Services

Сервис используется для подключения подов к внешней сети. Сервис использует метки для выбора подов. Все указанные метки должны быть на поде (дополнительные метки пода игнорируются)

В сервисе - раздел selector

```
spec:
  replicas: 10
  <Snip>
  template:
    metadata:
      labels:
        project: tkb
        zone: prod
<Snip>
---
apiVersion: v1
kind: Service
metadata:
  name: tkb
spec:
  ports:
  - port: 8080
  selector:
    project: tkb
    zone: prod
```

Типы сервисов

ClusterIP Используется для доступности подов внутри кластера. Доступность по имени сервиса.

NodePort Services Используется для доступа приложений снаружи кластера. Добавляет указанный порт на каждую ноду.

```
apiVersion: v1
kind: Service
```

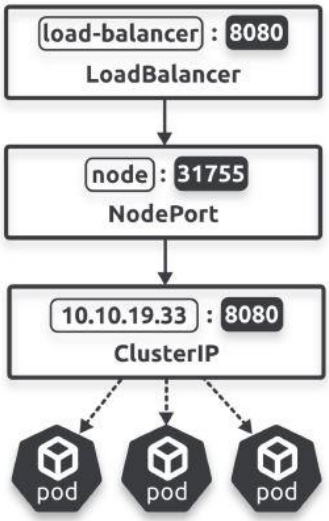
```
metadata:
  name: skippy <<==== Registered with the internal cluster DNS (ClusterIP)
spec:
  type: NodePort <<==== Service type
  ports:
  - port: 8080 <<==== ClusterIP port
    targetPort: 9000 <<==== Application port in container
    nodePort: 30050 <<==== External port on every cluster node (NodePort)
selector:
  app: hello-world
```

В данном примере изнутри под доступен по порту 8080, снаружи - по порту 30050. Диапазон портов 30000-32767.

LoadBalancer Services Используется для сервисов со стартовым диапазоном портов, наиболее часто.

```
apiVersion: v1
kind: Service
metadata:
  name: lb <<==== Registered with cluster DNS
spec:
  type: LoadBalancer
  ports:
  - port: 8080 <<==== Load balancer port
    targetPort: 9000 <<==== Application port inside container
selector:
  project: tkb
```

Мой кубер использует flannel, а для корректной работы LoadBalancer нужен балансировщик, например MetalLB. Стек в случае балансировщика:



Headless сервисы

Сервисы без IP адреса. Их цель - создать DNS записи для StatefulSet подов. Клиенты запрашивают DNS имена подов и направляют непосредственно им запросы вместо использования кластерного IP. Пример сервиса:

```
apiVersion: v1
kind: Service <<==== Normal Kubernetes Service
metadata:
  name: dullahan
  labels:
    app: web
spec:
  ports:
    - port: 80
  name: web
  clusterIP: None <<==== Make this a headless Service
  selector:
    app: web
```

Структура YAML файла

Верхний уровень

Параметр	Описание
kind	Тип, в данном случае Service
spec	Спецификация

spec

Параметр	Описание
type	тип (ClusterIP, NodePort, LoadBalancer)
ports	port - порт фронтенда targetPort - порт на бэке
selector	правила выбора меток

Пример yaml файла

```
apiVersion: v1
kind: Service
metadata:
  name: cloud-lb
spec:
  type: LoadBalancer
  ports:
  - port: 9000
    targetPort: 8080
  selector:
    chapter: services
```

Основные команды

Команда	Доп. пар.	Описание
kubectl expose deployment dep_name --type=LoadBalancer		Ручное создание сервиса. Вот только не поехало.
kubectl get svc -o wide		Информация по сервисам
kubectl get endpointslices		Список ендпоинтов
kubectl describe endpointslice epname		Описание ендпоинта

Revision #6
Created 21 March 2025 15:55:48 by Admin
Updated 28 March 2025 08:28:54 by Admin