

Базовые объекты K8s

Pod, Service, Endpoints, Labels, port-forward



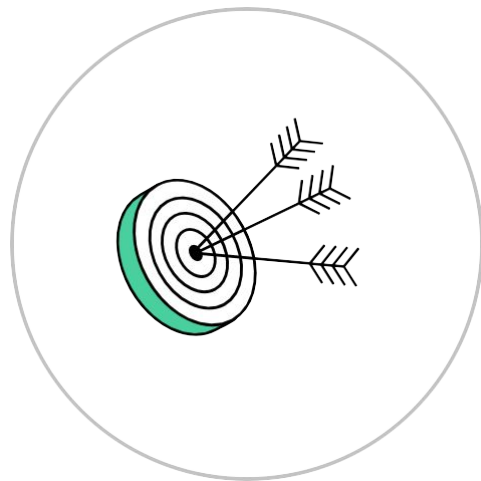
Кирилл Касаткин

DevOps-инженер, Renuе



Цели занятия

- Познакомиться с базовыми объектами Kubernetes (Pod, Labels, Service, Endpoints)
- Обсудить, какие задачи выполняют эти объекты
- Рассмотреть варианты конфигурации объектов K8s
- Научиться связывать Pods и Services, а также получать доступ к Pod снаружи трафика



План занятия

- 1 Pod
- 2 Конфигурация объектов
- 3 Labels, Selector
- 4 Service
- 5 Endpoints
- 6 Подключение к Pod
- 7 Итоги
- 8 Домашнее задание

*Нажми на нужный раздел для перехода

Pod



1

Вспоминаем прошлое занятие

Вопрос: что такое нода кластера K8s?



Вспоминаем прошрое занятие

Вопрос: что такое нода кластера K8s?

Ответ: один из серверов, компьютеров или виртуальных машин, находящийся под единым управлением кластера, на котором запускаются контейнерные приложения

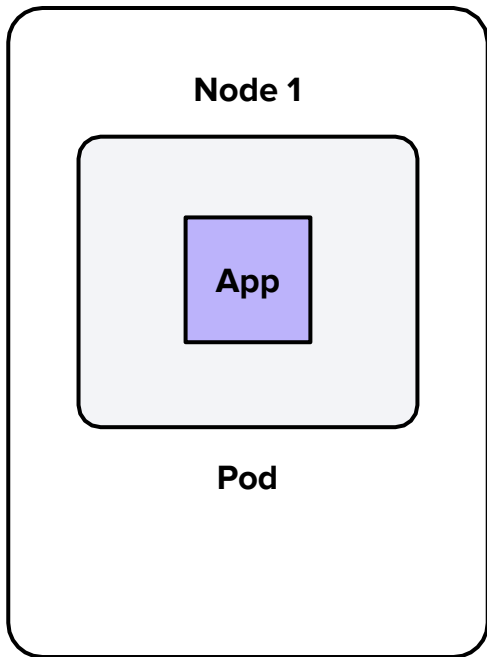




Pod — минимальная единица
развёртывания

Pod — минимальная единица развёртывания

Pod (под) — в Kubernetes самый маленький объект, в котором может быть запущена полезная нагрузка.

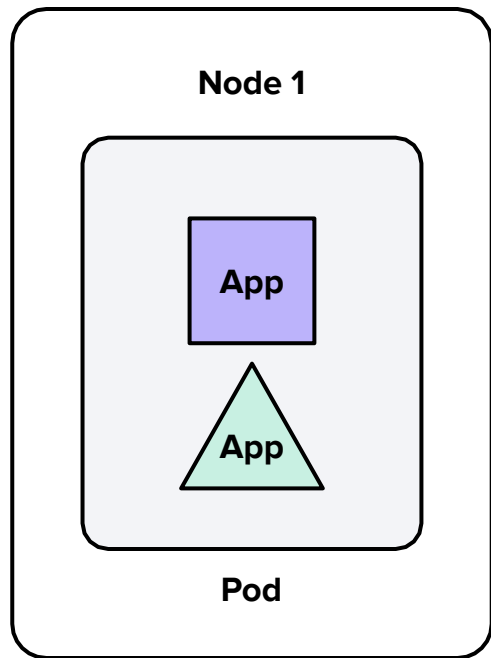


- Абстракция K8s над container runtime ноды



Pod — минимальная единица развёртывания

Pod (под) — в Kubernetes самый маленький объект, в котором может быть запущена полезная нагрузка.

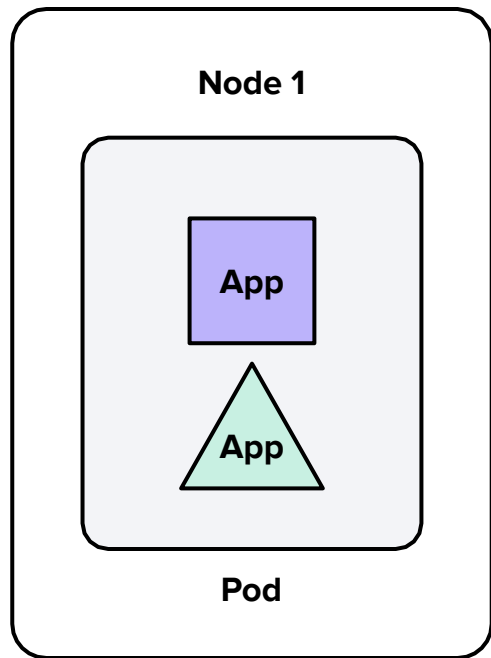


- Абстракция K8s над container runtime ноды
- Состоит из одного или нескольких контейнеров. Обычно помещают только один основной контейнер + вспомогательный (sidecar)



Pod — минимальная единица развёртывания

Pod (под) — в Kubernetes самый маленький объект, в котором может быть запущена полезная нагрузка.

