KUB 24: Получение сертификатов с помощью letsencrypt

Описание:

Вернемся на одно задание назад к теме Ingress, поскольку теперь у нас есть кластер с внешним IP-адресом, и мы можем разобрать еще один очень полезный инструмент.
Cert-manager — это демон для Kubernetes Cluster (даже точнее — для Ingress), который дает возможность в автоматическом режиме получать TLS-сертификаты для ваших Ingress и хранить их в системе в виде Secret, а не создавать их вручную.
Перед тем как начать работу с такой утилитой, разберем архитектуру приложения:
Cert-manager оставляет свои сущности в Custom Resource Definitions:

- issuers — данные, которые используются для получения сертификатов — определяют, откуда получать сертификат (cert-manager поддерживает не только Let's Encrypt, но и других поставщиков) и с какими данными. Применяется на уровне Namespace.
- clusterissuers — практически то же самое, что и issuers, но применяется на весь кластер.
- orders — на время работы хранит здесь все запросы, которые требуется выполнить для получения сертификата.
- certificaterequests — во время получения сертификата сохраняет запросы на сертификаты, которые отправляются в сторону провайдера с приватным ключом.
- challenges — после получения challenge со стороны поставщика сертификата сохраняет его здесь для создания Challenge Service.
- certificates — хранит данные о сертификатах — когда они выданы, в каком секрете хранятся и для какого Ingress создавались.

Сертификат, в котором находятся сертификат и приватный ключ, согласно требованию Ingress, хранится в Secret, указанном в манифесте Ingress.
Как понятно, самый распространенный кейс использования Cert-manager — автоматизация получения сертификата от Let's Encrypt. И для этого в Cert-manager реализовано несколько методов получения — HTTP и DNS (для ряда провайдеров, которых поддерживает и Certbot).
Все параметры использования Cert-manager записываются в манифестах Ingress через аннотации, позволяя узнать, управляется ли сертификат через Cert-manager и какой Issuer или ClusterIssuer используются для конкретного Ingress. Именно эти аннотации и вылавливает Cert-manager, подключаясь к Kubernetes Events API.
Обновлением сертификатов Cert-manager управляет сам, выставляя себе метки, когда важно их обновлять (за сколько дней до окончания сертификата можно выставлять в параметрах запуска Cert-manager).
Для установки достаточно применить один манифест:

```
$ kubectl apply -f
https://github.com/jetstack/cert-manager/releases/download/v1.1.
0/cert-manager.yaml
```

После чего будет необходимо создать новый clusterIssuer, который будет выписывать сертификаты через letsencrypt:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt
spec:
  acme:
    email: ov@fevlake.com
    privateKeySecretRef:
      name: letsencrypt-private-key
    server: https://acme-v02.api.letsencrypt.org/directory
    solvers:
    - http01:
       ingress:
         class: nginx
```

В данном случае мы указываем ingress class nginx, который мы установили несколько заданий назад. После этого можно подредактировать наш ingress до следующего вида:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: back
  namespace: default
  annotations:
    nginx.ingress.kubernetes.io/enable-cors: "true"
    nginx.ingress.kubernetes.io/auth-type: "basic"
    nginx.ingress.kubernetes.io/auth-secret: "basic-auth"
    nginx.ingress.kubernetes.io/auth-secret-type: "auth-file"
    # Указываем, каким образом выписывать сертификат
    cert-manager.io/cluster-issuer: "letsencrypt"
spec:
  rules:
  - host: back.dev.kis.im
    http:
      paths:
      - path: /
```

```
        backend:
          serviceName: nginx-back
          servicePort: 80
  # Указываем настройки для tls — для какого хоста нужен tls и
куда сохранить полученный сертификат
  tls:
  - hosts:
    - back.dev.kis.im
    secretName: back-dev-kis-im-cert
```

Собственно, на этом все — через некоторое время cert-manager получит сертификат для вашего домена!

Полезные ссылки:

- cert-manager official docs
- JetStack/Cert-Manager (github)

Задание:

1. Установите nginx-ingress контроллер в namespace ingress-nginx
2. Проверьте какой IP адрес получил service LoadBalancer в namespace ingress-nginx
3. Создайте DNS запись: ${domain name}, которая указывает на IP адрес из 2ого шага.
4. Создайте deployment с nginx nginx-dp в namespace default.
5. Создайте сервис для nginx-dp с именем svc-internal в namespace default с типом ClusterIP.
6. Установите cert-manager controller в namespace cert-manager.
7. Создайте ingress nginx-ingress в namespace default для доменного имени ${domain name}.
8. Получите валидный ssl сертификат от letsencrypt для вашего домена.
9. Отправьте задание на проверку.