

# Сетевое взаимодействие в K8s. Часть 1

DNS, типы Services



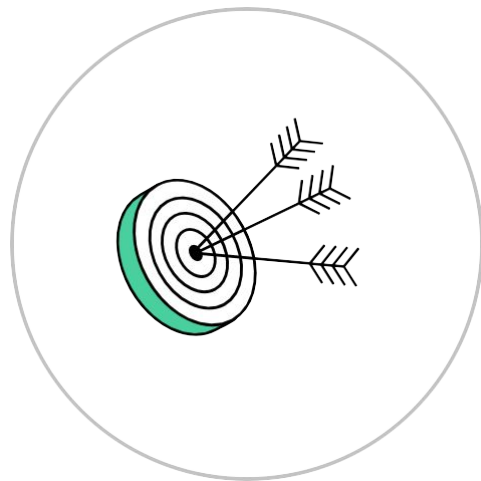
# Кирилл Касаткин

DevOps-инженер, Renuе



# Цели занятия

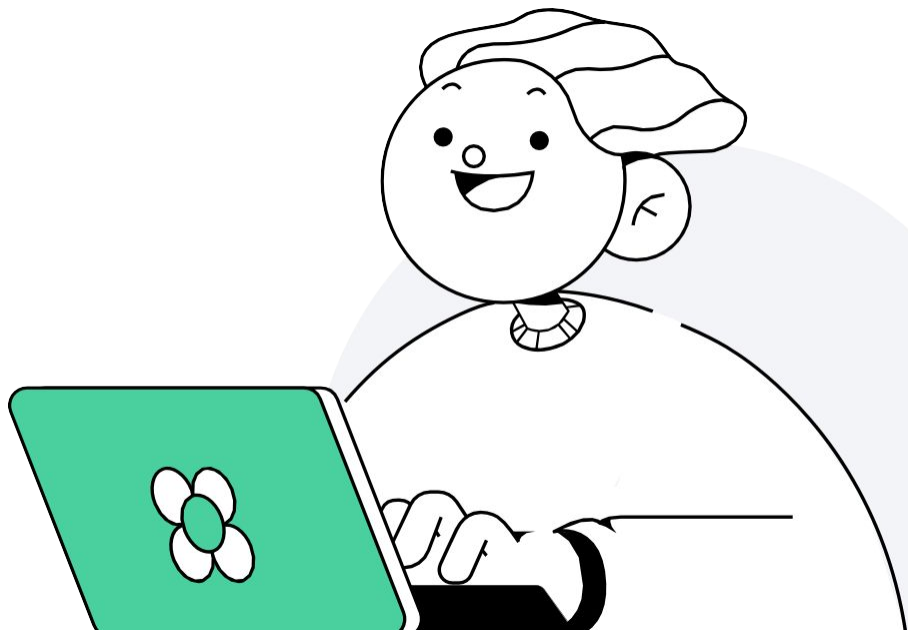
- Узнать:
  - какие DNS-записи создаются внутри кластера
  - какие типы Service бывают и для чего они нужны
- Познакомиться с примерами манифестов различных сервисов
- Понять, каким образом можно получить доступ к приложениям снаружи кластера
- Разобрать примеры манифестов объектов K8s



# План занятия

- 1 DNS для подов и сервисов
- 2 Типы Service
- 3 ClusterIP
- 4 NodePort
- 5 LoadBalancer
- 6 Итоги
- 7 Домашнее задание

\*Нажми на нужный раздел для перехода



# Вспоминаем прошлое занятие

**Вопрос:** что такое Service?



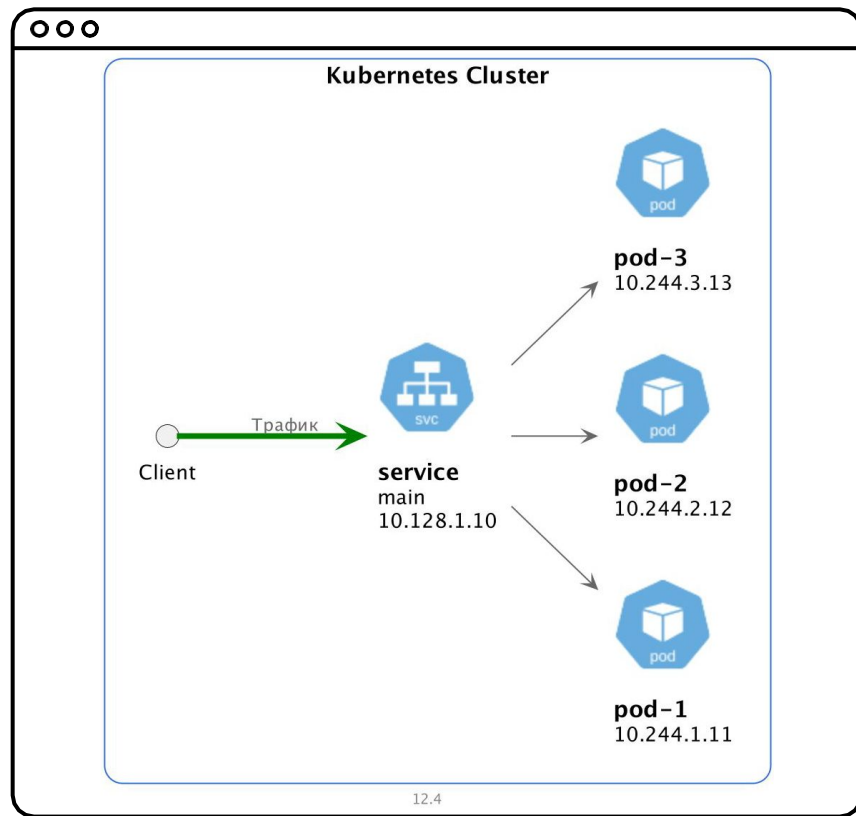
# Вспоминаем прошрое занятие

**Вопрос:** что такое Service?

**Ответ:** ресурс K8s, который предоставляет возможность доступа к приложению, состоящему из множества подов на L4-уровне (protocol:port)



# Вспоминаем прошлое занятие



# DNS для подов и сервисов



1



# DNS для подов и сервисов

K8s создаёт DNS-записи для сервисов и подов.

