Урок 4. Хранение данных и ресурсы

Практическое задание

Студент Виталий Зайцев, DE\_622

Contents

[Домашнее задание 2](#_Toc83041596)

[Проверка работоспособности/готовности kubernetes 4](#_Toc83041597)

[Подготовка манифестов 5](#_Toc83041598)

[Создание secret.yml 5](#_Toc83041599)

[secret-lesson-04.yml 6](#_Toc83041600)

[НЕ УДАЛОСЬ СКОНФИГУРИРОВАТЬ!!! Создание PersistentVolume.yml 7](#_Toc83041601)

[pgdata-pvc-disk-lesson-04.yml 7](#_Toc83041602)

[Изменения в deployment.yml 7](#_Toc83041603)

[Создание deployment.yml 9](#_Toc83041604)

[deployment-lesson-04.yml 9](#_Toc83041605)

[Применение манифестов 11](#_Toc83041606)

[Cоздание namespace my-postgres 11](#_Toc83041607)

[Установка контекста в namespace 11](#_Toc83041608)

[Применение манифестов 11](#_Toc83041609)

[Проверка работосопосбности 13](#_Toc83041610)

[1.Узнайте IP пода postgresql 13](#_Toc83041611)

[2.Запустите рядом тестовый под 13](#_Toc83041612)

[3.Внутри тестового пода выполните команду для подключения к БД 13](#_Toc83041613)

[4.Все в том же тестовом поде, после подключения к инстансу БД выполните команду для создания таблицы 14](#_Toc83041614)

[5.Проверьте что таблица создалась. Для этого все в том же тестовом поде выполните команду 14](#_Toc83041615)

[6.Выйдите из тестового пода. Попробуйте удалить под с postgresql 14](#_Toc83041616)

[7.После его пересоздания повторите все с п.1, кроме п.4 Проверьте что созданная ранее таблица никуда не делась (НЕУСПЕХ!!!! Таблицы нет) 15](#_Toc83041617)

[Проблемы 16](#_Toc83041618)

[Проблема 1. Таблица исчезла 16](#_Toc83041619)

[Проблема 2. 0/1 nodes are available: 1 pod has unbound immediate PersistentVolumeClaims. 16](#_Toc83041620)

# Домашнее задание

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 c 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.

