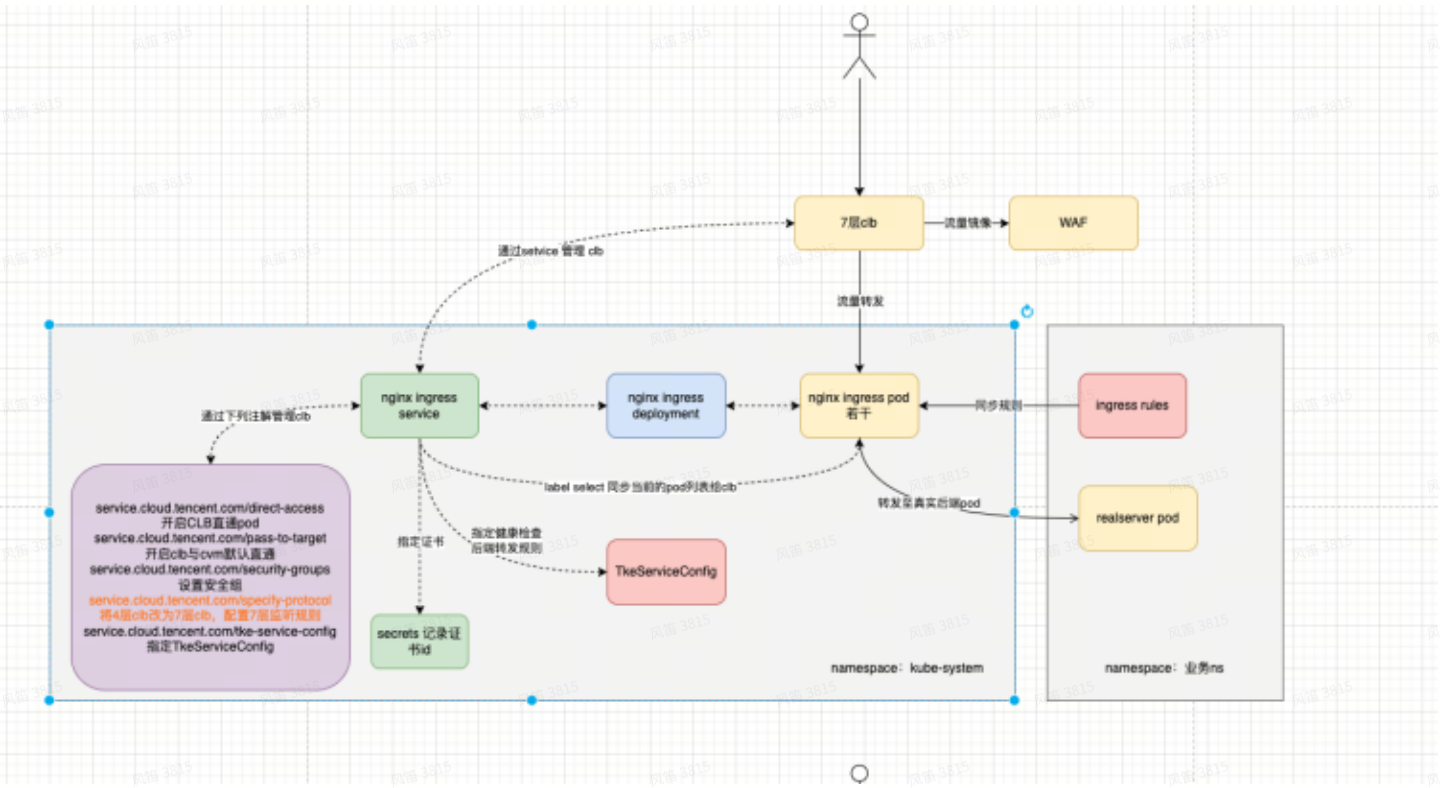
nginx ingress架构说明&注意事项

架构说明



核心架构还是沿用官方nginx ingress的架构。由于腾讯云对官方nginx ingress做了一些调整，运维侧又为了满足业务上的需求，在最腾讯云提供的nginx ingress架构上又添加了一些额外的配置、注解等。在后续维护、管理上需注意这部分。下文会对此部分详细说明。

CLB 负载均衡

CLB用作于业务的唯一入口，相关域名直接解析至CLB的IP地址。WAF配置业务域名，通过旁路的方式接入。请注意：

- nginx Ingress 实例是管理CLB的唯一方式。为保持稳定性和一致性，禁止直接通过控制台、Terraform或API修改CLB。所有配置更新必须通过nginx Ingress 实例进行。
- 负载均衡器的IP地址稳定性对业务至关重要，因为客户会将该地址加入访问白名单。IP地址的任何变动都需谨慎处理，并确保及时通知所有客户

nginx ingress

鉴于腾讯云提供的默认 Nginx Ingress 模式未能完全符合我们的业务及运维需求，我们对其进行了定制化的调整和优化。对默认逻辑进行了改进，并引入了额外的扩展功能。

