

Домашнее задание для к уроку 4 - Хранение данных и ресурсы

Напишите deployment для запуска сервера базы данных PostgreSQL.

Приложение должно запускаться из образа postgres:10.13

Должен быть описан порт:

5432 TCP

В деплойменте должна быть одна реплика, при этом при обновлении образа НЕ ДОЛЖНО одновременно работать несколько реплик. (то есть сначала должна удаляться старая реплика и только после этого подниматься новая).

Это можно сделать или с помощью maxSurge/maxUnavailable или указав стратегию деплоя Recreate.

В базе данных при запуске должен автоматически создаваться пользователь testuser с паролем testpassword. А также база testdatabase.

Для этого нужно указать переменные окружения POSTGRES_PASSWORD, POSTGRES_USER, POSTGRES_DB в деплойменте. При этом значение переменной POSTGRES_PASSWORD должно браться из секрета.

Так же нужно указать переменную PGDATA со значением /var/lib/postgresql/data/pgdata См. документацию к образу https://hub.docker.com/_/postgres раздел PGDATA

База данных должна хранить данные в PVC с размером диска в 10Gi, замонтированном в pod по пути /var/lib/postgresql/data

Проверка

Для проверки работоспособности базы данных:

Узнайте IP пода postgresql

```
kubectl get pod -o wide
```

Запустите рядом тестовый под

```
kubectl run -t -i --rm --image postgres:10.13 test bash
```

Внутри тестового пода выполните команду для подключения к БД

```
psql -h <postgresql pod IP из п.1> -U testuser testdatabase
```

Введите пароль - testpassword

Все в том же тестовом поде, после подключения к инстансу БД выполните команду для создания таблицы

```
CREATE TABLE testtable (testcolumn VARCHAR (50) );
```

Проверьте что таблица создалась. Для этого все в том же тестовом поде выполните команду

```
\dt
```

Выйдите из тестового пода. Попробуйте удалить под с postgresql.

После его пересоздания повторите все с п.1, кроме п.4 Проверьте что созданная ранее таблица никуда не делась.

1) Создан файл deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: post-depl

spec:

replicas: 1

selector:

matchLabels:

app: post-depl

strategy:

rollingUpdate:

maxSurge: 0

maxUnavailable: 1

type: RollingUpdate

template:

metadata:

labels:

app: post-depl

spec:

containers:

- image: postgres:10.13

name: postgres

env:

- name: POSTGRES_DB

value: testdatabase

- name: PGDATA

value: /var/lib/postgresql/data/pgdata

- name: POSTGRES_USER

value: testuser

- name: POSTGRES_PASSWORD

valueFrom:

secretKeyRef:

name: my-secret

key: db-pass

ports:

- containerPort: 5432

protocol: TCP

volumeMounts:

- name: data

mountPath: /var/lib/postgresql/data

volumes:

- name: data

persistentVolumeClaim:

claimName: post-depl

2) Создан файл pvc.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: post-depl

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 10Gi

3) Выполнено

export KUBECONFIG=kubernetes-cluster-6022_kubeconfig.yaml

kubectl create secret generic my-secret --from-literal db-pass='testpassword'

kubectl get secret my-secret

kubectl get secret my-secret -o yaml

kubectl apply -f pvc.yaml

kubectl apply -f deployment.yaml

kubectl get pod -o wide

kubectl run -t -i --rm --image postgres:10.13 test bash

в базе

psql -h 10.100.135.82 -U testuser testdatabase

\dt

```
[root@centos7-6 postgres]# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
post-depl-779dbf6cfb-khmhr         1/1     Running   0           2m15s  10.100.135.80   kubernetes-cluster-6022-default-gr
oup-1                               <none>   <none>
[root@centos7-6 postgres]# kubectl run -t -i --rm --image postgres:10.13 test bash
If you don't see a command prompt, try pressing enter.
root@test:/# psql -h 10.100.135.80 -U testuser testdatabase
Password for user testuser:
psql (10.13 (Debian 10.13-1.pgdg90+1))
Type "help" for help.

testdatabase=# CREATE TABLE testtable (testcolumn VARCHAR(50));
CREATE TABLE
testdatabase=# \dt
               List of relations
Schema |   Name   | Type  | Owner
-----+-----+-----+-----
public | testtable | table | testuser
(1 row)
```

Под удален, все пересоздано

```

[root@centos7-6 postgres]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
post-depl-779dbf6cfb-c2mcb         1/1     Running   0           2m48s
[root@centos7-6 postgres]# kubectl delete pod post-depl-779dbf6cfb-c2mcb
pod "post-depl-779dbf6cfb-c2mcb" deleted
[root@centos7-6 postgres]# kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
post-depl-779dbf6cfb-46zhm         0/1     ContainerCreating   0           33s
[root@centos7-6 postgres]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
post-depl-779dbf6cfb-46zhm         1/1     Running   0           63s
[root@centos7-6 postgres]# kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
post-depl-779dbf6cfb-46zhm         1/1     Running   0           70s   10.100.135.82   kubernetes-cluster-6022-default-grou
p-1   <none>               <none>
[root@centos7-6 postgres]# kubectl run -t -i --rm --image postgres:10.13 test bash
If you don't see a command prompt, try pressing enter.
root@test:/# psql -h 10.100.135.82 -U testuser testdatabase
Password for user testuser:
psql (10.13 (Debian 10.13-1.pgdg90+1))
Type "help" for help.

testdatabase=# \dt
               List of relations
 Schema |   Name   | Type | Owner
-----+-----+-----+-----
 public | testtable | table | testuser
(1 row)

```

База сохранилась.