

Homework_Lesson41_Report

Задание:

1. Создайте новый сервисный аккаунт (Service Account) в вашем кластере Kubernetes.
2. Создайте роль (Role) и привяжите ее к сервисному аккаунту. Роль должна предоставлять разрешения только для выполнения определенных операций, например, чтение и запись конфигураций подов.
3. Создайте привязку роли (Role Binding), связывающую роль с сервисным аккаунтом.
4. Создайте манифест пода, который использует новый сервисный аккаунт.
5. Создайте новый сервис типа ClusterIP, который обслуживает приложение в вашем кластере.
6. Создайте новый сервис типа NodePort, который экспонирует (export) приложение на определенном порту на каждом рабочем узле кластера.
- 7.* Создайте новый сервис типа LoadBalancer, который автоматически создает внешний балансировщик нагрузки для доступа к вашему приложению из внешней сети.
8. Создайте Persistent Volume с использованием локального хранилища.
9. Создайте Persistent Volume Claim, чтобы запросить доступ к Persistent Volume, созданному в предыдущем шаге.
10. Создайте Pod, который монтирует Persistent Volume Claim в качестве тома и выполняет операции чтения и записи данных.
- 11.* Измените Storage Class для вашего Persistent Volume, чтобы использовать другой тип хранилища, например, предоставляемый облачным провайдером. (S3). Обновите Persistent Volume Claim, чтобы использовать обновленный Storage Class.

1 Наш созданный Service Account.

```
PS D:\kubernetes\kubernetes2> kubectl get sa -n blue
NAME      SECRETS  AGE
blue-sa   0        36m
default   0        7d20h
```

2 Наша созданная Role.

```
PS D:\kybernetes\kubernetes2> kubectl get roles -n blue
NAME          CREATED AT
pod-redactor  2025-04-11T08:58:58Z
```

3. Созданная Role Bindings.

```
PS D:\kybernetes\kubernetes2> kubectl get rolebinding -n blue
NAME                     ROLE          AGE
pod-redactor-binding    Role/pod-redactor  39m
```

4. Созданный Persistent Volume

```
PS D:\kybernetes\kubernetes2> kubectl get pv -n blue
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  VOLUMEATTRIBUTESCLASS
local-pv      1Gi       RWO           Retain          Bound   blue/local-pvc  local-storage  <unset>
```

5. Созданный Persistent Volume Claim.

```
PS D:\kybernetes\kubernetes2> kubectl get pvc -n blue
NAME          STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
local-pvc     Bound   local-pv  1Gi       RWO           local-storage  <unset>                 3h
```

Наши созданные поды с двумя приложениями.

К двум нашим подам привязан сервис аккаунт blue-sa.

К поде my-aap подключается persistent volume claim local-pvc и монтируется в каталог контейнера /app/static

```
PS D:\kybernetes\kubernetes2> kubectl get pod -n blue
NAME          READY  STATUS   RESTARTS  AGE
my-aap        1/1    Running  0          32m
pod-nginx-sa  1/1    Running  0          3h26m
```

Запущенные сервисы с типом ClusterIP и NodePort.

```
PS C:\Users\Admin> kubectl get svc -n blue
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
clusterip-nginx  ClusterIP   34.118.230.169  <none>       8080/TCP         71m
nodeport-myapp   NodePort    34.118.235.28   <none>       5000:31028/TCP   68m
```

Подключимся к нашему приложению через IP Ноды и порт.

