

Ingress 指南

AUTHOR: 彭玲 TIME: 2022/6/8

Ingress 指南

[Ingress 是什么?](#)

[Ingress 资源](#)

[Ingress 规则](#)

[主机名通配符](#)

[Ingress 配置](#)

1. 全局设置 (ConfigMap)

[ingress-controller 配置文件](#)

[nginx 配置文件 \(nginx.conf\)](#)

2. 注解 (annotations)

[身份验证](#)

[金丝雀](#)

[Rewrite](#)

[Server snippet](#)

[Configuration snippet](#)

[Client Body Buffer Size](#)

[Redirect from/to www](#)

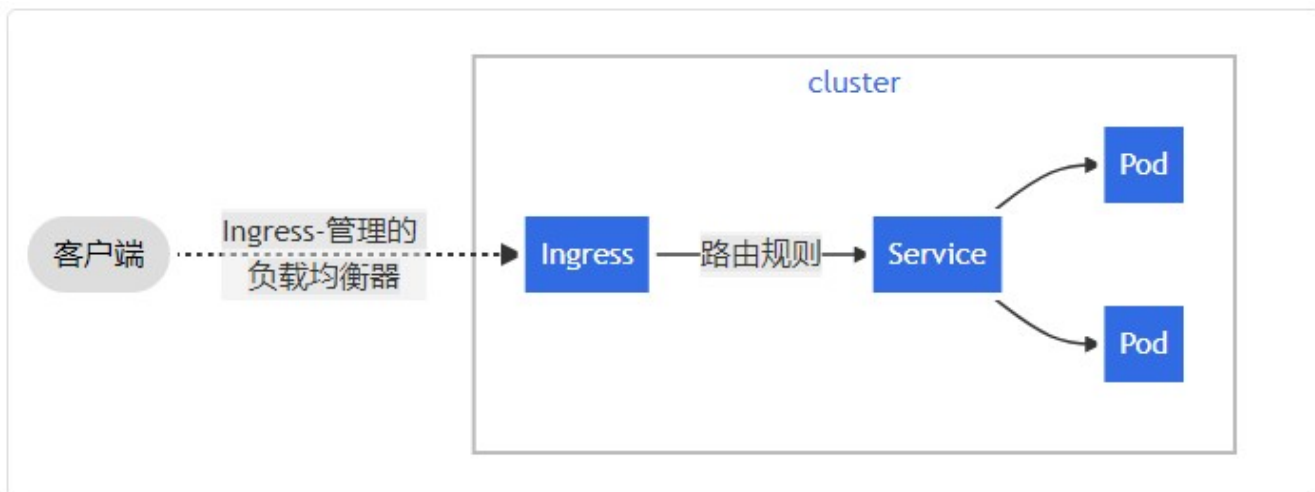
[Custom timeouts](#)

[Custom max body size](#)

3. custom template

Ingress 是什么?

Ingress 公开了从集群外部到集群内服务的 HTTP 和 HTTPS 路由。流量路由由 Ingress 资源上定义的规则控制。下面是一个将所有流量都发送到同一 Service 的简单 Ingress 示例:



Ingress 资源

Ingress 经常使用注解来配置一些选项，不同的 Ingress 控制器支持不同的注解，具体取决于 Ingress 控制器。

Ingress 规约提供了配置负载均衡器或者代理服务器所需的所有信息。最重要的是，其中包含与所有传入请求匹配的规则列表。

一个最小的 Ingress 资源示例：

```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: minimal-ingress
5    annotations:
6      nginx.ingress.kubernetes.io/rewrite-target: /
7  spec:
8    rules:
9      - http:
10         paths:
11           - path: /testpath
12             pathType: Prefix
13             backend:
14               service:
15                 name: test
16                 port:
17                   number: 80
```

Ingress 规则

每个 HTTP 规则都包含以下信息：

- **host** (可选)。在上面的一个最小的 Ingress 资源示例中未指定 **host**，因此该规则适用于通过指定 IP 地址的所有入站 HTTP 通信。如果提供了 **host** (例如 `foo.bar.com`)，则 **rules** 适用于该 **host**。
- **paths** 路径列表 (例如 `/testpath`)，每个路径都有一个由 **serviceName** 和 **servicePort** 定义的关联后端。在负载均衡器将流量定向到引用的 Service 之前，**host** 和 **path** 都必须匹配传入请求的内容。
- **backend** (后端) 是 **service** 和端口名称的组合。对 Ingress 的 HTTP (和 HTTPS) 请求 (该请求与 rule 的 **host** 和 **path** 匹配) 将发送到列出的 **backend**。