# Проверка работоспособности/готовности kubernetes

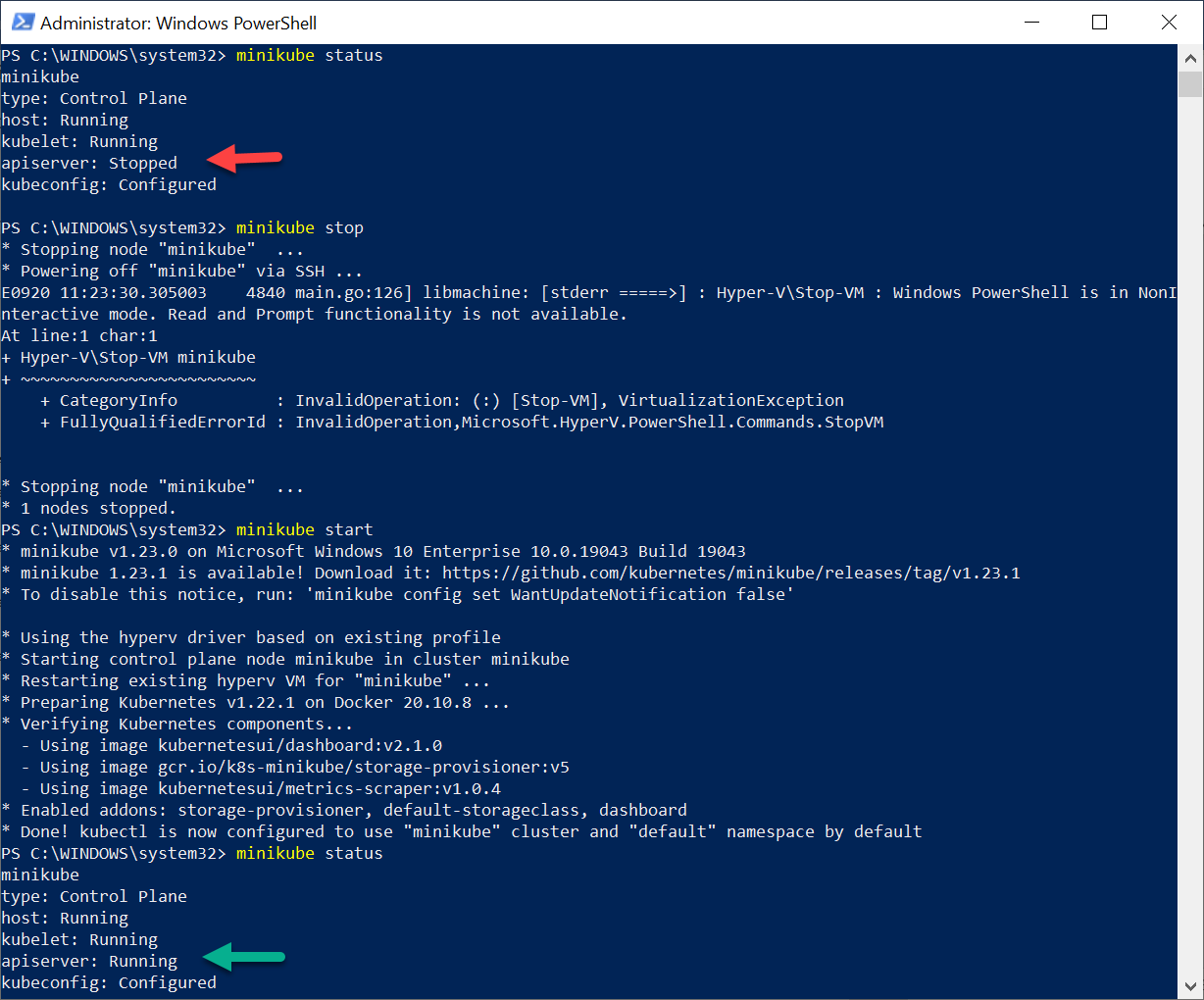
Run PowerShell as administrator

minikube status

minikube stop

minikube start

minikube status



# Подготовка манифестов

## Создание secret.yml

*…*

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

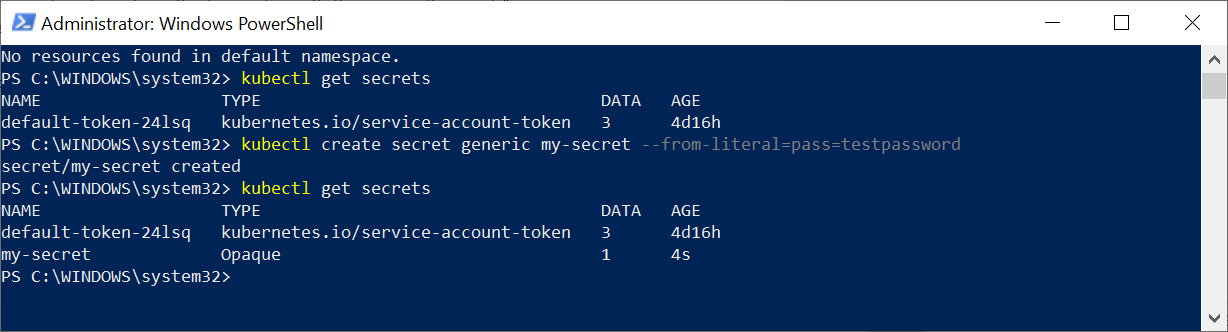
*Для этого нужно указать переменные окружения POSTGRES\_PASSWORD, POSTGRES\_USER, POSTGRES\_DB в деплойменте. При этом значение переменной POSTGRES\_PASSWORD должно браться из секрета.*

*…*

<https://kubernetes.io/docs/tasks/configmap-secret/managing-secret-using-kubectl/>

