

KUB 23: Аутентификация в Ingress (basic auth)

Описание:

Nginx ingress сможет взять на себя авторизацию пользователей. Поддерживается много решений. От basic-auth до oauth. Примеры доступны по ссылке — [examples/auth](#). Ранее мы с вами уже говорили, что такое аннотации, но лишь в одном слове обсудили, что аннотации используются для получения сервисной информации некоторыми сервисами, упомянув ingress и helm.

Пока мы подружались только с Ingress, так что давайте узнаем, как он связан с аннотациями. Но в разрезе Helm мы еще обязательно вернемся к этой теме. Итак, аннотации в Ingress используются для передачи сервисной информации, связанной с работой данных сервисов, компонентам кластера. Так, логично предположить, что как раз управление Ingress Controller и производится через аннотации в Ingress.

Если брать конкретно Nginx Ingress Controller, то он поддерживает большое количество опций — начиная от вставок кусков конфигов до вставок кода на Lua (да, в nginx ingress controller он есть).

Устанавливаются они согласно всем правилам обычных аннотаций — в блоке metadata. К примеру:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/server-snippet: |
      set $agentflag 0;

      if ($http_user_agent ~* "(Mobile)" ){
        set $agentflag 1;
      }

      if ( $agentflag = 1 ) {
        return 301 https://m.example.com;
      }
spec:
  rules:
  - host: example.com
    http:
      paths:
```

```
- path: /  
  backend:  
    serviceName: app  
    servicePort: 80
```

Как видно, все аннотации, которые связаны конкретно с Nginx Ingress Controller, имеют префикс `nginx.ingress.kubernetes.io/` (не путать с `nginx.org/` — это другой контроллер, созданный именно компанией Nginx, а не Kubernetes).

Как же внутри работают аннотации? На самом деле, как и все в Kubernetes — по подписке на события. То есть, при запуске Ingress controller подключается к Kubernetes API и запрашивает все Ingress и свою конфигурацию. И уже на базе такого генерирует конфигурационный файл. А далее подключается к Event API и на каждое изменение Ingress по новой генерирует конфиг и пытается применить конфигурацию.

Давайте попробуем добавить конфигурацию, которая обеспечивала бы нам аутентификацию пользователей. Для этого нам придется создать secret, который будет содержать в себе необходимую информацию и добавить несколько annotations к ingress ресурсу:

```
$ htpasswd -c auth vozerov
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user vozerov
```

```
$ kubectl create secret generic basic-auth --from-file=auth
```

```
secret/basic-auth created
```

```
$ kubectl get secret basic-auth -o yaml
```

```
apiVersion: v1
```

```
data:
```

```
  auth:
```

```
dm96ZXJvdjokYXB5MSRUb2VWRy9TLyRlM2dkUzZDdFFKWTltMk9lNzFqWXcwCg==
```

```
... skipped ...
```

Теперь подредактируем наш ingress:

```
apiVersion: extensions/v1beta1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: web-ingress
```

```
  namespace: default
```

```
  annotations:
```

```
    nginx.ingress.kubernetes.io/auth-type: basic
```

```
    nginx.ingress.kubernetes.io/auth-secret: basic-auth
```

```
    nginx.ingress.kubernetes.io/auth-realm: 'Authentication
Required'
spec:
  rules:
  - host: blue.example.com
    http:
      paths:
      - backend:
          serviceName: nginx-svc
          servicePort: 80
```

И попробуем обратиться к нашему сервису:

```
$ curl -D - -s -o /dev/null -H 'Host: blue.example.com'
http://84.201.133.236/
HTTP/1.1 401 Unauthorized
Date: Fri, 04 Dec 2020 10:20:00 GMT
Content-Type: text/html
Content-Length: 172
Connection: keep-alive
WWW-Authenticate: Basic realm="Authentication Required"
```

Как вы можете увидеть — нам вернулся ответ Unauthorized, давайте попробуем передать логин/пароль при запросе:

```
$ curl -u vozerov:test -D - -s -o /dev/null -H 'Host:
blue.example.com' http://84.201.133.236/
HTTP/1.1 200 OK
Date: Fri, 04 Dec 2020 10:20:05 GMT
Content-Type: text/html
Content-Length: 612
Connection: keep-alive
Last-Modified: Tue, 24 Nov 2020 13:02:03 GMT
ETag: "5fbd044b-264"
Accept-Ranges: bytes
```

Теперь все заработало, как надо!

Полезные ссылки:

- [Nginx ingress controller -- Annotations \(official docs\)](#)

Создание DNS записи в cloudflare через API

Для создание DNS записи типа A в cloudflare вы можете использовать curl, пример которого приведен ниже:

```
curl -X POST
"https://api.cloudflare.com/client/v4/zones/9152ec3c08b1a4faeaa9
5353a929fcc5/dns_records" -H "Authorization: _Bearer_token_" -H
"Content-Type:application/json" --data
'{"type":"A","name":"${domain name}","content":"${ip
address}","proxied":false}'
```

В параметре data надо заменить `${domain name}` на параметр, полученный при создании окружения. Так же необходимо заменить параметр `${ip address}` на адрес, полученный на 2 шаге.

В ответ на эту команду вы получите уникальный идентификатор записи:

```
{"result":{"id":"f3e4808896173309cbe80254e37225b7","zone_id":"91
52ec3c08b1a4faeaa95353a929fcc5","zone_name":"rbr-kubernetes.com"
,"name":"${domain name}","type":"A","content":"${ip
address}","proxiable":false,"proxied":false,"ttl":1,"locked":fal
se,"meta":{"auto_added":false,"managed_by_apps":false,"managed_b
y_argo_tunnel":false,"source":"primary"},"created_on":"2021-04-2
0T13:58:35.180297Z","modified_on":"2021-04-20T13:58:35.180297Z"}
,"success":true,"errors":[],"messages":[]}
```

В данном случае id - `f3e4808896173309cbe80254e37225b7`. Он потребуется, если вам придется обновить dns запись на новый адрес (например при передаче задания). Для обновления записи воспользуйтесь следующей командой:

```
curl -X PUT
"https://api.cloudflare.com/client/v4/zones/9152ec3c08b1a4faeaa9
5353a929fcc5/dns_records/${dns id}" -H "Authorization: Bearer
_bearer_token_" -H "Content-Type:application/json" --data
'{"type":"A","name":"${domain name}","content":"${ip
address}","proxied":false}'
```

Вместо переменной `${dns id}` - подставьте свой идентификатор, который вы получили при создании записи.

После создания записи вы можете проверить созданную запись с помощью dig:

```
dig ${domain name}
```

Если в ответе отображается IP адрес все отлично! Напоминаем, что при изменении записи должно пройти некоторое время для обновления информации в DNS серверах.

После завершения задания запись можно не удалять - мы удалим ее автоматически через некоторое время.

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: `${domain name}`, которая указывает на IP адрес из 2ого шага.
4. Создайте deployment с nginx nginx-dp в namespace default.
5. Создайте сервис для nginx-dp с именем svc-internal в namespace default с типом ClusterIP.
6. Создайте ingress nginx-ingress в namespace default для доменного имени `${domain name}`.
7. Настройте basic аутентификацию для данного ingress с логином user и паролем password.
8. Отправьте задание на проверку.