 80
~
13,10 All
```

kubectl apply -f ingress.yaml

error: unable to recognize "ingress.yaml": no matches for kind "Ingress" in version "extensions/v1beta1"

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ vi ingress.yaml
tolstikov@my3vm:~$ kubectl apply -f ingress.yaml
error: unable to recognize "ingress.yaml": no matches for kind "Ingress" in version "extensions/v1beta1"
tolstikov@my3vm:~$
```

kubectl get pod


```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl get pod
NAME                                READY  STATUS   RESTARTS  AGE
pg-db-56dd9b6975-5k8wl             1/1    Running  0          38m
redmine-app-58fc94cf66-xww6t       1/1    Running  0          15m
tolstikov@my3vm:~$
```

kubectl get svc -A

```
Терминал - tolstikov@my3vm: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка

tolstikov@my3vm:~$ kubectl get svc -A
NAMESPACE   NAME                                TYPE              CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
default     balanced                           LoadBalancer    10.96.220.97    <pending>        8080:32562/TCP   32h
default     hello-minikube                     NodePort         10.111.213.86   <none>           8080:32682/TCP   5d2h
default     kubernetes                         ClusterIP        10.96.0.1       <none>           443/TCP          5d2h
default     my-iservice                        ClusterIP        10.107.178.92   <none>           8080/TCP         34h
ingress-nginx ingress-nginx-controller            NodePort         10.106.66.228   <none>           80:32762/TCP,443:31038/TCP 34h
ingress-nginx ingress-nginx-controller-admission ClusterIP        10.111.15.133   <none>           443/TCP          34h
kube-system kube-dns                           ClusterIP        10.96.0.10      <none>           53/UDP,53/TCP,9153/TCP 5d2h
pg          my-iservice                        ClusterIP        10.107.75.36    <none>           8080/TCP         173m
redmine     pg-service                         ClusterIP        10.103.134.50   <none>           5432/TCP         33m
redmine     redmine-service                    ClusterIP        10.102.251.137  <none>           80/TCP          10m
tolstikov@my3vm:~$
```

kubectl get services

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

Normal Pulled 17m (x4 over 19m) kubelet Container image "redmine:4.1.1" already present on machine
Normal Started 17m (x5 over 19m) kubelet Started container redmine
Warning BackOff 4m26s (x68 over 19m) kubelet Back-off restarting failed container

igor@ubuntu-server:~$ kubectl get services
NAME            TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
pg-service      ClusterIP 10.109.52.102    <none>           5432/TCP         33m
redmine-service ClusterIP 10.100.187.140   <none>           80/TCP          20m
igor@ubuntu-server:~$
```

kubectl get services redmine-service

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

NAME            TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
pg-service      ClusterIP 10.109.52.102    <none>           5432/TCP         33m
redmine-service ClusterIP 10.100.187.140   <none>           80/TCP          20m
igor@ubuntu-server:~$ kubectl describe service ^C
igor@ubuntu-server:~$ kubectl get services redmine-service
NAME            TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
redmine-service ClusterIP 10.100.187.140   <none>           80/TCP          23m
igor@ubuntu-server:~$
```

minikube addons enable ingress

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

redmine-service ClusterIP 10.100.187.140 <none> 80/TCP 23m
igor@ubuntu-server:~$ minikube addons enable ingress
  ■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
  ■ Using image k8s.gcr.io/ingress-nginx/controller:v1.0.0-beta.3
  ■ Using image k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.0
Verifying ingress addon...
🌟 The 'ingress' addon is enabled
igor@ubuntu-server:~$ kubectl get pods
```