

Сетевое взаимодействие в K8s. Часть 2

Ingress

Кирилл Касаткин
DevOps-инженер, Renue



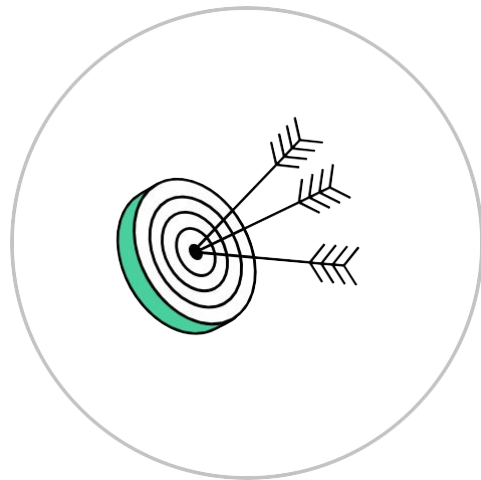
Кирилл Касаткин

DevOps-инженер, Renuе



Цели занятия

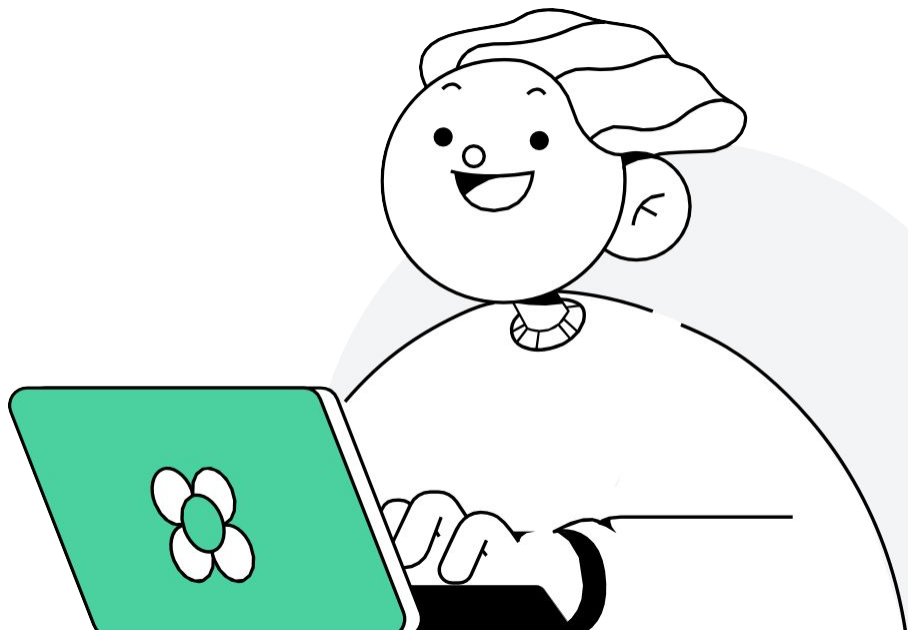
- Узнать, что такое Ingress и Ingress-контроллер
- Разобраться, как Ingress маршрутизирует трафик
- Познакомиться с примерами конфигураций Ingress
- Обсудить, какие виды Ingress-контроллеров существуют



План занятия

- 1 Ingress
- 2 Ingress-контроллер
- 3 Итоги
- 4 Домашнее задание

*Нажми на нужный раздел для перехода



Вспоминаем прошлые занятия

Вопрос: каким образом можно получить доступ к приложению извне?



Вспоминаем прошрое занятие

Вопрос: каким образом можно получить доступ к приложению извне?

Ответ: с помощью Service типа NodePort



Вспоминаем прошное занятие

Вопрос: на каком уровне модели OSI
работает Service?