主机名通配符

主机名可以是精确匹配 (例如 `"foo.bar.com"`) 或者使用通配符来匹配 (例如 `"*.foo.com"`)。

- 精确匹配要求 HTTP **host** 头部字段与 **host** 字段值完全匹配。
- 通配符匹配则要求 HTTP **host** 头部字段与通配符规则中的后缀部分相同。

```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: ingress-wildcard-host
5  spec:
6    rules:
7    - host: "foo.bar.com"
8      http:
9        paths:
10       - pathType: Prefix
11         path: "/bar"
12         backend:
13           service:
14             name: service1
15             port:
16               number: 80
17    - host: "/*.foo.com"
18      http:
19        paths:
20       - pathType: Prefix
21         path: "/foo"
22         backend:
23           service:
24             name: service2
25             port:
26               number: 80
```

Ingress 配置

在 `ingress` 中配置分为了三部分：

- `configmap`：设置 `ingress-nginx` 的全局设置。
- `annotations`：`ingress` 特定的设置。
- `custom template`。

1. 全局设置 (ConfigMap)

全局配置一般是使用 `configmap`。

ingress-controller 配置文件

下面是一个 `ingress-controller` 的配置：

```
1  kind: ConfigMap
2  apiVersion: v1
3  metadata:
4    name: ingress-nginx-controller
5    namespace: ingress-nginx
6    labels:
7      app.kubernetes.io/component: controller
```

```

8   app.kubernetes.io/instance: ingress-nginx
9   app.kubernetes.io/managed-by: Helm
10  app.kubernetes.io/name: ingress-nginx
11  app.kubernetes.io/version: 0.40.2
12  helm.sh/chart: ingress-nginx-3.6.0
13  annotations:
14  data:
15    allow-backend-server-header: 'true'
16    client-body-buffer-size: 20m
17    enable-underscores-in-headers: 'true'
18    generate-request-id: 'true'
19    gzip-level: '6'
20    gzip-types: >-
21      application/atom+xml application/javascript application/x-javascript
22      application/json application/rss+xml application/vnd.ms-fontobject
23      application/x-font-ttf application/x-web-app-manifest+json
24      application/xhtml+xml application/xml font/opentype image/svg+xml
25      image/x-icon text/css text/javascript text/plain text/x-component
26    ignore-invalid-headers: 'true'
27    keep-alive: '75'
28    large-client-header-buffers: 4 128k
29    log-format-upstream: >-
30      $remote_addr - [$remote_addr] - $remote_user [$time_local] "$request"
31      $status $body_bytes_sent "$http_referer" "$http_user_agent" $request_length
32      "$http_x_forwarded_for" $remote_addr $request_time [$proxy_upstream_name]
33      $upstream_addr $upstream_response_length $upstream_response_time
34      $upstream_status $req_id $host
35    max-worker-connections: '65536'
36    proxy-body-size: 20m
37    proxy-buffer-size: 64k
38    proxy-connect-timeout: '300'
39    proxy-next-upstream-timeout: '10'
40    proxy-read-timeout: '300'
41    proxy-send-timeout: '300'
42    reuse-port: 'true'
43    server-tokens: 'false'
44    upstream-keepalive-connections: '20000'
45    upstream-keepalive-requests: '100000'
46    upstream-keepalive-timeout: '3000'
47    use-forwarded-headers: 'true'
48    use-gzip: 'true'
49    worker-cpu-affinity: auto

```

nginx 配置文件 (nginx.conf)

上面的 ingress-controller 配置，相当于 nginx 的 nginx.conf 配置文件：

```

1  user www-data;
2  worker_processes auto;
3  pid /run/nginx.pid;
4  include /etc/nginx/modules-enabled/*.conf;
5
6  events {

```

```
7     worker_connections 768;
8     multi_accept on;
9 }
10
11 http {
12
13     ##
14     # Basic Settings
15     ##
16
17     sendfile on;
18     tcp_nopush on;
19     tcp_nodelay on;
20     keepalive_timeout 65;
21     types_hash_max_size 2048;
22     server_tokens off;
23
24     # server_names_hash_bucket_size 64;
25     # server_name_in_redirect off;
26
27     include /etc/nginx/mime.types;
28     default_type application/octet-stream;
29
30     ##
31     # SSL Settings
32     ##
33
34     ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
35     ssl_prefer_server_ciphers on;
36
37     ##
38     # Logging Settings
39     ##
40
41     access_log /var/log/nginx/access.log;
42     error_log /var/log/nginx/error.log;
43
44     ##
45     # body size
46     ##
47     client_max_body_size 10m;
48     proxy_read_timeout 300;
49
50     ##
51     # Gzip Settings
52     ##
53
54     gzip on;
55     gzip_vary on;
56     gzip_proxied any;
57     gzip_comp_level 6;
58     gzip_buffers 16 8k;
59     gzip_http_version 1.1;
```

```

60     gzip_types text/plain text/css application/json application/javascript text/xml
    application/xml application/xml+rss text/javascript;
61
62     ##
63     # Virtual Host Configs
64     ##
65
66     include /etc/nginx/conf.d/*.conf;
67     include /etc/nginx/sites-enabled/*;
68 }