kubectl get secrets

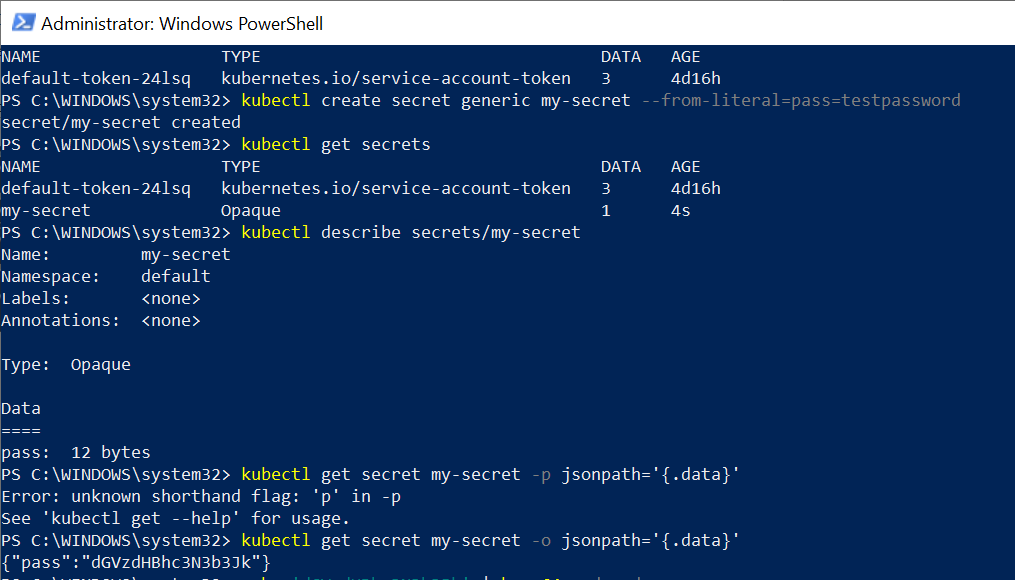
kubectl create secret generic my-secret --from-literal=pass=testpassword



kubectl describe secrets/my-secret

kubectl get secret my-secret -o jsonpath='{.data}'

{"pass":"dGVzdHBhc3N3b3Jk"}



### secret-lesson-04.yml

apiVersion: v1

kind: Secret

metadata:

name: my-secret

type: Opaque

data:

POSTGRES\_PASSWORD: dGVzdHBhc3N3b3Jk

## НЕ УДАЛОСЬ СКОНФИГУРИРОВАТЬ!!! Создание PersistentVolume.yml

ВНИМАНИЕ!!! Не удалось сконфигурировать PersistentVolume (см секцию ПРОБЛЕМЫ в конце документа)

…

*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 c размером диска в 10Gi, замонтированном в pod по пути /var/lib/postgresql/data*

*…*

### pgdata-pvc-disk-lesson-04.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pgdata-pvc-hdd

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 10Gi

storageClassName: csi-ceph-hdd-dp1

### Изменения в deployment.yml

spec:

template:

spec:

containers:

volumeMounts:

- name: pgdata

mountPath: /var/lib/postgresql/data

volumes:

- name: pgdata

persistentVolumeClaim:

claimName: pgdata-pvc-hdd

для исправления ошибки «pod has unbound PersistentVolumeClaims» пришлось добавить следующую секцию

<https://stackoverflow.com/questions/52668938/pod-has-unbound-persistentvolumeclaims>

resources:

requests:

cpu: 100m

memory: 100Mi

limits:

cpu: 100m

memory: 100Mi

## Создание deployment.yml

*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 c размером диска в 10Gi, замонтированном в pod по пути /var/lib/postgresql/data*

### deployment-lesson-04.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: deployment-lesson-04

namespace: my-postgres

spec:

replicas: 1

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 0

maxUnavailable: 1

selector:

matchLabels:

app: deployment-lesson-04

template:

metadata:

labels:

app: deployment-lesson-04

spec:

containers:

- image: postgres:10.13

name: postgresql

ports:

- containerPort: 5432

name: postgresql

protocol: TCP

env:

- name: POSTGRES\_DB

value: testdatabase

- name: POSTGRES\_USER

value: testuser

- name: PGDATA

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

- name: POSTGRES\_PASSWORD

valueFrom:

secretKeyRef:

name: my-secret

key: POSTGRES\_PASSWORD

resources:

requests:

cpu: 100m

memory: 100Mi

limits:

cpu: 100m

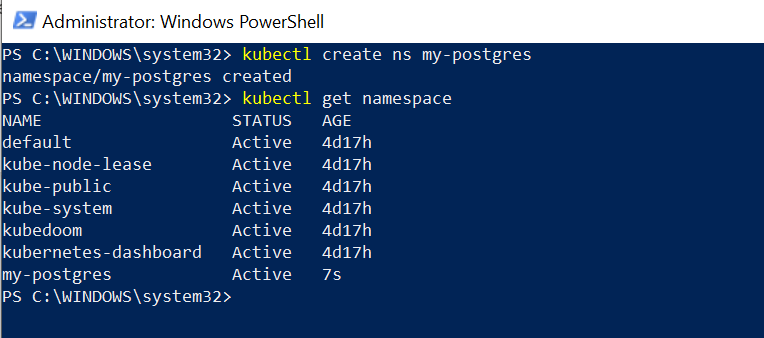
memory: 100Mi

# Применение манифестов

## Cоздание namespace my-postgres

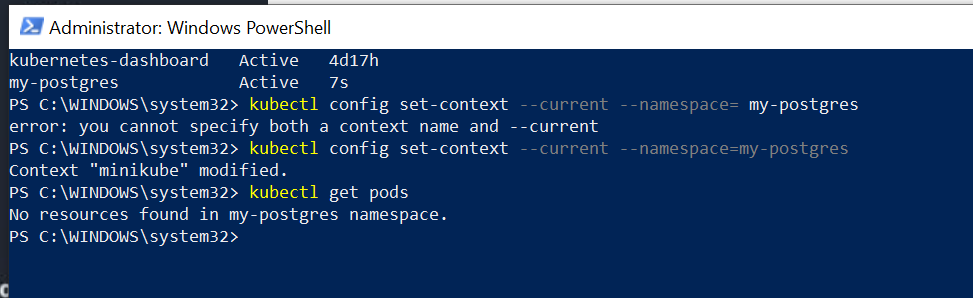
kubectl create ns my-postgres

kubectl get namespace



## Установка контекста в namespace

kubectl config set-context --current --namespace=my-postgres



## Применение манифестов[[1]](#footnote-1)

cd C:\1

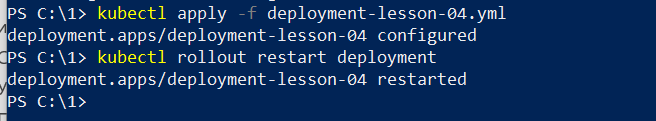
kubectl apply -f secret-lesson-04.yml

kubectl apply -f deployment-lesson-04\_6.yml

## Проверка работосопосбности

Для проверки работоспособности базы данных:

kubectl rollout restart deployment



kubectl describe pod deployment-lesson-04-89f49558d-n8qss

### 1.Узнайте IP пода postgresql

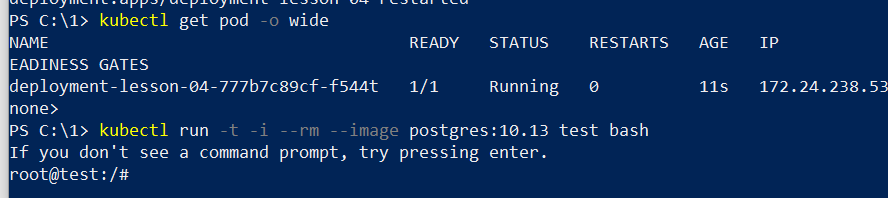
kubectl get pod -o wide



172.24.238.53

### 2.Запустите рядом тестовый под

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

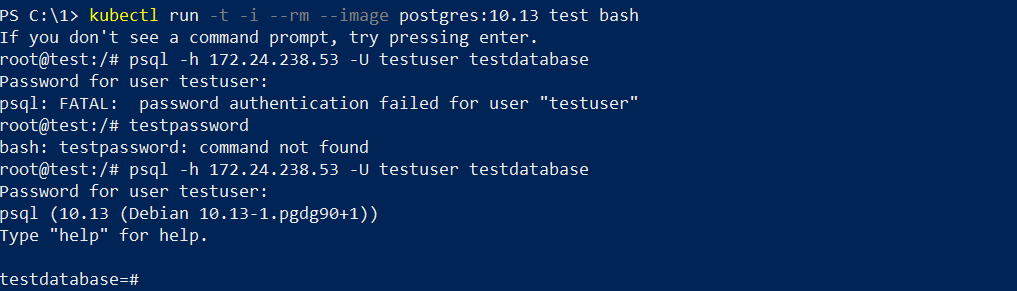


### 3.Внутри тестового пода выполните команду для подключения к БД

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

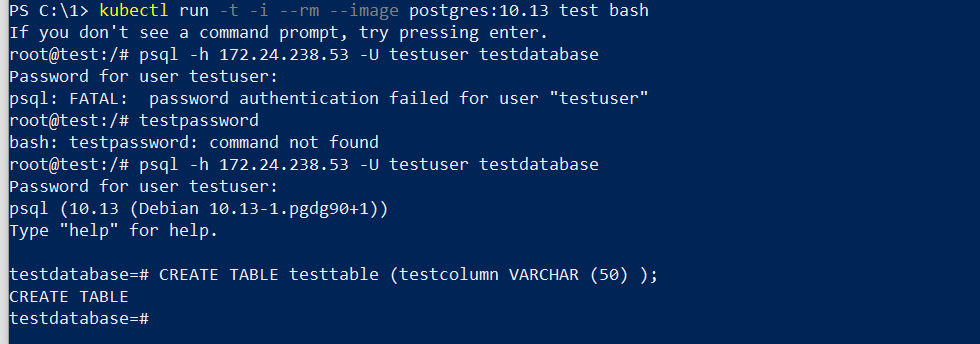
Введите пароль - testpassword

psql -h 172.24.238.53 -U testuser testdatabase



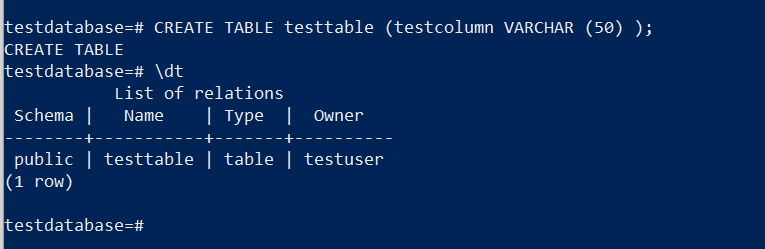
### 4.Все в том же тестовом поде, после подключения к инстансу БД выполните команду для создания таблицы

CREATE TABLE testtable (testcolumn VARCHAR (50) );

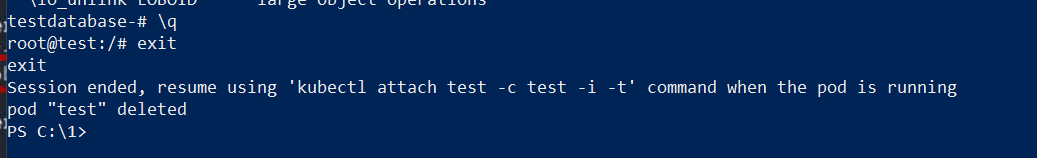


### 5.Проверьте что таблица создалась. Для этого все в том же тестовом поде выполните команду

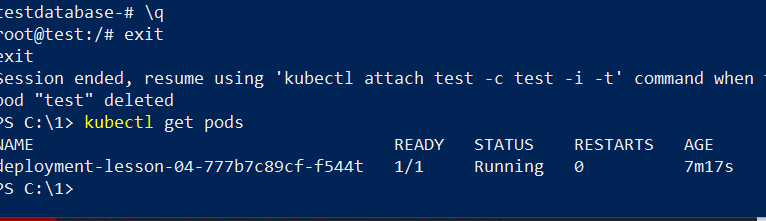
\dt



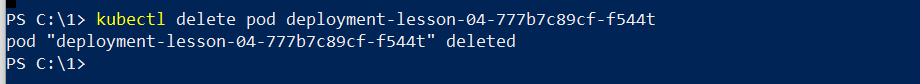
### 6.Выйдите из тестового пода. Попробуйте удалить под с postgresql



