

1. Установили утилиты minikube и kubectl. Также установили docker для работоспособности minikube.

2.

```
user@user:~$ minikube start --driver=docker --memory=2G --nodes=2
user@user:~$ minikube addons enable registry
user@user:~$ kubectl -n kube-system expose deployment/registry --type=ClusterIP
--port=5000 --target-port=5000 --name=local-registry --selector='actual-registry
=true'
service/local-registry exposed
user@user:~$ export REGISTRY_IP=$(kubectl -n kube-system get svc/local-registry
-o=template={{.spec.clusterIP}})
user@user:~$ minikube ssh "echo '$REGISTRY_IP local-registry.kube-system.svc.clu
ster.local' | sudo tee -a /etc/hosts"
user@user:~$ echo "127.0.0.1 local-registry.kube-system.svc.cluster.local" | sud
o tee -a /etc/hosts
user@user:~$ kubectl port-forward --namespace kube-system service/local-registry
--address=127.0.0.1 5000 &
```

3. maria.yaml

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mariadb
  labels:
    app: mariadb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mariadb
  serviceName: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      containers:
        - name: mariadb
          image: mariadb:latest
          imagePullPolicy: "IfNotPresent"
          env:
            - name: MARIADB_ROOT_PASSWORD
              value: "123qwe"
            - name: MARIADB_DATABASE
              value: "library"
            - name: MARIADB_USER
              value: "n_user"
            - name: MARIADB_PASSWORD
              value: "password"
          ports:
            - name: mysql
              containerPort: 3306
          volumeMounts:
            - name: data
              mountPath: /var/lib/mysql
      volumeClaimTemplates:
        - metadata:
            name: data
            labels:
              app: mariadb
          spec:
            accessModes:
              - "ReadWriteOnce"
            resources:
              requests:
                storage: "1Gi"
```

maria-ser.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: mariadb
  labels:
    app: mariadb
spec:
  type: ClusterIP
  ports:
    - protocol: TCP
      port: 3306
      targetPort: 3306
  selector:
    app: mariadb

```

```

user@user:~$ kubectl apply -f maria-ser.yaml
service/mariadb created

```

```

user@user:~$ kubectl apply -f maria.yaml
statefulset.apps/mariadb configured

```

init.sql

```

create database if not exists library;
use library;

create table if not exists books (b_id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(100) NOT NULL);

insert into books values(1,'name1');
insert into books values(2,'name2');
insert into books values(3,'name3');
insert into books values(4,'name4');

create table if not exists orders (o_id INT PRIMARY KEY AUTO_INCREMENT, b_id INT REFERENCES books(b_id), address VARCHAR(100) NOT NULL, status BOOLEAN);

create user 'n_user'@'%' identified by 'password';
grant all privileges on *.* to 'n_user'@'%';
flush privileges;

```

```

user@user:~$ kubectl exec -it mariadb-0 -- mariadb -u root -p123qwe < init.sql

```

4. Dockerfile

```

FROM php:8.1-fpm-alpine3.16

COPY ./lab4/add_main.php /var/www/html/
COPY ./lab4/show_main.php /var/www/html/

RUN docker-php-ext-install -j$(nproc) mysqli pdo_mysql

CMD ["php-fpm"]

```

```

user@user:~$ docker build -t local-registry.kube-system.svc.cluster.local:5000/php-fpm:latest .

```

```

user@user:~$ docker push local-registry.kube-system.svc.cluster.local:5000/php-fpm:latest

```

php-dep.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-fpm
  labels:
    app: php-fpm
spec:
  replicas: 1
  selector:
    matchLabels:
      app: php-fpm
  template:
    metadata:
      labels:
        app: php-fpm
    spec:
      containers:
        - name: php-fpm
          image: local-registry.kube-system.svc.cluster.local:5000/php-fpm:latest
          ports:
            - containerPort: 9000
```

php-ser.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: php-fpm
  namespace: "default"
  labels:
    app: php-fpm
spec:
  type: ClusterIP
  ports:
    - name: php-fpm
      port: 9000
      protocol: TCP
      targetPort: 9000
  selector:
    app: php-fpm
```

```
user@user:~$ kubectl apply -f php-ser.yaml
```

```
user@user:~$ kubectl apply -f php-dep.yaml
```

5. nginx-conf.yaml

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-config
data:
  nginx.conf: |
    events{
    }
    http {
      server {
        listen 80 default_server;
        listen [::]:80 default_server;
        root /var/www/html;
        server_name rgz.local;
        access_log /dev/stdout;
        error_log /dev/stderr debug;
        location / {
          try_files $uri $uri/ =404;
        }
        location ~ /\.php$ {
          include fastcgi_params;
          fastcgi_param REQUEST_METHOD $request_method;
          fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
          fastcgi_pass php-fpm:9000;
        }
      }
    }
  }
```

nginx-dep.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      volumes:
        - name: nginx-config-volume
          configMap:
            name: nginx-config
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-config-volume
              mountPath: /etc/nginx/nginx.conf
              subPath: nginx.conf
```

nginx-ser.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30000
  selector:
    app: nginx

```

```

user@user:~$ kubectl apply -f nginx-conf.yaml
configmap/nginx-config created
user@user:~$ kubectl apply -f nginx-dep.yaml
deployment.apps/nginx created
user@user:~$ kubectl apply -f nginx-ser.yaml
service/nginx created

```

```

user@user:~$ kubectl get service
NAME          TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)          AGE
kubernetes    ClusterIP   10.96.0.1     <none>       443/TCP          21h
mariadb       ClusterIP   10.97.145.239 <none>       3306/TCP         52m
nginx         NodePort    10.108.229.119 <none>       80:30000/TCP     19h
php-fpm       ClusterIP   10.102.52.92  <none>       9000/TCP         19h
user@user:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx         1/1     1             1           18h
php-fpm       1/1     1             1           18h
user@user:~$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
mariadb-0                           1/1     Running   0          31m
nginx-6bb8659b44-hjlrq              1/1     Running   0          25m
php-fpm-bc7769f55-xwfjf             1/1     Running   0          47m
user@user:~$

```

```

user@user:~$ minikube ip
192.168.49.2

```

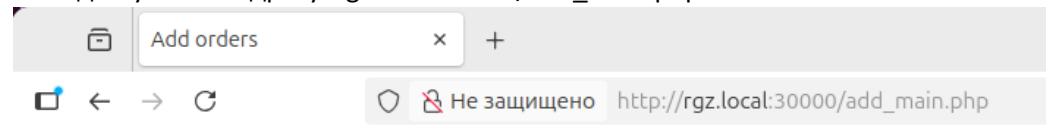
/etc/hosts

```

127.0.0.1 local-registry.kube-system.svc.cluster.local
192.168.49.2 rgz.local

```

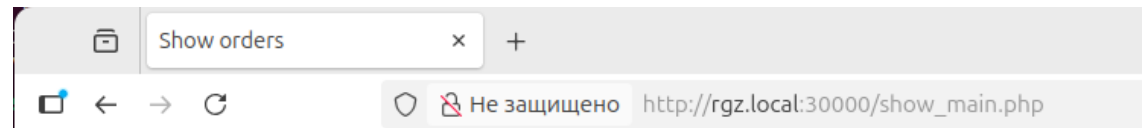
Сайт доступен по адресу: rgz.local:30000/add_main.php



Book name:

name1 ▾
add1
Add order

[Show status](#)



Orders:

#	Address	Status
1	add1	0

[Add orders](#)