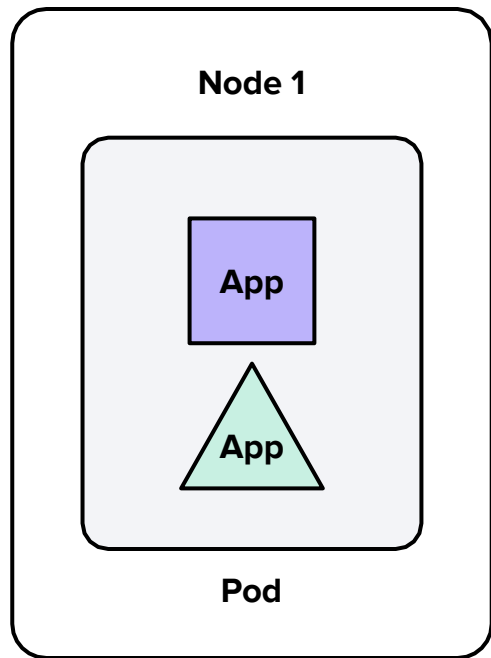


- Абстракция K8s над container runtime ноды
- Состоит из одного или нескольких контейнеров. Обычно помещают только один основной контейнер + вспомогательный (sidecar)
- Файловая система у контейнеров внутри пода разная. Можно шарить между контейнерами



Pod — минимальная единица развёртывания

Pod (под) — в Kubernetes самый маленький объект, в котором может быть запущена полезная нагрузка.



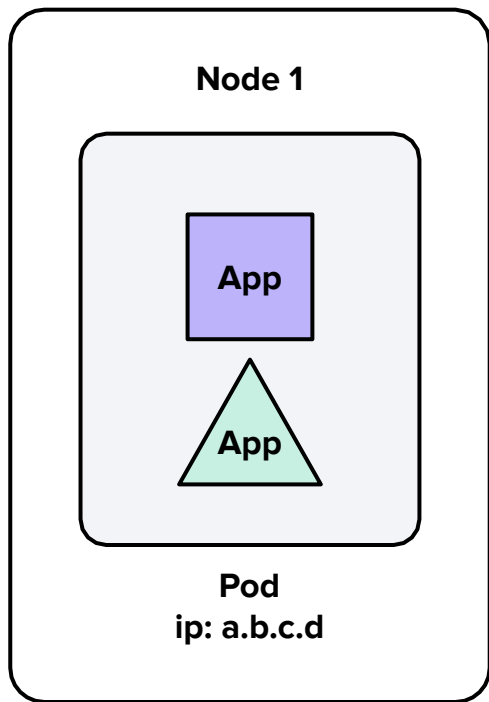
- Абстракция K8s над container runtime ноды
- Состоит из одного или нескольких контейнеров. Обычно помещают только один основной контейнер + вспомогательный (sidecar)
- Файловая система у контейнеров внутри пода разная. Можно шарить между контейнерами
- Контейнеры пода начинают запускаться одновременно

```
kubectl get pods
```



Pod — минимальная единица развёртывания

Сеть в поде

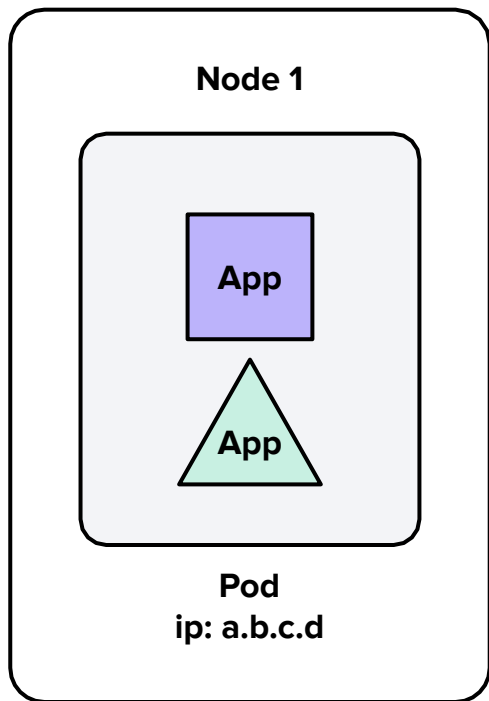


- У всех контейнеров внутри пода один и тот же IP-адрес



Pod — минимальная единица развёртывания

Сеть в поде

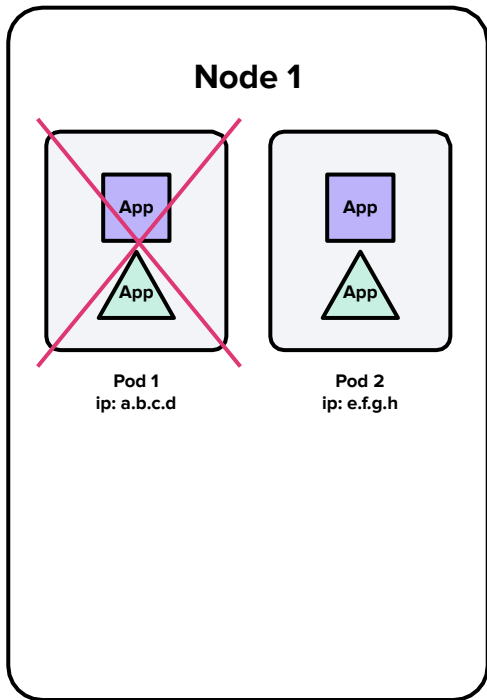


- У всех контейнеров внутри пода один и тот же IP-адрес
- Все контейнеры внутри видят друг друга как localhost