Kubectl get pods

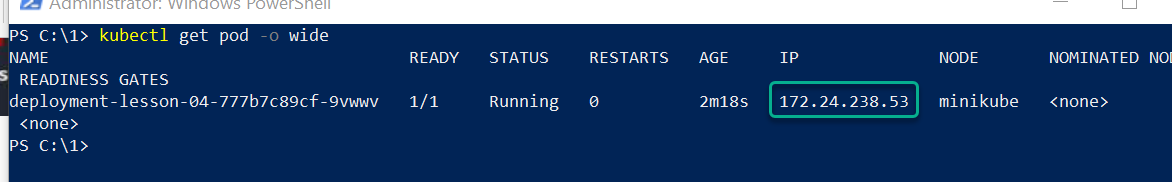


kubectl delete pod deployment-lesson-04-777b7c89cf-f544t

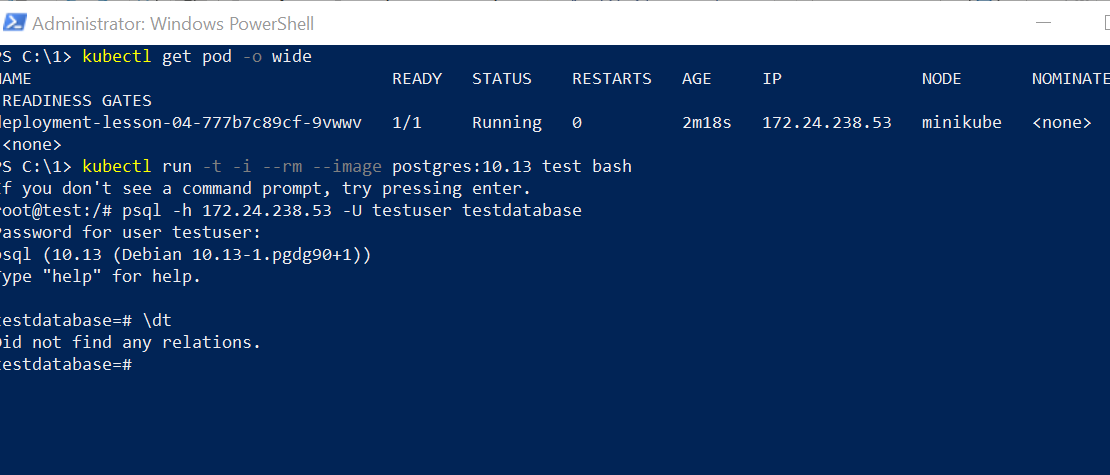


### 7.После его пересоздания повторите все с п.1, кроме п.4 Проверьте что созданная ранее таблица никуда не делась (НЕУСПЕХ!!!! Таблицы нет)

