K8S 32: Kubernetes. Advanced. Ingress controller

Описание:

Данный контроллер позволяет обойтись без NodePort и выставить приложение на легко запоминаемый URL. Также дополнительно предоставляет TLS, virtual-name хостинг, subpath маршрутизацию к приложению, расширенные правила обработки трафика. Существует большое количество реализаций Ingress Controller, которые под капотом, как правило, используют открытые решения, начиная от nginx и HAproxy, заканчивая более близкими к контейнерам Traefik и Envoy. Кроме того, облачные поставщики еще создают свои Ingress controllers, управляющие их облачными решениями (к примеру, в случае Google - управляет Google Load Balancer). Мы же в рамках этого задания будем рассматривать преимущественно Nginx Ingress Controller, как доступный в Minikube и богатый на возможности.
Важно разделять 2 сущности в Kubernetes - Ingress и Ingress Controller.

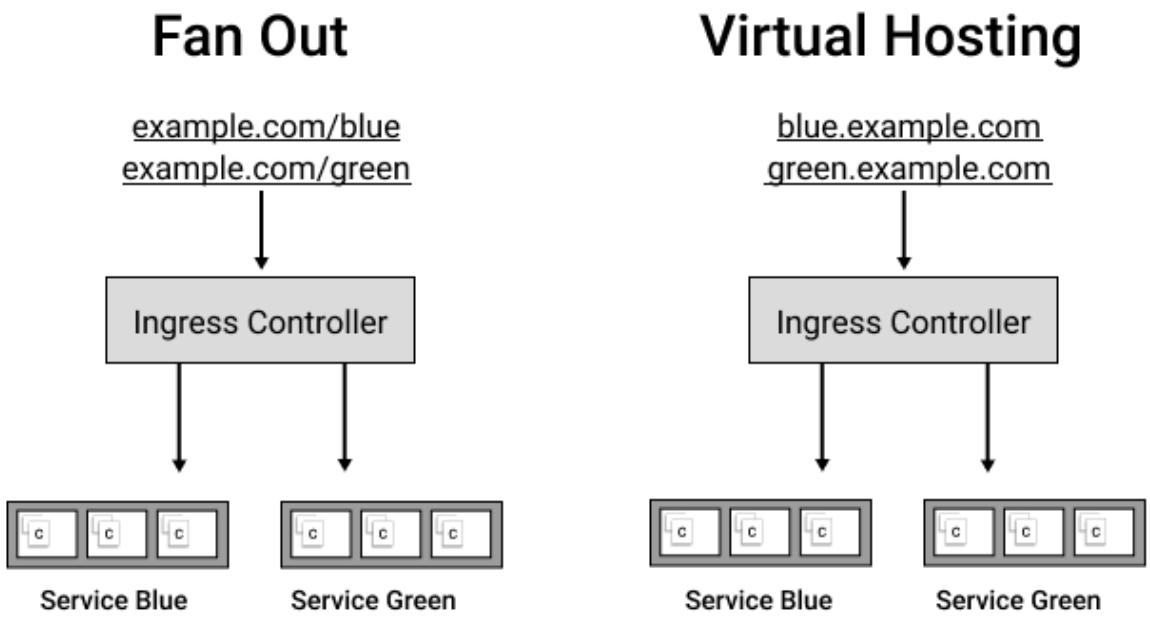
- Ingress Controller — это приложение, которое обрабатывает трафик, согласно манифестам (как правило, генерируя по шаблонам конфиги для определенного приложения).
- Ingress — как раз тот манифест, который описывает, как должен обрабатываться трафик и в какие сервисы запросы должны проксироваться.