Сервисы и поды внутри Namespace могут быть доступны по короткому имени.

**Сервисы и поды имеют записи следующего формата:**

```
service-name.namespace-name.svc.cluster-domain.example  
pod-ip-address.namespace-name.pod.cluster-domain.example
```

# Типы Service

ClusterIP, NodePort, LoadBalancer



2

# Типы Service

Иногда требуется обеспечить доступ к приложению извне кластера K8s. Например, внешний мир должен иметь доступ к frontend вашего приложения. В таком случае требуется использовать другие типы Service.

## Тип сервиса определяет его поведение

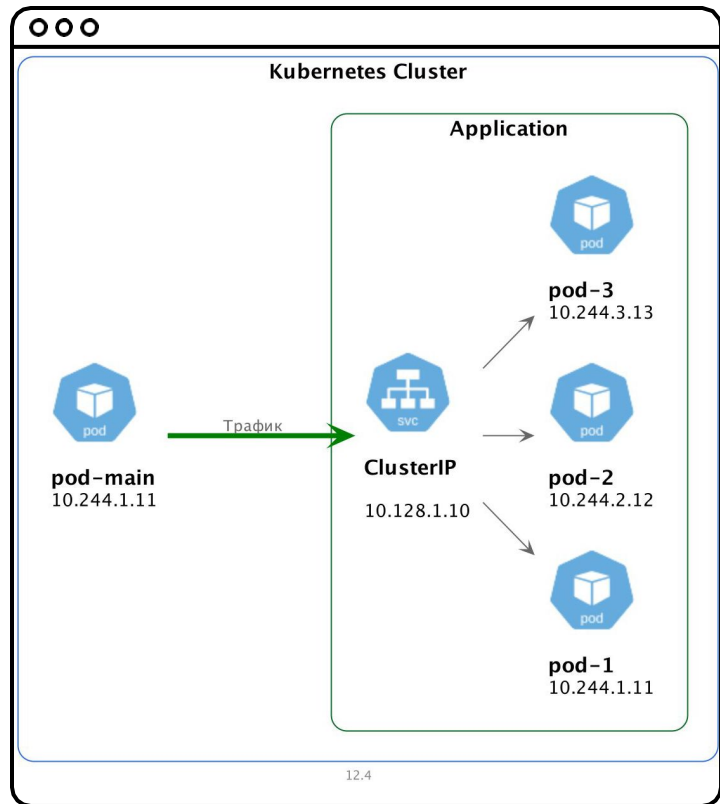
- 1 ClusterIP
- 2 NodePort
- 3 LoadBalancer