Pod — минимальная единица развёртывания

Сеть в поде



- У всех контейнеров внутри пода один и тот же IP-адрес
- Все контейнеры внутри видят друг друга как localhost
- После удаления и перезапуска — новый IP-адрес



Конфигурация объектов



2

Способы управления объектами K8s

Императивная конфигурация объекта:

```
kubectl run nginx --image nginx  
kubectl create deployment nginx --image nginx
```

Способы управления объектами K8s

Императивная конфигурация объекта:

```
kubectl run nginx --image nginx  
kubectl create deployment nginx --image nginx
```

Декларативная конфигурация объекта:

```
kubectl apply -f config_nginx.yaml
```

Файл конфигурации объекта

YAML-конфигурация объектов содержит секции:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: myapp
  name: pod-with-app
  namespace: default
spec:
  containers:
  - image: nginx:1.20
    imagePullPolicy: IfNotPresent
    name: nginx
  - image:
    praqma/network-multitool:alpine-extra
    imagePullPolicy: IfNotPresent
    name: multitool
    env:
    - name: HTTP_PORT
      value: "8080"
```

- объект

Файл конфигурации объекта

YAML-конфигурация объектов содержит секции:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: myapp
  name: pod-with-app
  namespace: default
spec:
  containers:
  - image: nginx:1.20
    imagePullPolicy: IfNotPresent
    name: nginx
  - image:
    praqma/network-multitool:alpine-extra
    imagePullPolicy: IfNotPresent
    name: multitool
  env:
  - name: HTTP_PORT
    value: "8080"
```

- объект
- metadata

Файл конфигурации объекта

YAML-конфигурация объектов содержит секции:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: myapp
  name: pod-with-app
  namespace: default
```

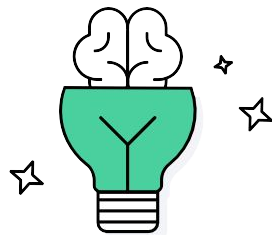
```
spec:
  containers:
  - image: nginx:1.20
    imagePullPolicy: IfNotPresent
    name: nginx
  - image:
    praqma/network-multitool:alpine-extra
    imagePullPolicy: IfNotPresent
    name: multitool
    env:
    - name: HTTP_PORT
      value: "8080"
```

- объект
- metadata
- specification

Labels Selector



3



**Labels (метки) — это пары
«ключ-значение», которые
добавляются к объектам для
идентификации**

Особенности Labels

- Labels могут быть присоединены к объектам во время их создания или добавлены позже в любое время
- Каждый объект может иметь набор из нескольких Labels
- Каждый ключ должен быть уникальным в рамках объекта
- Labels могут быть использованы при ручном поиске или в качестве селекторов

Особенности Labels

- Labels могут быть присоединены к объектам во время их создания или добавлены позже в любое время
- Каждый объект может иметь набор из нескольких Labels
- Каждый ключ должен быть уникальным в рамках объекта
- Labels могут быть использованы при ручном поиске или в качестве селекторов

```
kubectl label pod pod-with-app app=myapp
```

```
...  
metadata:  
  labels:  
    app: myapp  
...
```

Selector (селектор) —
средство группировки
в Kubernetes, с помощью
которого можно
идентифицировать набор
объектов

Особенности селекторов

Есть два типа селекторов:

- **на равенстве** — позволяют отфильтровывать объекты по ключам и значениям меток
- **на наборе** — фильтруют ключи в соответствии с набором значений

Особенности селекторов

Есть два типа селекторов:

- **на равенстве** — позволяют отфильтровывать объекты по ключам и значениям меток
- **на наборе** — фильтруют ключи в соответствии с набором значений