```

2. 注解 (annotations)

注解都是以 `nginx.ingress.kubernetes.io` 作为前缀添加到特定的 ingress 实例上的。

身份验证

```

1  # http 身份验证的类型
2  nginx.ingress.kubernetes.io/auth-type: [basic|digest]
3
4  # Secret的名称, 其中包含用户名和密码
5  nginx.ingress.kubernetes.io/auth-secret: secretName
6
7  # auth-secret 有两种格式:
8  # auth-file 默认情况下, 密钥'auth'中的htpasswd文件在密钥内
9  # auth-map 密钥的密钥是用户名, 值是哈希密码
10 nginx.ingress.kubernetes.io/auth-secret-type: [auth-file|auth-map]
11
12 nginx.ingress.kubernetes.io/auth-realm: "realm string"

```

金丝雀

```

1  nginx.ingress.kubernetes.io/canary-by-header
2  nginx.ingress.kubernetes.io/canary-by-header-value
3  nginx.ingress.kubernetes.io/canary-by-cookie
4  nginx.ingress.kubernetes.io/canary-by-cookie

```

Rewrite

在某些情况下, 后端服务中公开的 URL 与 Ingress 规则中指定的路径不同。没有重写, 任何请求都将返回 404。设置 `nginx.ingress.kubernetes.io/rewrite-target` 注解到服务期望的路径。如果应用程序根目录暴露在其他路径中, 并且需要重定向, 请设置注释 `nginx.ingress.kubernetes.io/app-root` 重定向请求到 `/`。

Server snippet

使用注解 `nginx.ingress.kubernetes.io/server-snippet` 可以在服务器配置块中添加自定义配置。

```

1  apiVersion: extensions/v1beta1
2  kind: Ingress
3  metadata:
4    annotations:

```

```
5 | nginx.ingress.kubernetes.io/server-snippet: |
6 |     set $agentflag 0;
7 |
8 |     if ($http_user_agent ~* "(Mobile)" ){
9 |         set $agentflag 1;
10 |     }
11 |
12 |     if ( $agentflag = 1 ) {
13 |         return 301 https://m.example.com;
14 |     }
```

Configuration snippet

使用此注释，您可以将其他 `配置` 添加到 NGINX 位置。例如：

```
1 | nginx.ingress.kubernetes.io/configuration-snippet: |
2 |     more_set_headers "Request-Id: $req_id";
```

Client Body Buffer Size

设置缓冲区大小，以读取每个位置的客户端请求正文。如果请求主体大于缓冲区，整个 Body 或只将其一部分写入一个临时文件。默认情况下，缓冲区大小等于两个内存页。在 x86、其他 32 位平台和 x86-64 上为 8K。在其他 64 位平台上通常为 16K。

该注解是 应用于入口规则中提供的每个位置。

```
1 | nginx.ingress.kubernetes.io/client-body-buffer-size: "1000" # 1000 bytes
2 | nginx.ingress.kubernetes.io/client-body-buffer-size: 1k # 1 kilobyte
3 | nginx.ingress.kubernetes.io/client-body-buffer-size: 1K # 1 kilobyte
4 | nginx.ingress.kubernetes.io/client-body-buffer-size: 1m # 1 megabyte
5 | nginx.ingress.kubernetes.io/client-body-buffer-size: 1M # 1 megabyte
```

Redirect from/to www

在某些情况下，需要从 www.domain.com 重定向到 `domain.com`，反之亦然。

使用 `nginx.ingress.kubernetes.io/from-to-www-redirect: "true"` 注解开启此功能。

Custom timeouts

使用配置 `configmap` 可以为 `ingress-nginx` 设置默认的全局超时。在某些情况下，要求具有不同的值。为此，允许自定义注释：

```
1 | nginx.ingress.kubernetes.io/proxy-connect-timeout
2 | nginx.ingress.kubernetes.io/proxy-send-timeout
3 | nginx.ingress.kubernetes.io/proxy-read-timeout
4 | nginx.ingress.kubernetes.io/proxy-next-upstream
5 | nginx.ingress.kubernetes.io/proxy-next-upstream-timeout
6 | nginx.ingress.kubernetes.io/proxy-next-upstream-tries
7 | nginx.ingress.kubernetes.io/proxy-request-buffering
```

Custom max body size

当请求中的大小超过客户端请求正文的最大允许大小时，将向客户端返回 413 错误。大小可以通过 `client_max_body_size` 参数配置。与上面类似，想要对所有 ingress 规则进行全局设置，可以在 nginx ConfigMap 中设置 `proxy-body-size`。要在 Ingress 规则中使用自定义值，请定义以下注解：

```
1 | nginx.ingress.kubernetes.io/proxy-body-size: 8m
```

3. custom template

暂无。