# ClusterIP



3

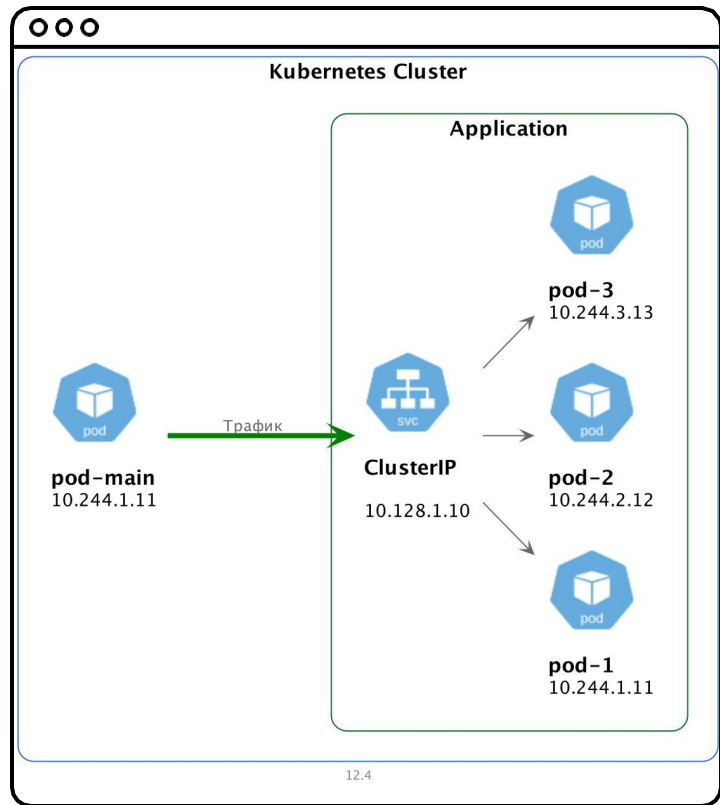
# Типы Service. ClusterIP



- Выставляет сервис внутри кластера на внутреннем IP-адресе



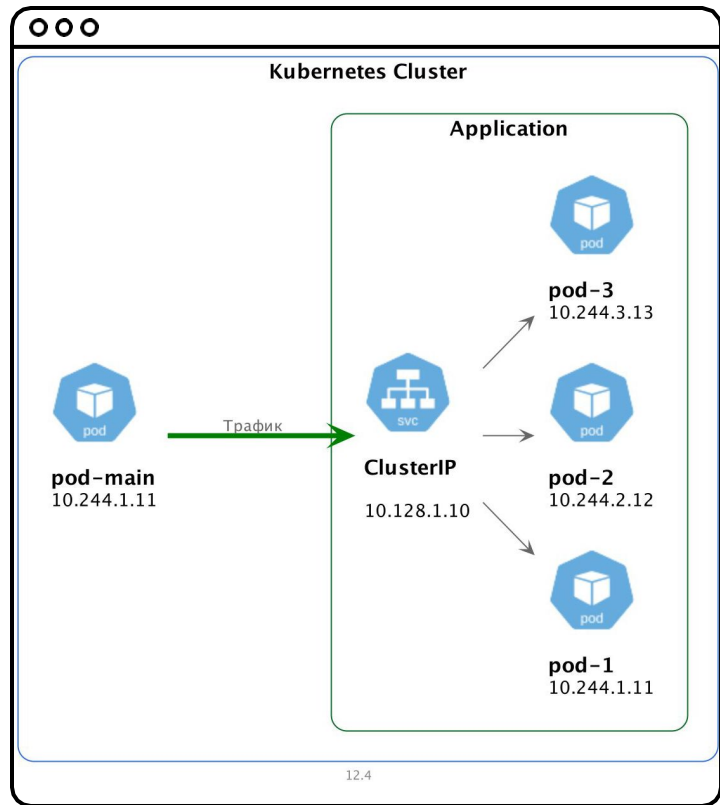
# Типы Service. ClusterIP



- Выставляет сервис внутри кластера на внутреннем IP-адресе
- Такой сервис доступен **только** внутри кластера



# Типы Service. ClusterIP



- Выставляет сервис внутри кластера на внутреннем IP-адресе.
- Такой сервис доступен **только** внутри кластера.
- Является типом по умолчанию



# Пример конфигурации

Пример конфигураций Service типа ClusterIP

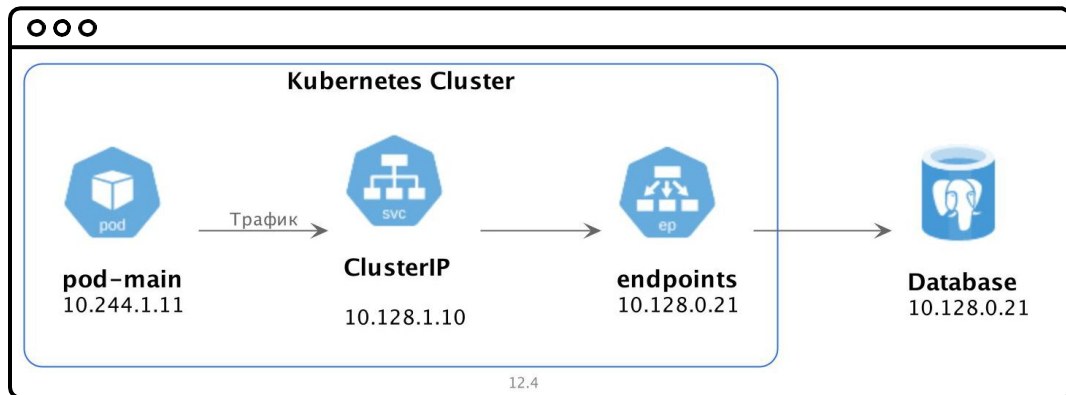
```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  namespace: default
spec:
  ports:
    - name: web
      port: 80
  selector:
    app: nginx
  type: ClusterIP
```





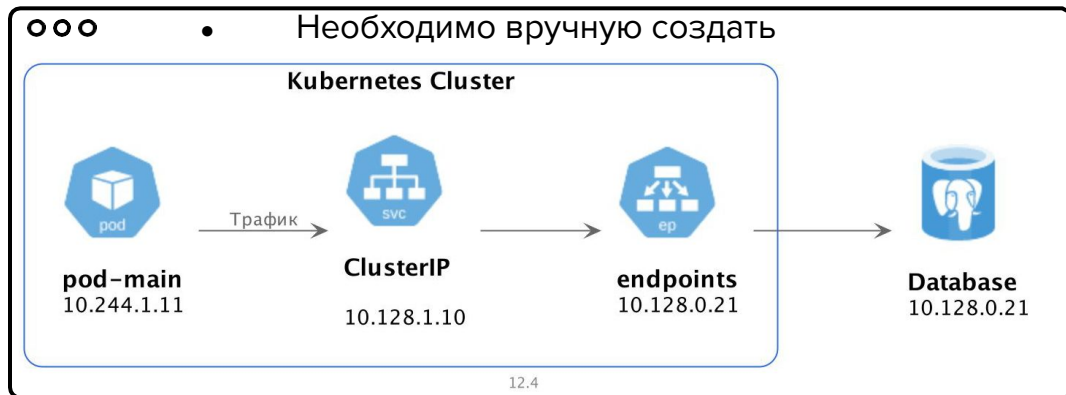
# Типы Service. ClusterIP без селекторов

ClusterIP без указания Selector обычно используются для обращения за пределы кластера по IP-адресу к какому-либо приложению

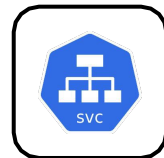


# Типы Service. ClusterIP без селекторов

ClusterIP без указания Selector обычно используются для обращения за пределы кластера по IP-адресу к какому-либо приложению.