```
apiVersion: v1
kind: Service
metadata:
  name: pod-with-app-service
```

spec:

Selector:

app: myapp

ports:

- name: web
- port: 80

Service

4

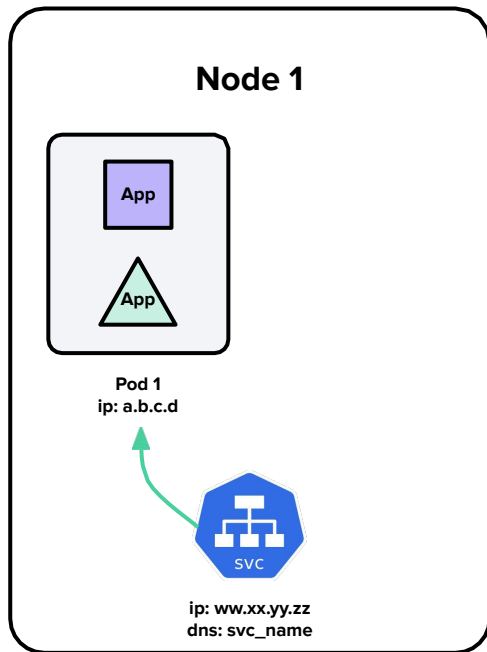


Service (сервис) — это объект K8s, который определяет логический набор подов и политику доступа к ним

Service обеспечивает сетевую связность

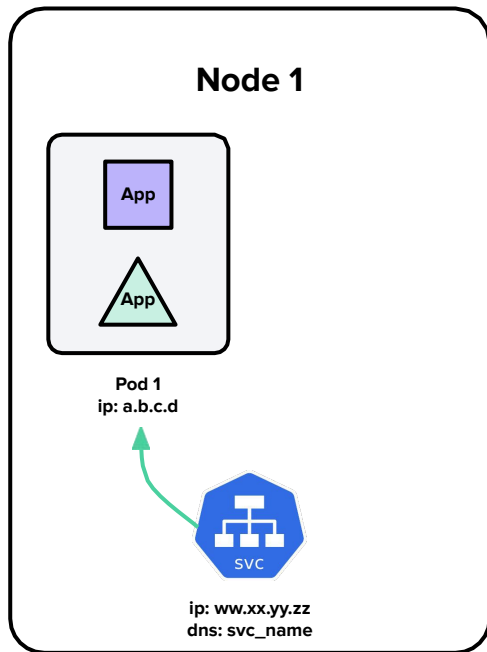
Service обладает постоянным IP-адресом и DNS-именем, которые можно связать с подом или подами.

- Service служит для маршрутизации сети в кластере

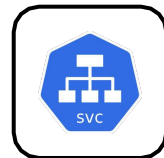


Service обеспечивает сетевую связность

Service обладает постоянным IP-адресом и DNS-именем, которые можно связать с подом или подами.

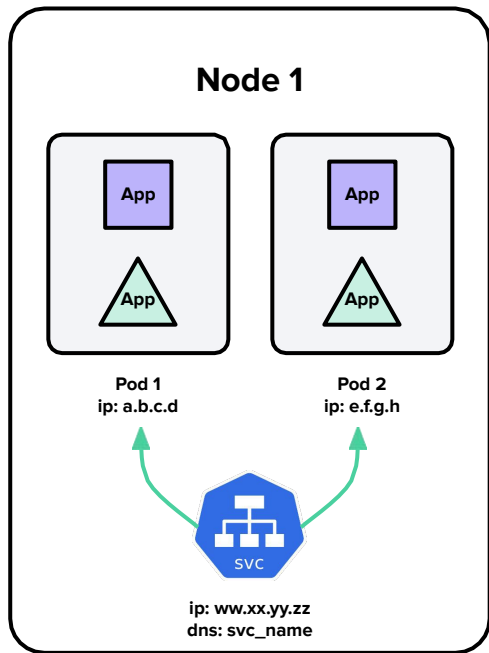


- Service служит для маршрутизации сети в кластере
- Можно обратиться с любой ноды

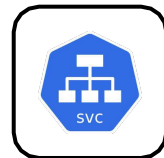


Service обеспечивает сетевую связность

Service обладает постоянным IP-адресом и DNS-именем, которые можно связать с подом или подами.

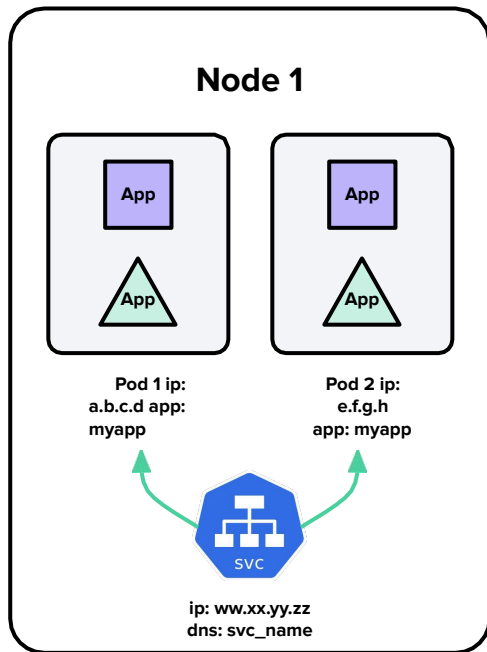


- Service служит для маршрутизации сети в кластере
- Можно обратиться с любой ноды
- При наличии нескольких реплик пода Service обеспечивает балансировку нагрузки



Service обеспечивает сетевую связность

Service обладает постоянным IP-адресом и DNS-именем, которые можно связать с подом или подами.

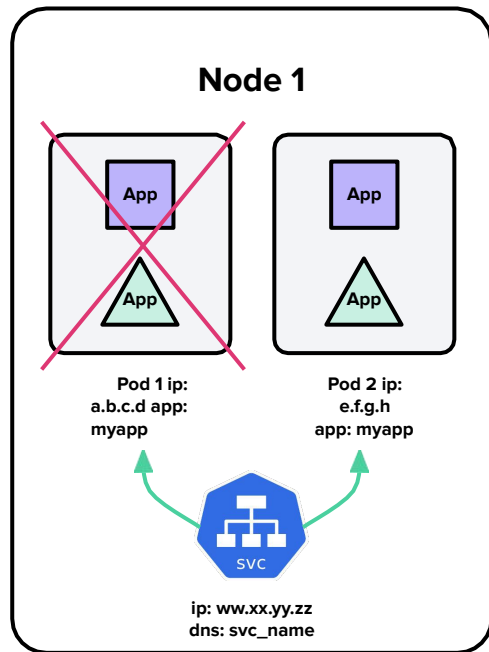


- Service служит для маршрутизации сети в кластере
- Можно обратиться с любой ноды
- При наличии нескольких реплик пода Service обеспечивает балансировку нагрузки
- Service выбирает поды для связки по меткам и селекторам



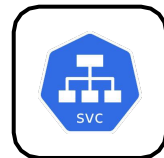
Service обеспечивает сетевую связность

Service обладает постоянным IP-адресом и DNS-именем, которые можно связать с подом или подами.



- Service служит для маршрутизации сети в кластере
- Можно обратиться с любой ноды
- При наличии нескольких реплик пода Service обеспечивает балансировку нагрузки
- Service выбирает поды для связки по меткам и селекторам
- Жизненные циклы пода и сервиса не зависят друг от друга