Итак, начнем наше знакомство в Ingress Controller.
Для начала работы нам необходимо включить Ingress Controller. В этом нам поможет документация по ссылке enable-the-ingress-controller.
Создадим несколько Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: web-ingress
  namespace: default
spec:
  rules:
  - host: blue.example.com
    http:
      paths:
      - backend:
          serviceName: webserver-blue-svc
          servicePort: 80
  - host: green.example.com
```

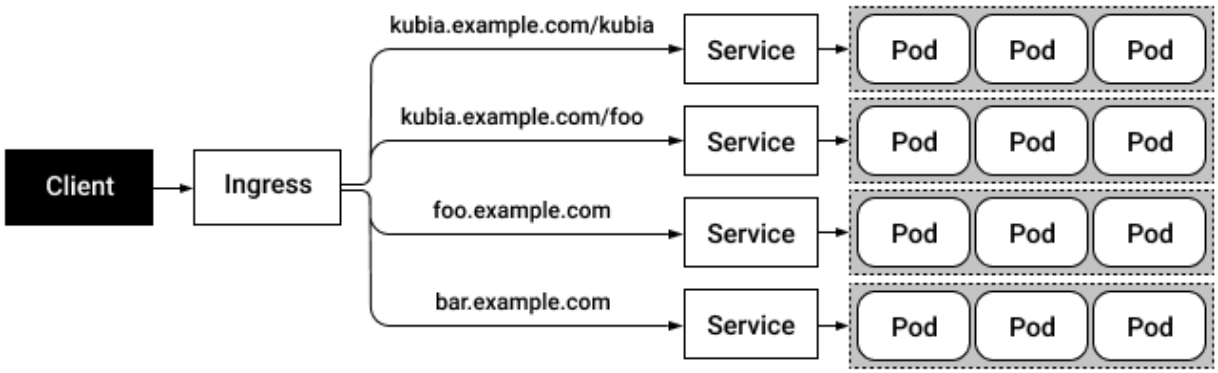
```
    http:
      paths:
      - backend:
          serviceName: webserver-green-svc
          servicePort: 80
```

В текущем виде мы сделали host based virtual hosting, но возможен и другой пример.



Так как мы используем minikube, нужно добавить записи в /etc/hosts.
$(minikube ip) blue.example.com green.example.com

Конечно, при открытии в браузере данных страниц получим 404. Это из-за отсутствия указанных сервисов и endpoints к ним. Запустите pod с nginx, свяжите с ними сервисы blue и green и проверьте доступность.

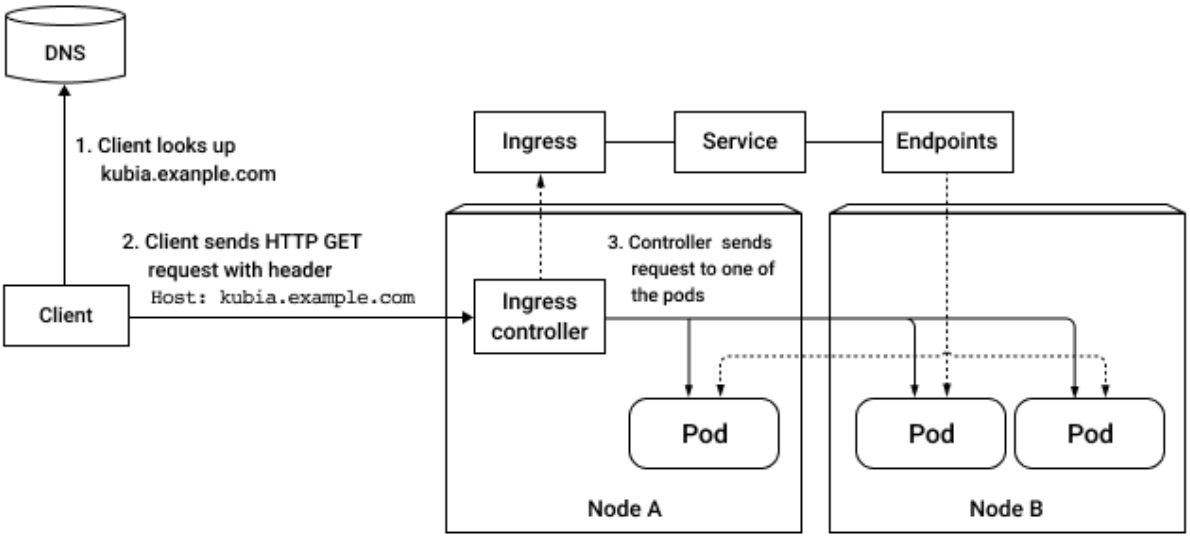


Пример описания ingress для subpath

```
apiVersion: extensions/v1beta1
```

```yaml
kind: Ingress
metadata:
  name: my-service-ingress
spec:
  rules:
  - host: my.hostname.com
    http:
      paths:
      - path: /my-service
        backend:
          serviceName: my-service
          servicePort: 80
```

В этом примере сервис будет доступен пользователю по адресу my.hostname.com/my-service. Ingress зависит от service, который, в свою очередь, зависит от endpoints.



Некоторые реализации ingress предоставляют дополнительные возможности. Например, такие как автоматическая настройка TLS, авторизация, дополнительные заголовки. Рассмотрим реализацию ingress на nginx. В примере ниже в директиве tls мы добавляем сертификат. Это практически также, как в nginx при помощи семейства директив ssl*. Для того чтобы это корректно заработало, предварительно нужно создать секрет с сертификатом и ключом.

```
kubectl create secret tls tls-secret --cert=tls.cert --key=tls.key
```

Затем указать данный сертификат при создании ingress объекта.

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
```

```
metadata:
  name: kubia
spec:
  tls:
  - hosts:
  - myhost.example.com
  secretName: tls-secret
  rules:
  - host: myhost.example.com
  http:
    paths:
    - path: /
      backend:
        serviceName: my-service
        servicePort: 80
```

Можете проверить доступность сервиса через 80 и 443 порт вашего ingress. Множество дополнительных настроек конфигурируется через аннотации ресурса annotations.md. Nginx ingress может взять на себя авторизацию пользователей. Поддерживается множество решений. От basic-auth до oath. Примеры доступны по ссылке - examples/auth.

Полезные ссылки:

- Ingress Controllers (official docs)
- Kubernetes Ingress (comparison)
- Ingress (official docs)

Задание:

1. Запустите манифесты из задания про StatefulSet в вашем Minikube-кластере, если он еще не запущен.
2. Установите Nginx Ingress контроллер в вашем кластере (команды и вывод сохраните).
3. При помощи curl -i обратитесь к ingress controller - должна вернуться страница с кодом 404 (команды и вывод сохраните).
4. Напишите манифест для Ingress, ведущий на StatefulSet, порт 15672 с именем rbmk8s31ingress.
5. Примените манифест, после чего выведите его описание через describe (команды и вывод сохраните).
6. При помощи curl -i обратитесь к ingress controller (команды и вывод сохраните).

7. Измените манифест так, чтобы он начал слушать на домене, созданном при помощи nip.io.
8. Примените манифест, после чего выведите его описание через describe (команды и вывод сохраните).
9. При помощи curl -i обратитесь к ingress controller по доменному имени (команды и вывод сохраните).
10. На проверку отправьте написанный манифест, сохраненные команды и их выводы.