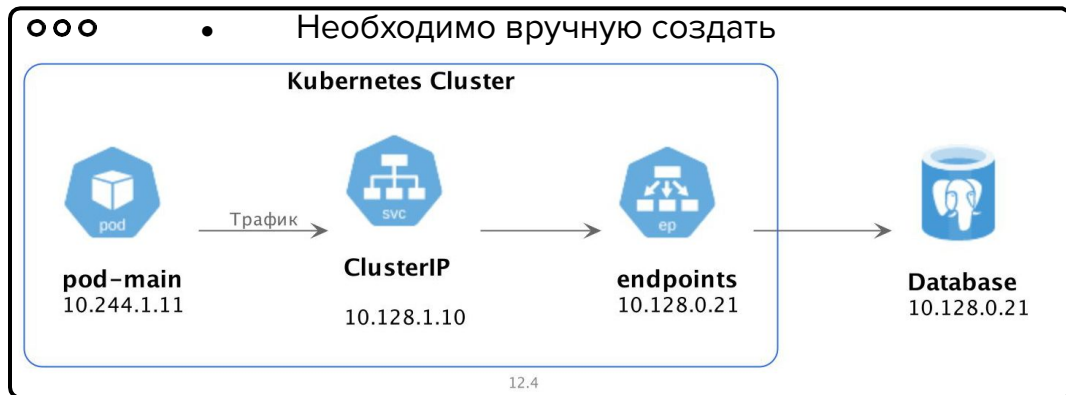


endpoints



# Типы Service. ClusterIP без селекторов

ClusterIP без указания Selector обычно используются для обращения за пределы кластера по IP-адресу к какому-либо приложению.



endpoints

- Имя сервиса и endpoints должны совпадать

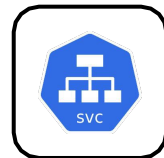


# Пример конфигурации

Пример конфигураций Service типа ClusterIP без селектора + пример endpoints

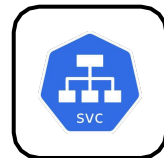
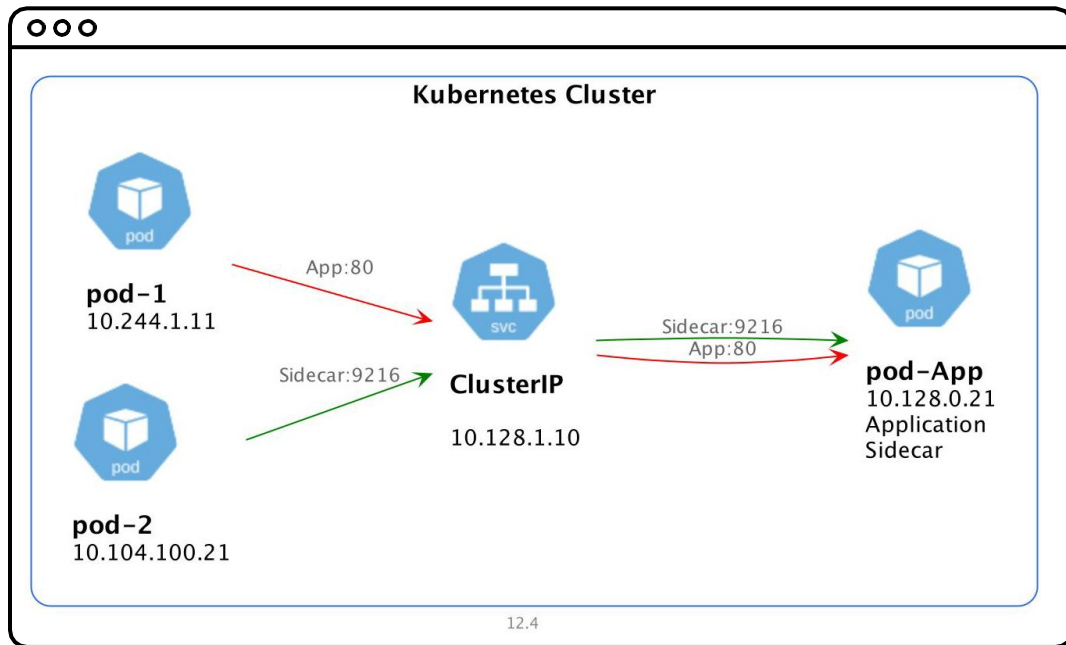
```
apiVersion: v1
kind: Service
metadata:
  name: out-db
  namespace: default
spec:
  ports:
    - protocol: TCP
      port: 35432
      targetPort: 5432
```

```
apiVersion: v1
kind: Endpoints
metadata:
  name: out-db
  namespace: default
subsets:
  - addresses:
      - ip: 10.128.0.21
    ports:
      - port: 5432
```