```
kubectl get svc
```



Файлы конфигурации Pod и Service

Пример конфигураций Pod и Service. Связь между ними с помощью Labels:

```
apiVersion: v1
kind: Service
metadata:
  name: main
spec:
  ports:
    - name: web
      port: 80
      protocol: TCP
      targetPort: 9376
  selector:
    app: myapp
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-app
  labels:
    app: myapp
spec:
  containers:
    - image: nginx:1.20
      name: web_proxy
```

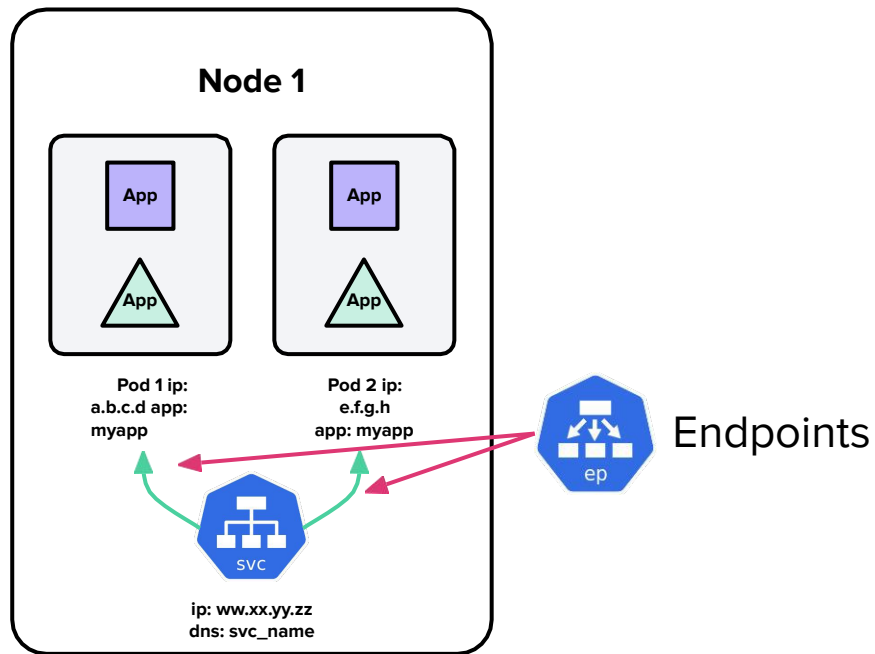
Endpoints



5

Endpoints объединяет сервис с подами

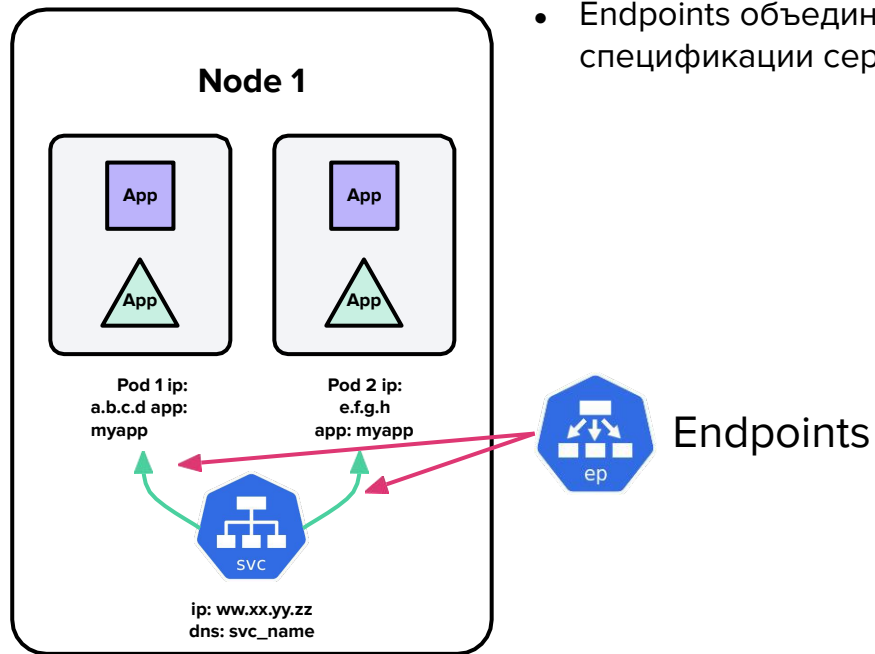
Цепочка событий выглядит так: Service —> Endpoints —> Pod



Endpoints объединяет сервис с подами

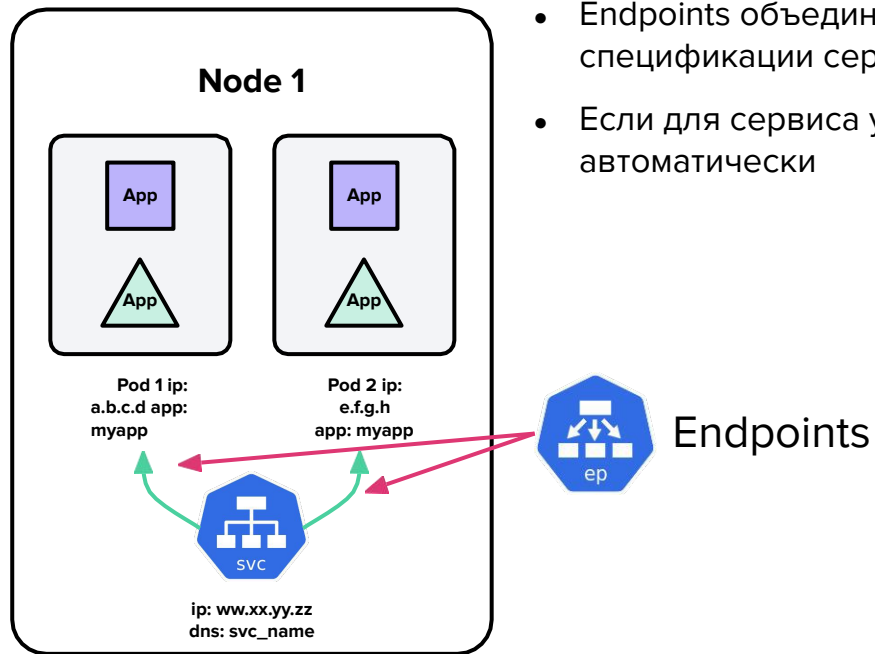
Цепочка событий выглядит так: Service —> Endpoints —> Pod

- Endpoints объединяет сервис с подами по селекторам, которые указаны в спецификации сервиса