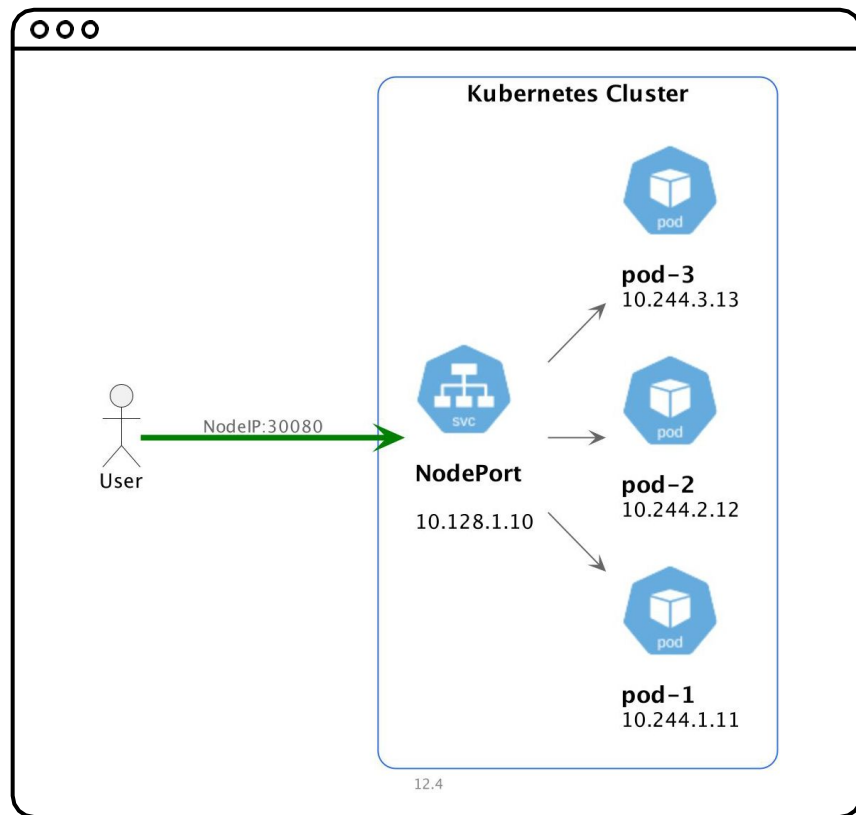
Вспоминаем прошное занятие

Вопрос: на каком уровне модели OSI работает Service?

Ответ: на L4-уровне (protocol:port)