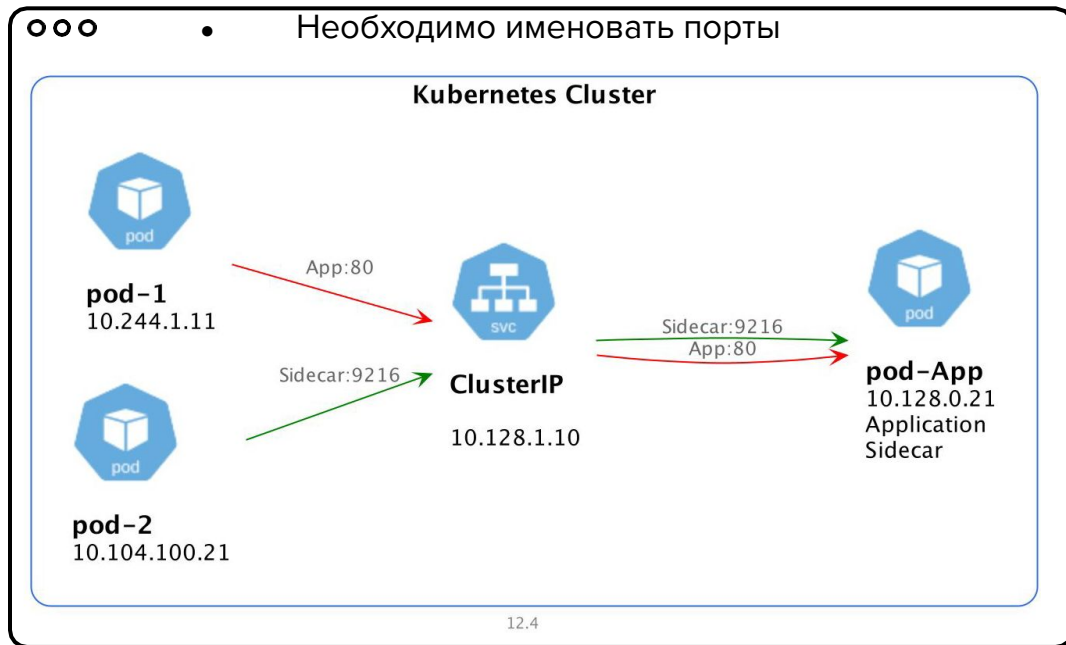
# Типы Service. ClusterIP MultiPort

ClusterIP MultiPort используется тогда, когда необходимо получить доступ к нескольким портам пода



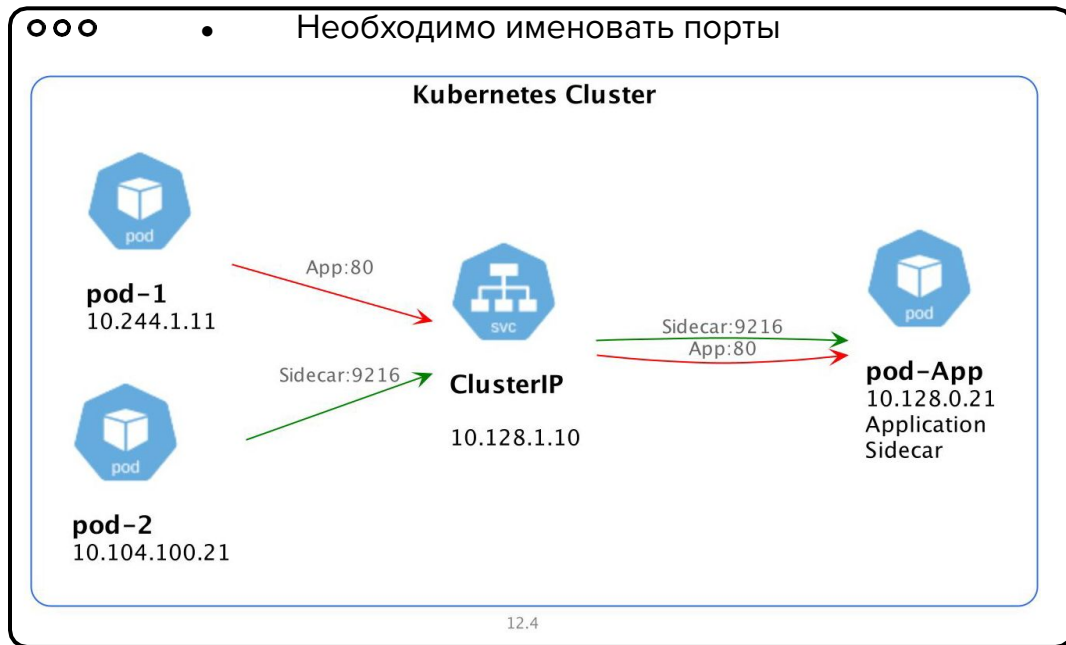
# Типы Service. ClusterIP MultiPort

ClusterIP MultiPort используется тогда, когда необходимо получить доступ к нескольким портам пода.



# Типы Service. ClusterIP MultiPort

ClusterIP MultiPort используется тогда, когда необходимо получить доступ к нескольким портам пода



- Обычно используется для доступа к различным контейнерам (sidecar)



# Пример конфигурации

Пример конфигураций Service типа ClusterIP MultiPort

```
apiVersion: v1
kind: Service
metadata:
  name: my-svc
spec:
  selector:
    app: MyApp
  ports:
    - name: web
      protocol: TCP
      port: 80
      targetPort: 8080
    - name: db
      protocol: TCP
      port: 9216
      targetPort: 19216
```





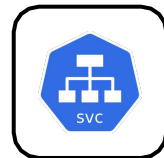
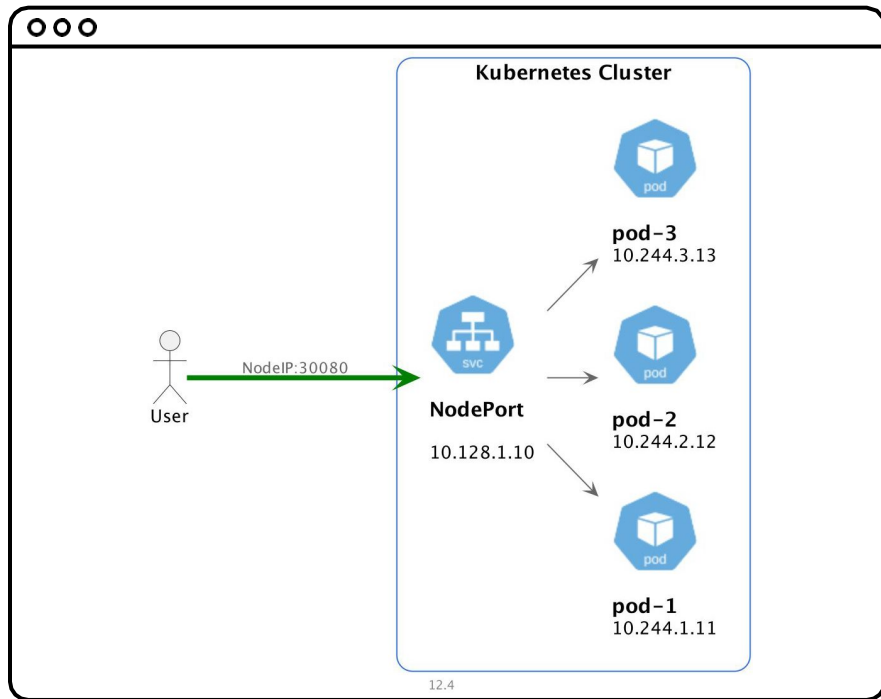
# NodePort