Endpoints объединяет сервис с подами

Цепочка событий выглядит так: Service —> Endpoints —> Pod

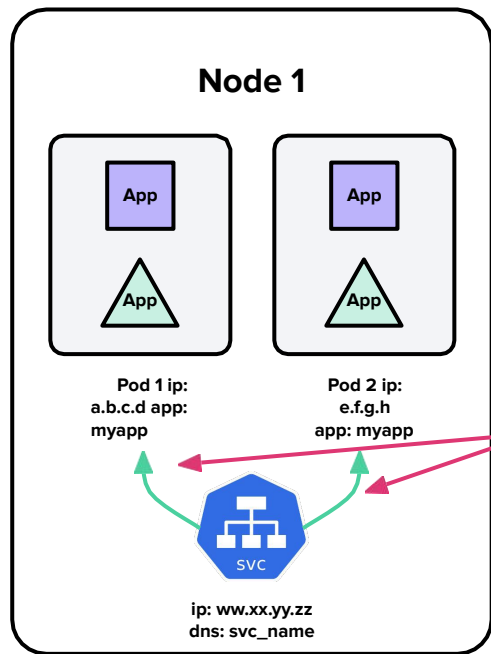


- Endpoints объединяет сервис с подами по селекторам, которые указаны в спецификации сервиса
- Если для сервиса указан селектор, то объект Endpoints будет создан автоматически



Endpoints объединяет сервис с подами

Цепочка событий выглядит так: Service —> Endpoints —> Pod



- Endpoints объединяет сервис с подами по селекторам, которые указаны в спецификации сервиса
- Если для сервиса указан селектор, то объект Endpoints будет создан автоматически
- Можно создать вручную, указав IP-адрес

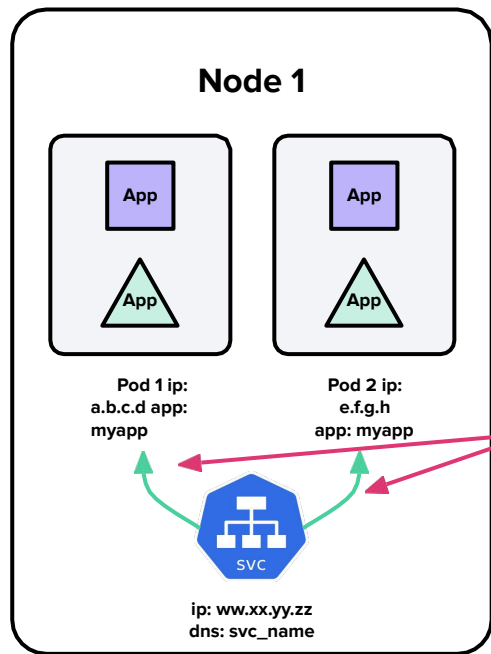


Endpoints



Endpoints объединяет сервис с подами

Цепочка событий выглядит так: Service —> Endpoints —> Pod

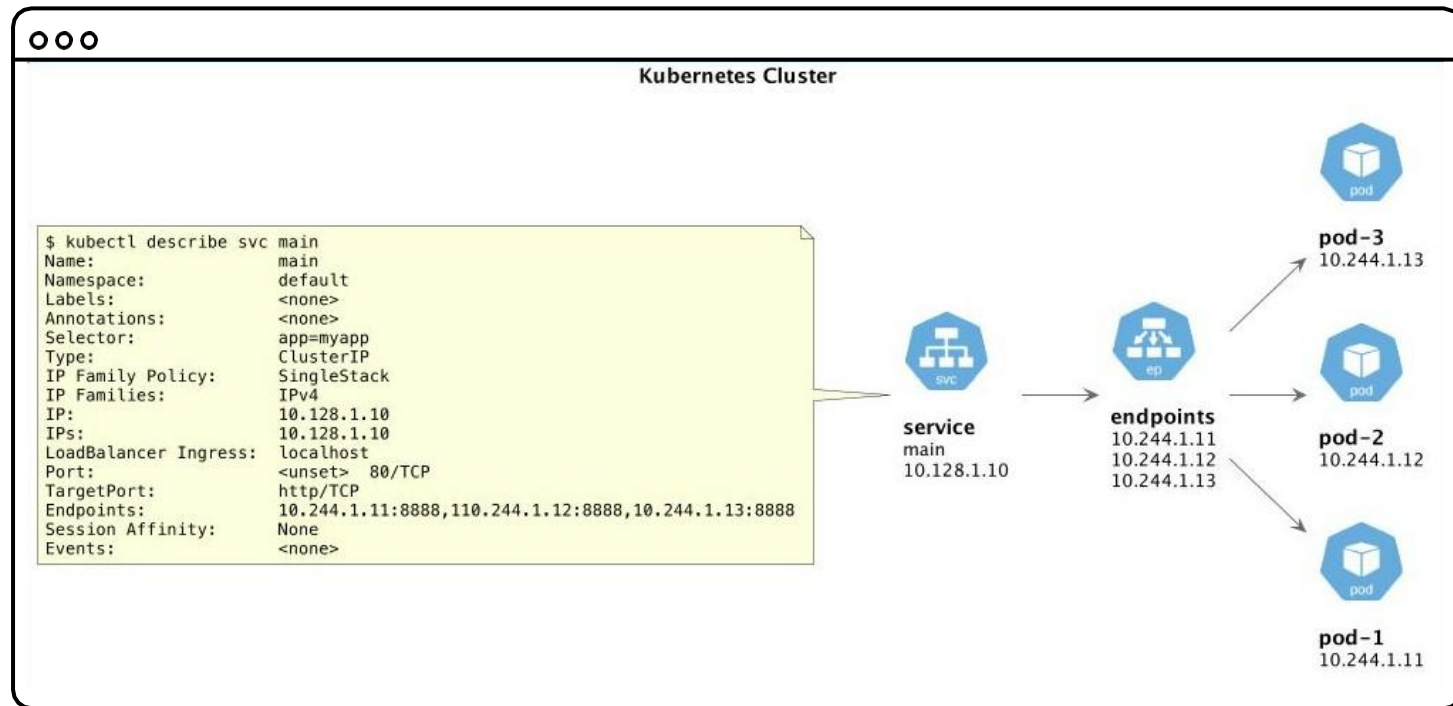


- Endpoints объединяет сервис с подами по селекторам, которые указаны в спецификации сервиса