kubectl get pod -o wide

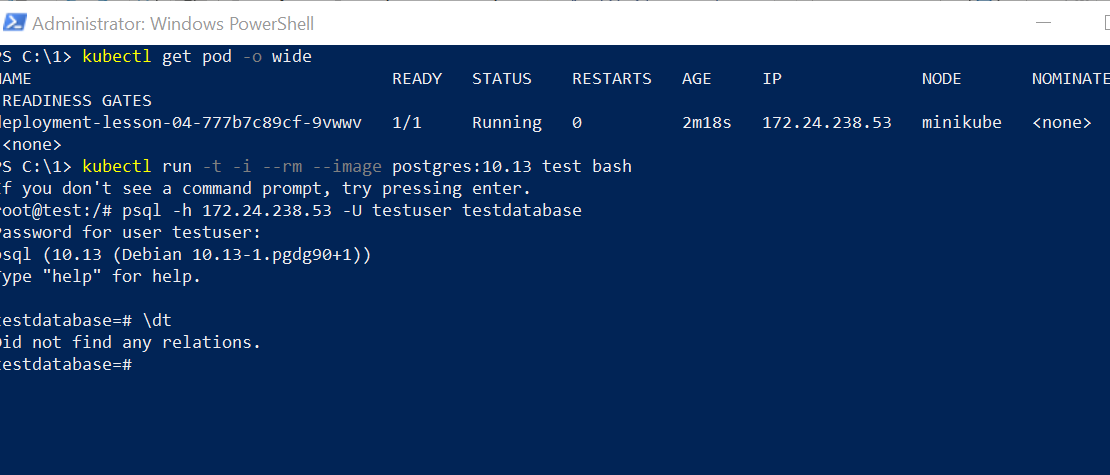


НЕУСПЕХ!!!! Таблицы нет



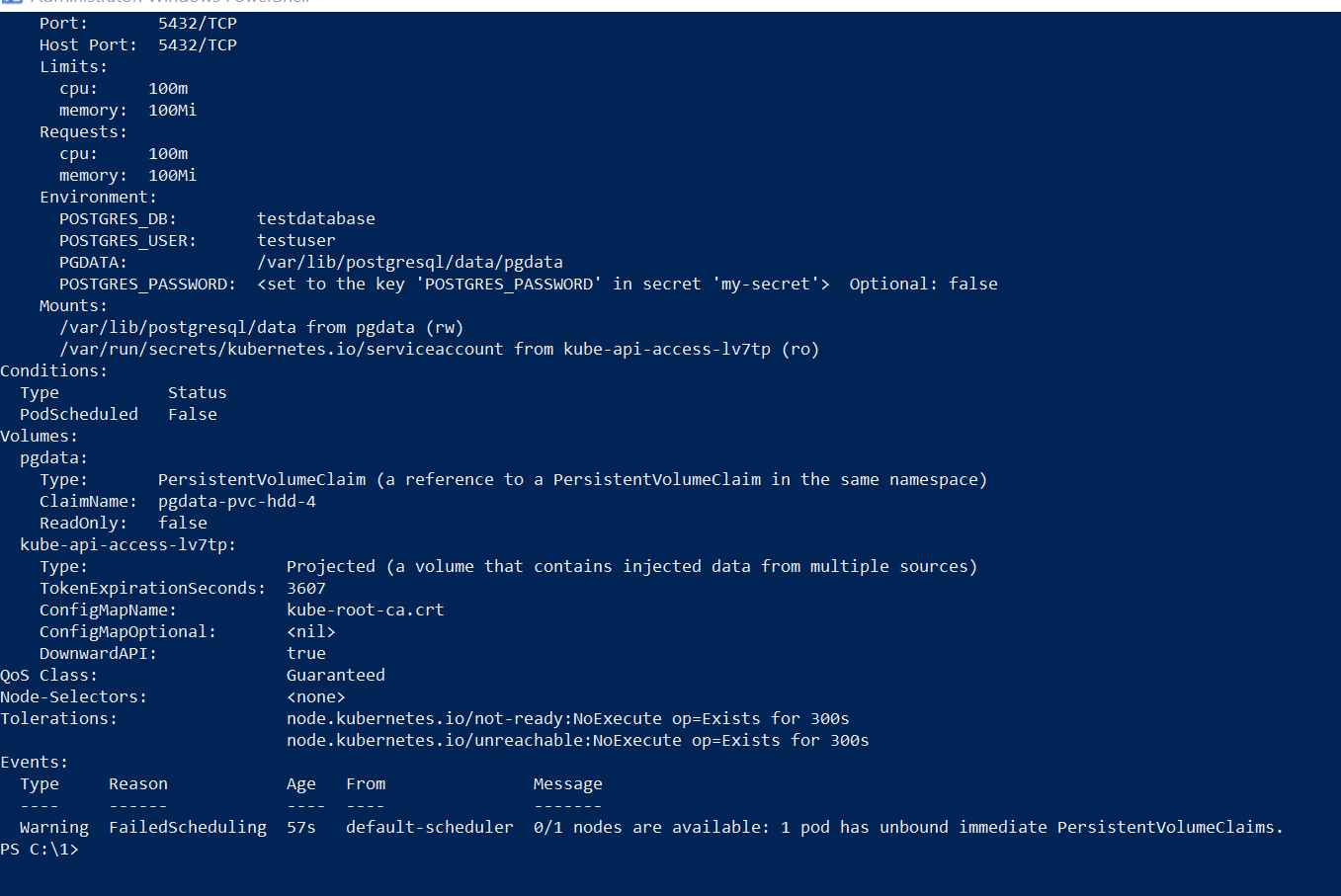
# Проблемы

## Проблема 1. Таблица исчезла



## Проблема 2. 0/1 nodes are available: 1 pod has unbound immediate PersistentVolumeClaims.

kubectl describe pod deployment-lesson-04-6c8dc9bb4b-xqlx8



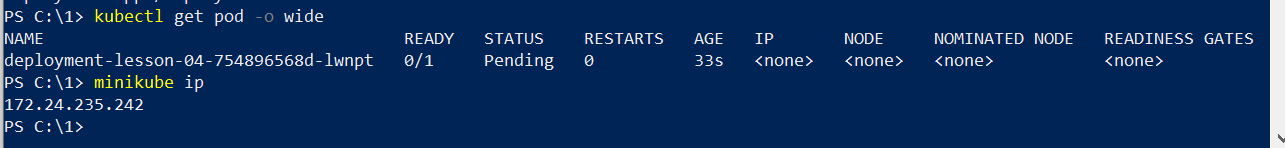
Прочитал документацию <https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/>

Перебрал несколько настроек <https://stackoverflow.com/questions/52668938/pod-has-unbound-persistentvolumeclaims>

Но НЕ СМОГ решить проблему

POD при запуске зависал

kubectl get pod -o wide



1. НЕ усдалось создать persistent storage. Команды оставлю на будущее

   kubectl apply -f pgdata-pvc-disk-lesson-04\_2.yml

   kubectl apply -f pgdata-pvc-disk-lesson-04\_5.yml

   kubectl get pvc pgdata-pvc-hdd-3

   kubectl delete pvc task-pv-claim [↑](#footnote-ref-1)