4

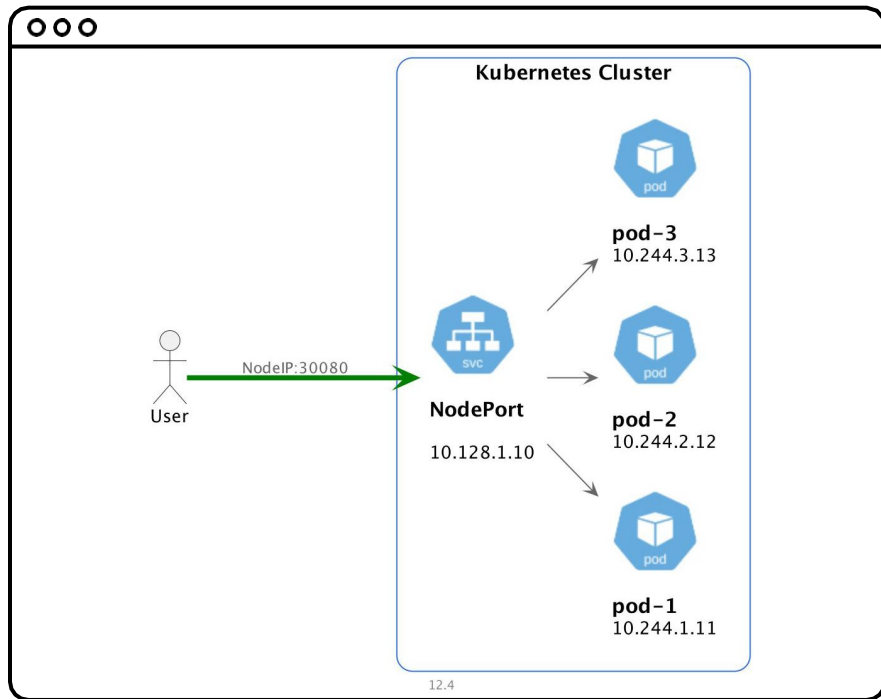
# Типы Service. NodePort

Делает доступным сервис на статическом порту ноды, где запущен



# Типы Service. NodePort

Делает доступным сервис на статическом порту ноды, где запущен.

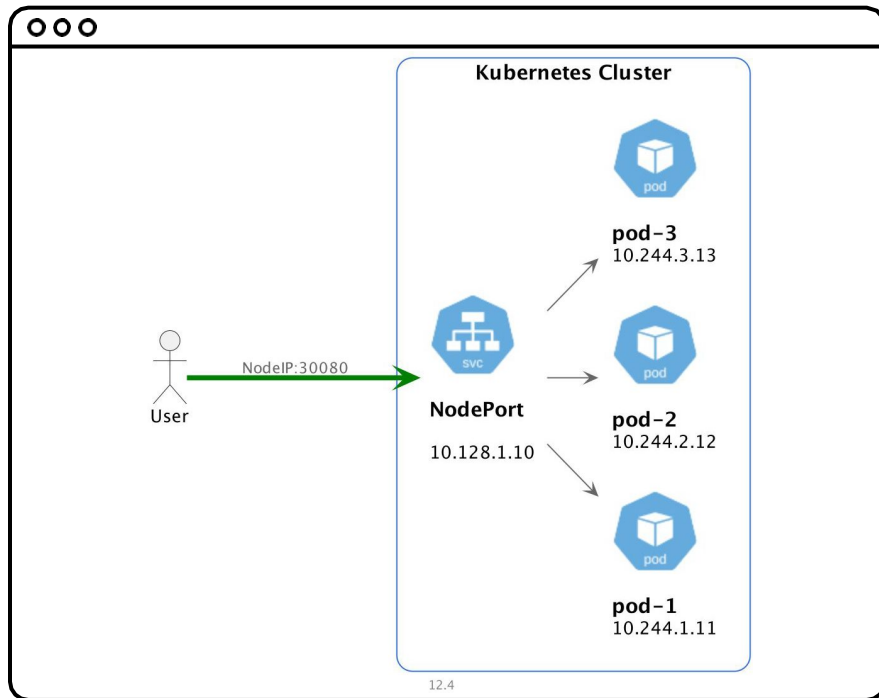


- Можно получить доступ к сервису снаружи кластера по IP-адресу ноды. Этот NodePort открыт на **каждом узле** даже в тех случаях, когда ни один под на узле не запущен