- Если для сервиса указан селектор, то объект Endpoints будет создан автоматически
- Можно создать вручную, указав IP-адрес
- Имя сервиса и имя Endpoints должны совпадать



Схема взаимодействия подов

Цепочка событий выглядит так: Service —> Endpoints —> Pod



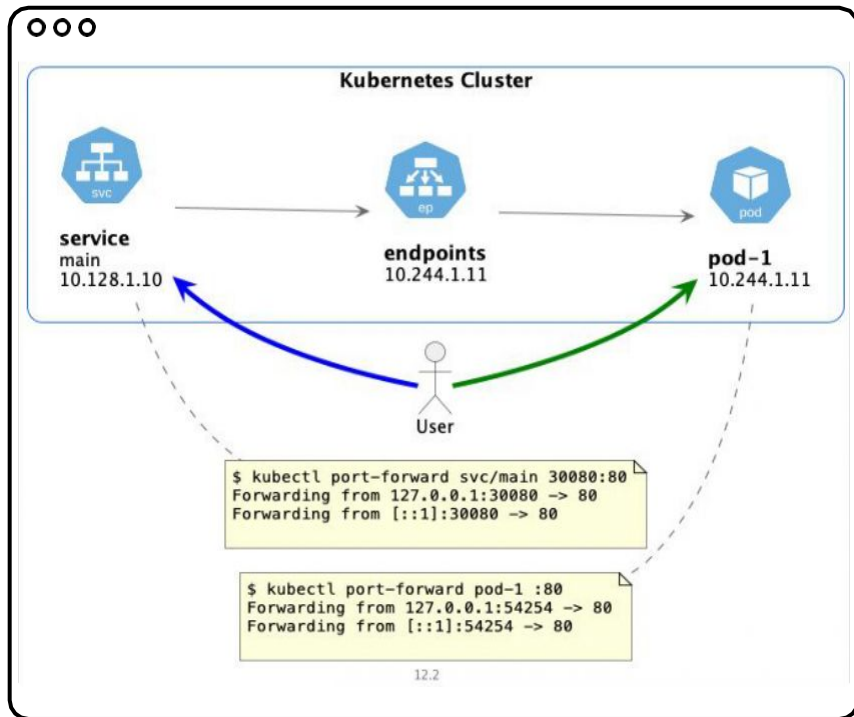
Подключение к Pod с помощью `kubectl` `port-forward`



6

Подключение с помощью kubectl port-forward

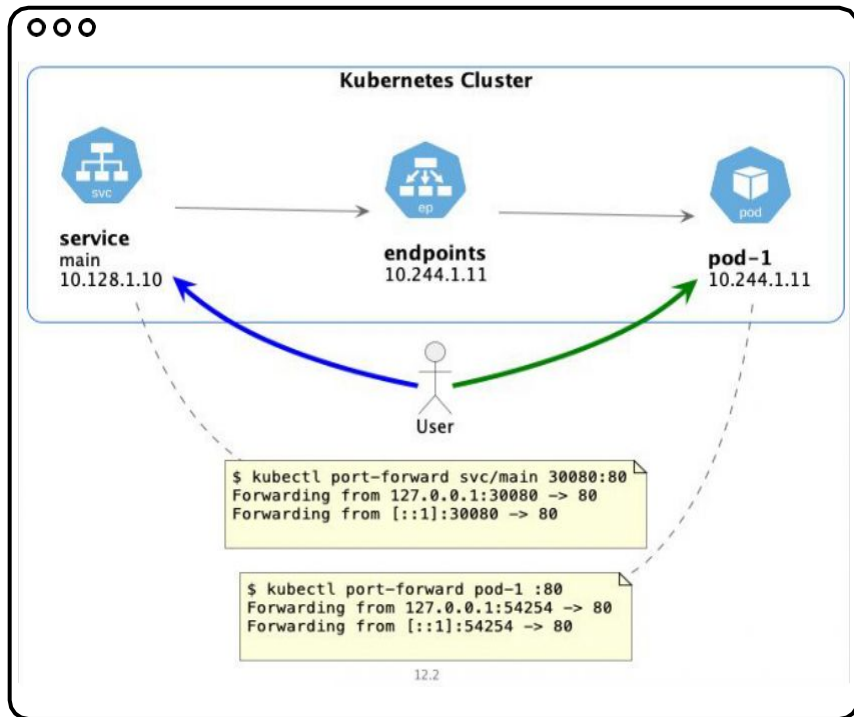
Нужно переадресовать один или несколько локальных портов в ресурс K8s



- Подключиться можно напрямую к поду, либо к сервису

Подключение с помощью kubectl port-forward

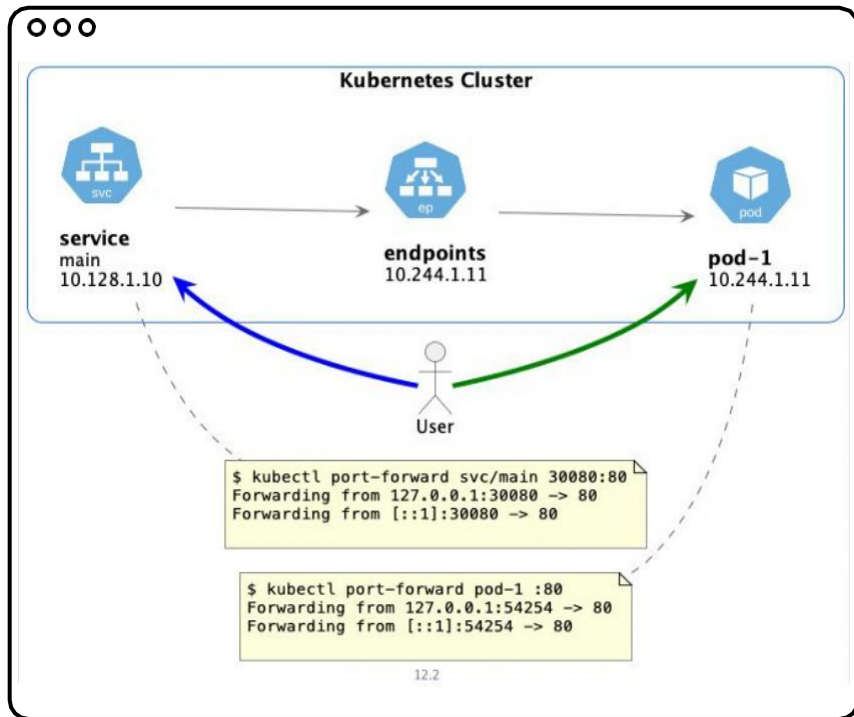
Нужно переадресовать один или несколько локальных портов в ресурс K8s



- Подключиться можно напрямую к поду, либо к сервису
- Команда kubectl port-forward не возвращает значение