Вспоминаем прошлое занятие



Ingress



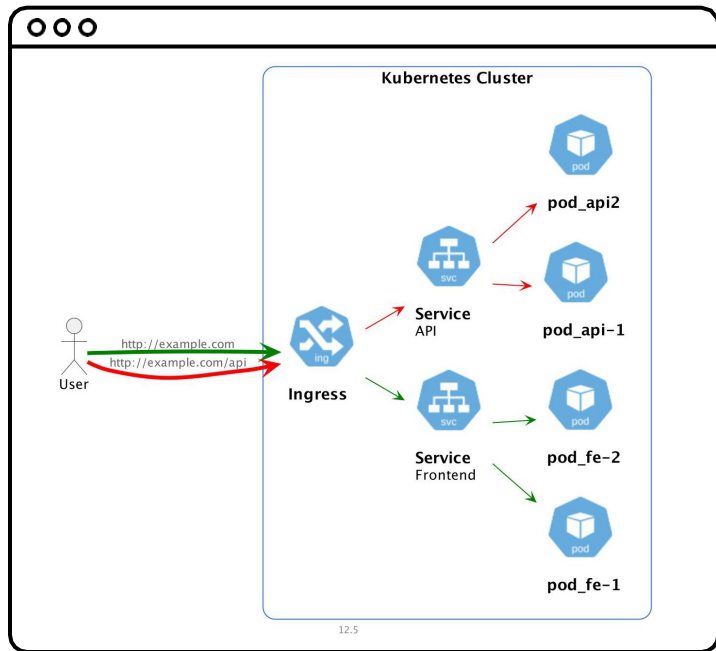
1



**Ingress — объект K8s,
который обеспечивает
внешний доступ к сервисам
на 7 уровне (HTTP/HTTPS)**

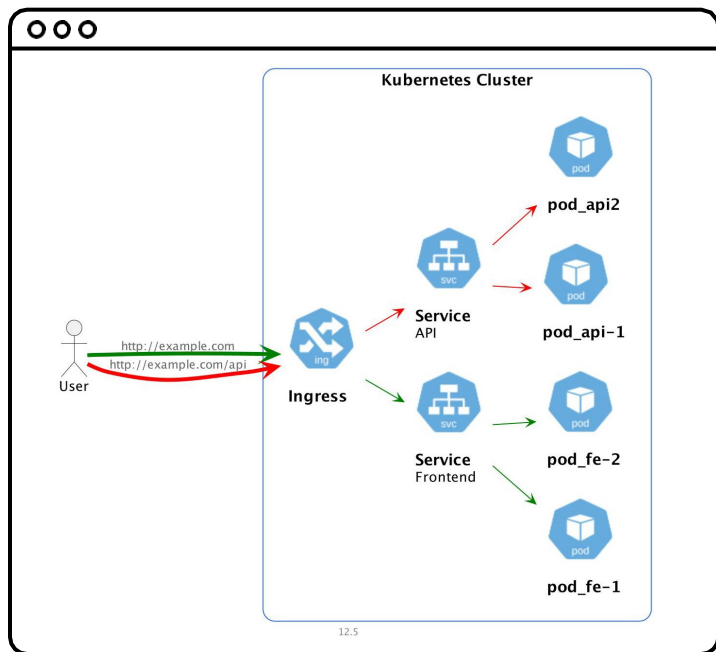
Ingress маршрутизирует трафик

Ingress консолидирует правила маршрутизации трафика в одном месте, что позволяет использовать несколько сервисов под одним IP-адресом



Ingress маршрутизирует трафик

Ingress консолидирует правила маршрутизации трафика в одном месте, что позволяет использовать несколько сервисов под одним IP-адресом

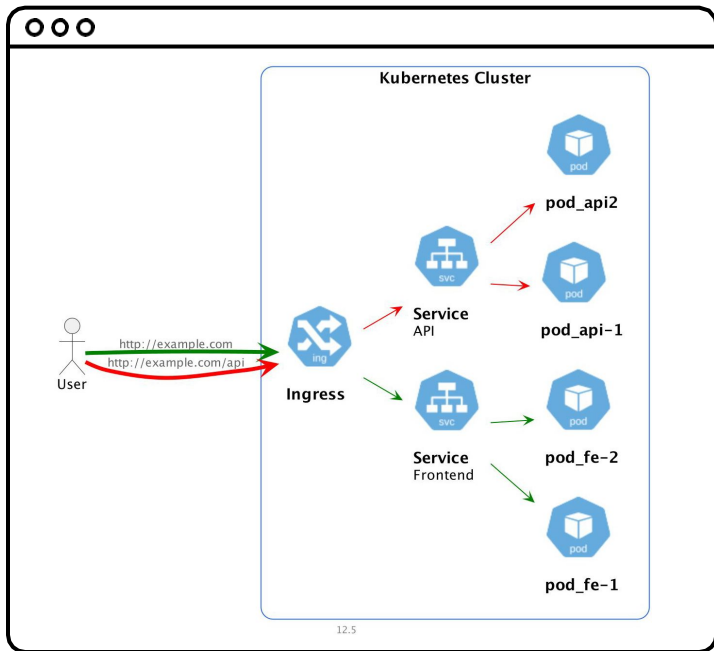


- Ingress определяет правила маршрутизации трафика — Rule



Ingress маршрутизирует трафик

Ingress консолидирует правила маршрутизации трафика в одном месте, что позволяет использовать несколько сервисов под одним IP-адресом

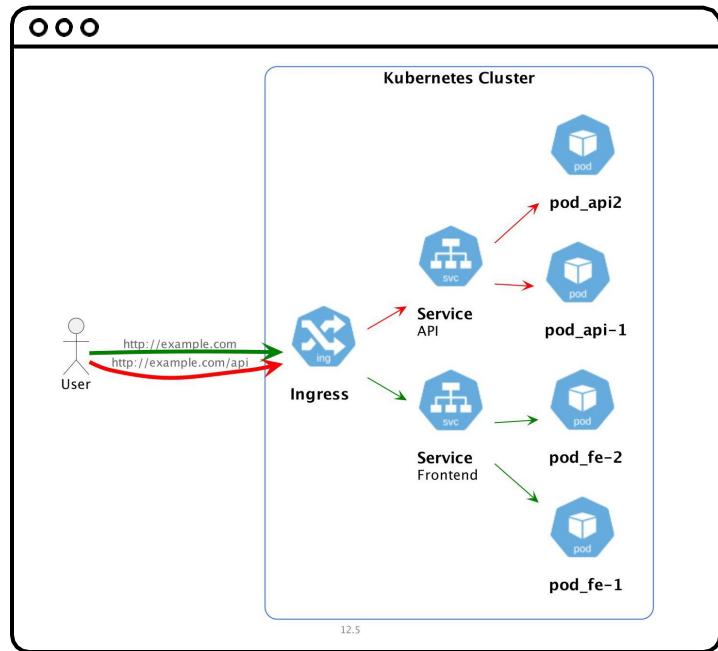


- Ingress определяет правила маршрутизации трафика — Rule
- Каждое Rule определяет Path со своим бэкендом, указывающим на Service. Запросы с определёнными Path будут направлены в соответствующие Service