# Типы Service. NodePort

Делает доступным сервис на статическом порту ноды, где запущен

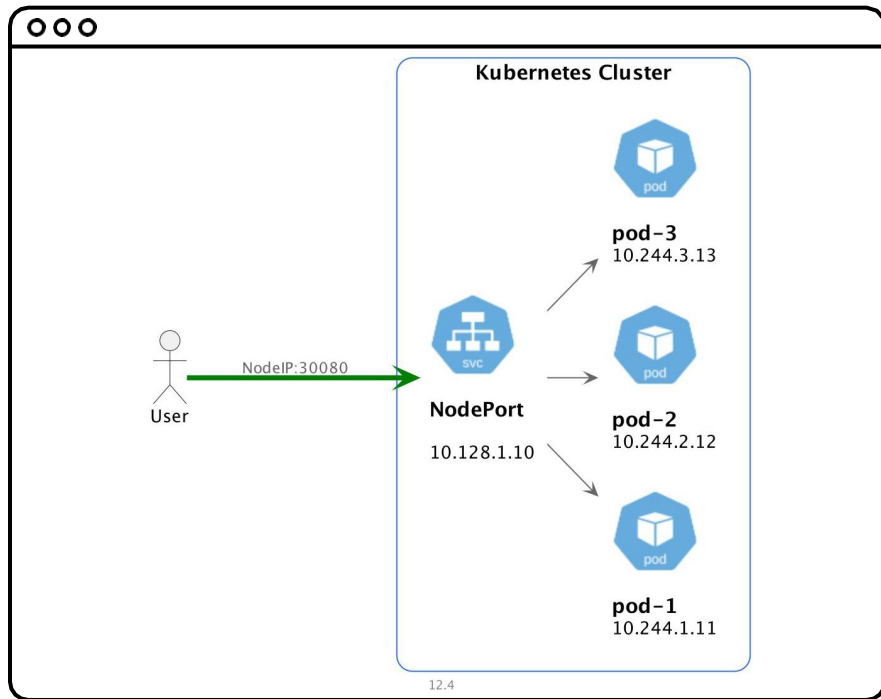


- Можно получить доступ к сервису снаружи кластера по IP-адресу ноды. Этот NodePort открыт на **каждом узле** даже в тех случаях, когда ни один под на узле не запущен
- Номер порта для сервиса NodePort должен быть в диапазоне 30000–32767



# Типы Service. NodePort

Делает доступным сервис на статическом порту ноды, где запущен



- Можно получить доступ к сервису снаружи кластера по IP-адресу ноды. Этот NodePort открыт на **каждом узле** даже в тех случаях, когда ни один под на узле не запущен
- Номер порта для сервиса NodePort должен быть в диапазоне 30000–32767
- При создании сервиса NodePort, будет автоматически создан ещё и сервис ClusterIP, к которому обращается сервис NodePort