Подключение с помощью kubectl port-forward

Нужно переадресовать один или несколько локальных портов в ресурс K8s

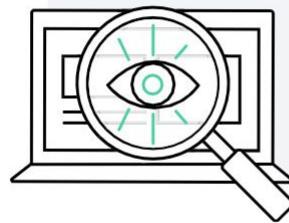


- Подключиться можно напрямую к поду, либо к сервису
- Команда kubectl port-forward не возвращает значение
- Команда может быть применена на разные ресурсы

kubectl port-forward

Демонстрация работы

Работа с объектами K8s



Итоги

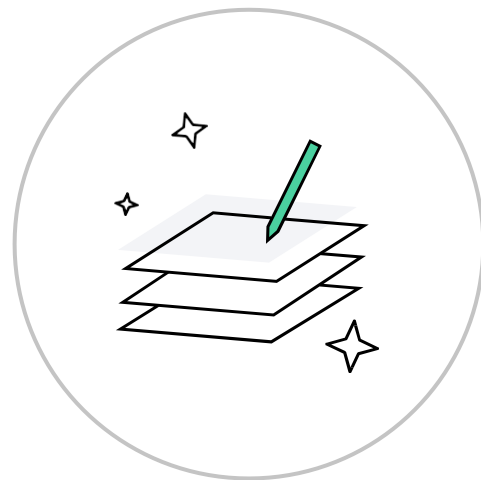
Сегодня мы

- 1 Узнали, что такое Pods, Labels, Services, Endpoints
- 2 Разобрались, как связать Pods и Services
- 3 Поняли, как получить доступ к Pod снаружи трафика
- 4 Рассмотрели варианты конфигурации объектов K8s
- 5 Подключились к кластеру, посмотрели поды, сервисы и подключились снаружи с помощью port-forward

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



Задавайте вопросы и пишите отзыв о лекции

Кирилл Касаткин
DevOps-инженер, Renue