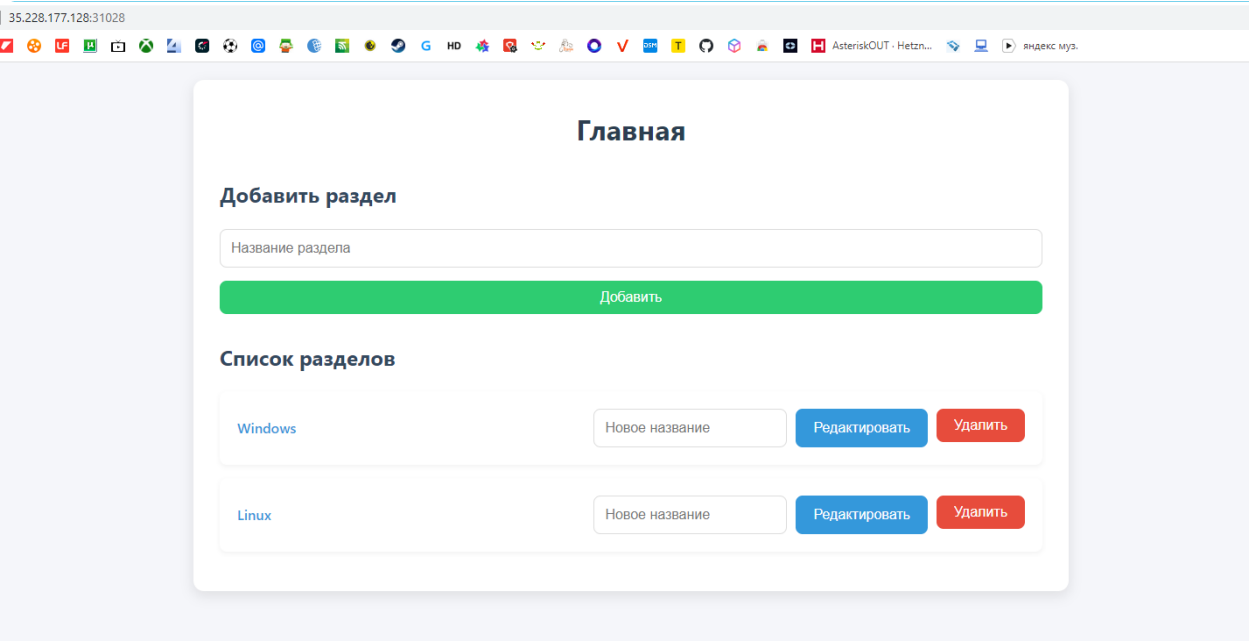
Посмотрим на какой ноды крутится наша пода my-aap.

```
PS C:\Users\Admin> kubectl get pods -o wide -n blue
NAME          READY  STATUS   RESTARTS  AGE  IP          NODE
my-aap        1/1    Running  0          104m  10.96.0.20  gke-my-gke-cluster-my-node-pool-d3b235a2-2854
pod-nginx-sa  1/1    Running  0          4h39m  10.96.0.10  gke-my-gke-cluster-my-node-pool-d3b235a2-2854
```

Посмотрим внешний IP ноды.

```
PS C:\Users\Admin> kubectl get nodes -o wide -n blue
NAME                                                    STATUS  ROLES  AGE  VERSION  INTERNAL-IP  EXTERNAL-IP
gke-my-gke-cluster-my-node-pool-d3b235a2-2854        Ready   <none>  9d   v1.31.6-gke.1020000  10.166.0.36  35.228.177.128
gke-my-gke-cluster-my-node-pool-d3b235a2-stq3        Ready   <none>  9d   v1.31.6-gke.1020000  10.166.0.37  34.88.122.133
```

Как видим приложение доступно.



Вот наши поды, которые будут прокидываться через Load Balancer наружу.

```
PS D:\kubernetes\kubernetes> kubectl get pods -n red
NAME                                READY   STATUS    RESTARTS   AGE
http-aap-ff9dbf875-jdp96           1/1     Running   0           11s
http-aap-ff9dbf875-vctzr           1/1     Running   0           11s
```

Сервис типа Load Balancer.

```
PS D:\kubernetes\kubernetes> kubectl get svc -n red
NAME                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
http-aap-service    LoadBalancer  34.118.232.126 35.228.106.45 5050:31179/TCP   8d
```

Как видим приложение доступно.