Ingress маршрутизирует трафик

Ingress консолидирует правила маршрутизации трафика в одном месте, что позволяет использовать несколько сервисов под одним IP-адресом



- Ingress определяет правила маршрутизации трафика — Rule
- Каждое Rule определяет Path со своим бэкендом, указывающим на Service. Запросы с определёнными Path будут направлены в соответствующие Service
- Можно ссылаться на имя порта вместо номера, если он указан в настройке сервиса



Пример конфигурации

Примеры конфигураций Ingress HTTP

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: my-app-service
            port:
              number: 80
      - path: /api
        backend:
          service:
            name: my-app-api
            port:
              number: 80
```



Пример конфигурации

Примеры конфигураций Ingress HTTP

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: my-app-service
            port:
              number: 80
      - path: /api
        backend:
          service:
            name: my-app-api
            port:
              number: 80
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - backend:
          service:
            name: my-app-service
            port:
              number: 80
      - host: api.my-app.com
        http:
          paths:
          - backend:
              service:
                name: my-app-api
                port:
                  number: 80
```



Пример конфигурации

Примеры конфигураций Ingress TLS

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: my-app-service
            port:
              number: 80
  tls:
  - hosts:
    - my-app.com
    secretName: my-app-secret-tls
```



Пример конфигурации

Примеры конфигураций Ingress TLS

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: my-app.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: my-app-service
            port:
              number: 80
  tls:
  - hosts:
    - my-app.com
    secretName: my-app-secret-tls
```

```
apiVersion: v1
kind: Secret
metadata:
  name: my-app-secret-tls
data:
  tls.crt: base64 encode cert
  tls.key: base64 encode key
type: kubernetes.io/tls
```



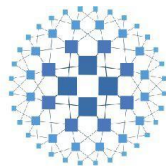
Ingress-контроллер



2

**Ingress-контроллер — под
(набор подов) с запущенным
приложением-контроллером,
выполняющим правила
Ingress**

Ingress-контроллеры бывают разные



HAPROXY



CONTOUR

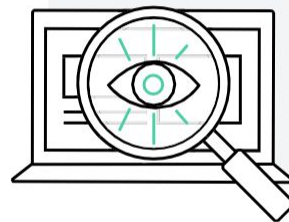


Kong



Демонстрация работы

Конфигурация Ingress с несколькими правилами маршрутизации.
Настройка TLS, Nginx Ingress Controller



Итоги/Выводы/Вы узнали/научились

Сегодня мы:

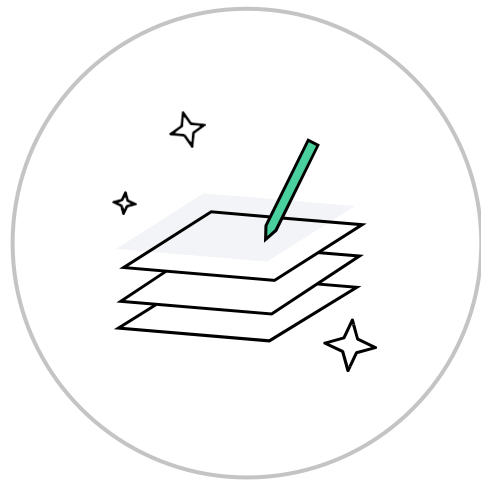
- 1 Узнали, что такое Ingress и Ingress-контроллер.
Рассмотрели несколько контроллеров: какие бывают, чем отличаются
- 2 Разобрались, как происходит маршрутизация трафика,
как конфигурируется TLS для HTTPS
- 3 Рассмотрели примеры манифестов объектов K8s
- 4 Попробовали подключиться к кластеру и посмотреть в работе объекты,
которые изучили на занятии



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

[Сравнительная таблица](#) разных Ingress-контроллеров



**Задавайте вопросы
и пишите отзыв о лекции**