# Пример конфигурации

Пример конфигураций Service типа NodePort

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  namespace: default
spec:
  ports:
    - name: web port:
      80 protocol:
        TCP nodePort:
          30080
  selector:
    app: nginx
  type: NodePort
```



# LoadBalancer

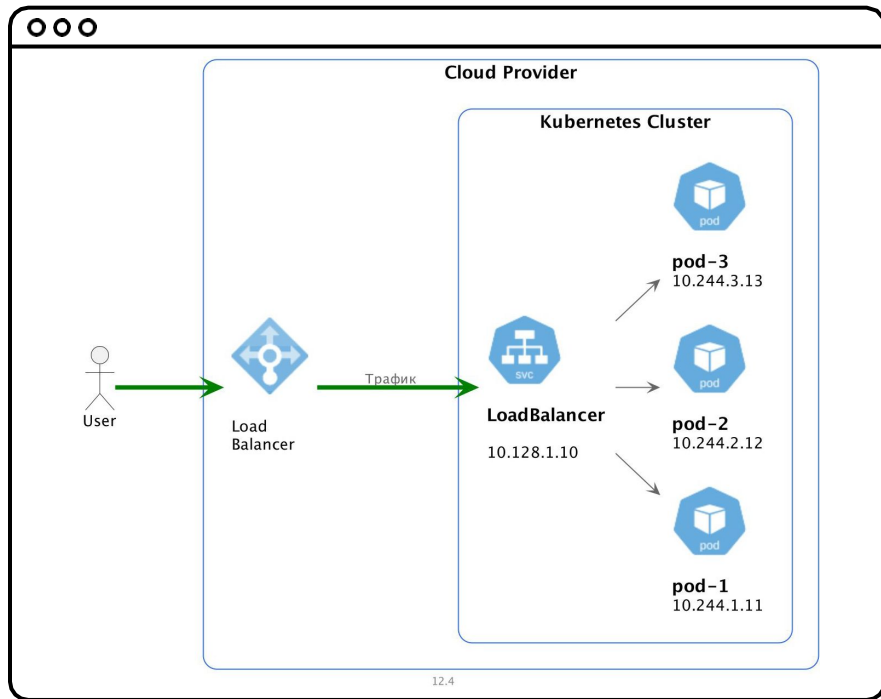


5

The diagram shows two overlapping circles on a teal background. The left circle is white with a teal number '5' in the center. The right circle is teal with a white outline and is partially cut off by the right edge of the frame.

# Типы Service. LoadBalancer

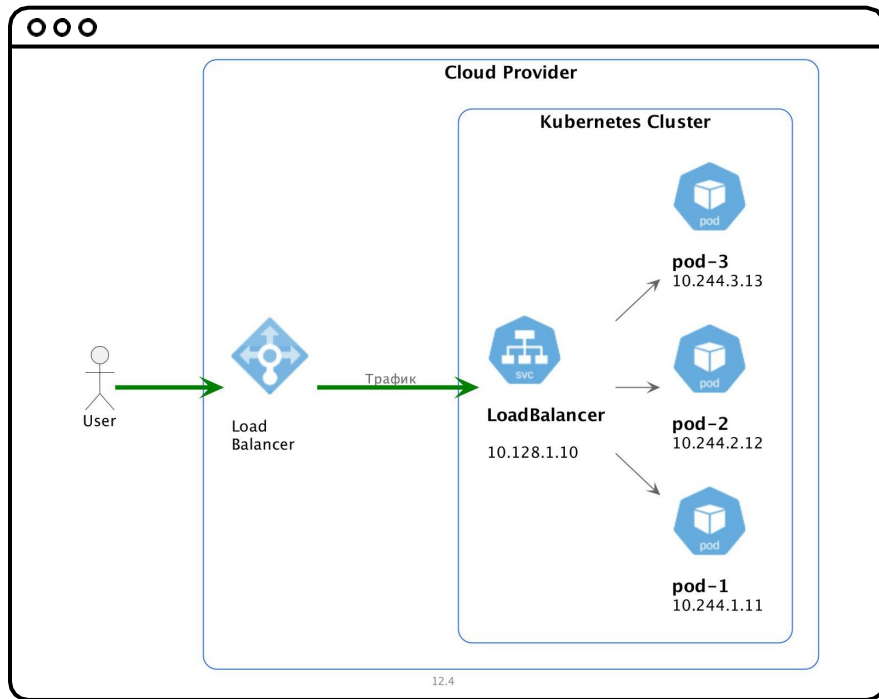
Делает доступным сервис снаружи кластера с помощью облачного балансировщика нагрузки





# Типы Service. LoadBalancer

Делает доступным сервис снаружи кластера с помощью облачного балансировщика нагрузки.

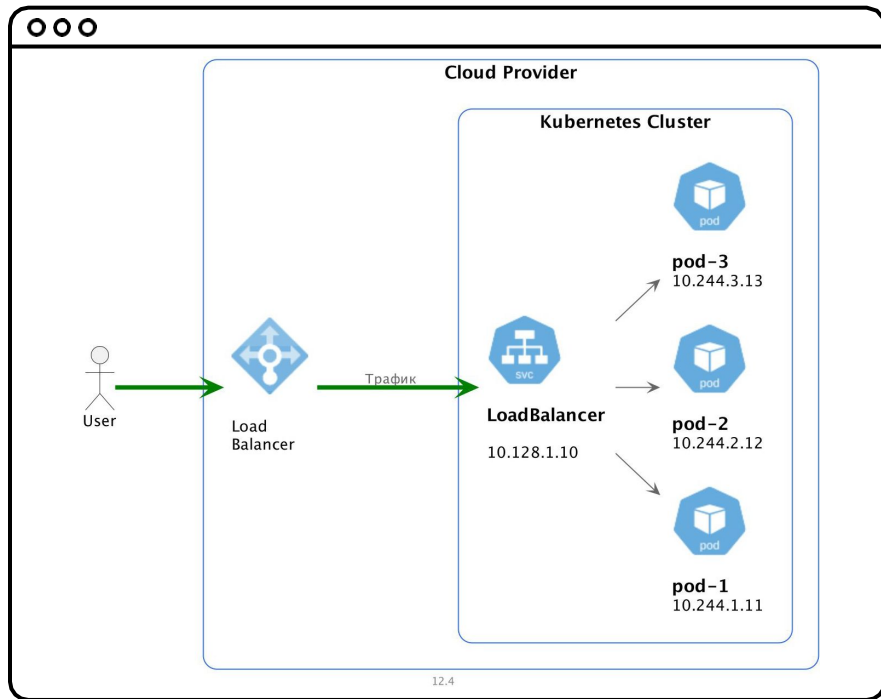


- Автоматически создаются сервисы NodePort и ClusterIP, к которым обращается сервис LoadBalancer

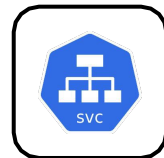


# Типы Service. LoadBalancer

Делает доступным сервис снаружи кластера с помощью облачного балансировщика нагрузки.



- Автоматически создаются сервисы NodePort и ClusterIP, к которым обращается сервис LoadBalancer
- Этот тип сервиса работает только в облачных провайдерах (используется Cloud Controller K8s). В противном случае такой сервис зависнет в стадии Pending



# Пример конфигурации

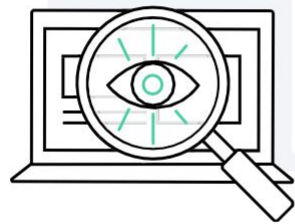
Пример конфигураций Service типа LoadBalancer

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  namespace: default
spec:
  ports:
    - name: web
      port: 80
      protocol: TCP
      targetPort: 9376
  selector:
    app: nginx
  clusterIP:
    10.0.171.239
  type:
    LoadBalancer
```



# Демонстрация работы

Работа с различными типами Service, связь с DNS



# Итоги

## Сегодня мы:

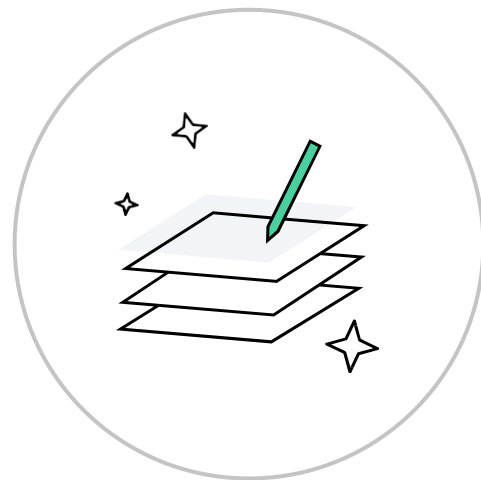
- 1 Узнали, как формируются DNS-имена у подов и сервисов
- 2 Разобрались, какие типы сервисов бывают и где применяется каждый из них
- 3 Рассмотрели примеры манифестов объектов K8s
- 4 Попробовали подключиться к кластеру и посмотреть объекты, пройденные на занятии в работе



# Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



# Дополнительные материалы

- [Статья](#) о Services
- [Статья](#) о том, как работает сеть в K8s



**Задавайте вопросы  
и пишите отзыв о лекции**