主要变动如下：

将CLB 4层转发修改为7层转发。

1. 腾讯云的负载均衡器WAF只能与7层负载均衡器（CLB）集成，而 Nginx Ingress 默认情况下配置为4层转发。
    - 出于IP地址稳定性考虑，决定不使用SAAS WAF服务，因为它只支持CNAME解析方式接入，无法保证IP地址的稳定性。这也可能导致开发、测试和预发布环境的网络链路与生产环境不一致。
2. 默认的Nginx Ingress采用4层转发，要求在每个命名空间中单独管理证书。改为7层转发后，可以在 kube-system 命名空间集中管理单一证书，简化了证书管理。

开启CLB直通Pod模式

默认模式下CLB的后端地址为nginx ingress Service的Nodeport端口。此模式下CLB后端服务器直接指定nginx ingress Pod地址。减少了一层Nodeport转发，效率更高，同时避免环路的问题。

nginx ingress 实例管理说明：

安装nginx ingress实例参考： https://cloud.tencent.com/document/product/457/50503#Nginx-ingress

安装完成和后续维护还需注意下面几个配置：

service.cloud.tencent.com/specify-protocol

对nginx ingress Service添加此Service注解，用于将clb默认的4层转发变更为7层转发，同时创建相对应的监听。

通常此注解只在nginx ingress Service（对应的CLB）刚刚创建时添加即可，后续无需维护。

有一种特殊情况是新增了一个域名需接入WAF防火墙，接入WAF的域名要求在CLB内必须有对应的域名转发规则。此情况需要在注解内添加相应的域名

```
  annotations:
    service.cloud.tencent.com/specify-protocol: '{"443":{"protocol":
["HTTPS"],"hosts":{"*.yingdao.com":{"tls":"yingdao-com"},"*.shadow-rpa.net":
{"tls":"shadow-rpa-net"},"*.winrobot360.com":{"tls":"winrobot360-com"}}},"80":
{"protocol":["HTTP"],"hosts":{"*.yingdao.com":{},"*.shadow-rpa.net":
{},"*.winrobot360.com":{}}}}'
```

注解内容为json格式，展开后为如下：

```
{
  "80": {  // 管理clb创建一个80端口的监听
```

```
      "protocol": [
        "HTTP" // 80端口监听的协议为HTTP
      ],
      "hosts": {
        "*.yingdao.com": {}, // 80端口监听下的域名转发规则
        "*.shadow-rpa.net": {},
        "*.winrobot360.com": {}
      }
    },
    "443": {
      "protocol": [
        "HTTPS"
      ],
      "hosts": {
        "*.yingdao.com": {
          "tls": "yingdao-com"  // 指定证书，此证书需要在kube-system命名空间下创建
        },
        "*.shadow-rpa.net": {
          "tls": "shadow-rpa-net"
        },
        "*.winrobot360.com": {
          "tls": "winrobot360-com"
        },
        "api.yingao.com": {
          // 一般情况下使用通配符域名即可。无需对精确域名添加解析规则。
          // 但若域名需接如WAF，则需专门创建一条此监听。接入WAF的规则不能使用通配符
          "tls": "yingdao-com"
        }

      }
    }
}
```

TkeServiceConfig

`TkeServiceConfig` 是腾讯云提供的CRD，一般用来配合 `service.cloud.tencent.com/specify-protocol` 注解使用。

`TkeServiceConfig` 的作用是完善CLB监听规则的具体配置，如负载均衡方式、后端协议、健康检查、会话保持等。在我们的使用场景下， `TkeServiceConfig` 主要用于完善通配符域名监听规则的配置（因为通配符域名监听规则较为特殊，无法自动创建健康检查规则。精确域名无此问题）。所以此CRD一般仅在创建nginx ingress Service后创建一次即可，后续无需维护。除非对精确域名有特殊的规则需求。

```
apiVersion: cloud.tencent.com/v1alpha1
```

```yaml
kind: TkeServiceConfig
metadata:
  name: staging
  namespace: kube-system
spec:
  loadBalancer:
    l7Listeners:
    - protocol: HTTPS
      port: 443
      defaultServer: "staging-api.yingdao.com"
      keepaliveEnable: 1
      domains:
      - domain: "*.yingdao.com"
        rules:
        - url: "/"
          forwardType: HTTPS
          session:
            enable: false
          healthCheck:
            enable: true
            intervalTime: 10
            timeout: 5
            healthNum: 2
            unHealthNum: 2
            httpCheckPath: "/"
            httpCheckDomain: "staging-api.yingdao.com"
            httpCheckMethod: HEAD
          scheduler: WRR
      - domain: "*.winrobot360.com"
        rules:
        - url: "/"
          forwardType: HTTPS
          session:
            enable: false
          healthCheck:
            enable: true
            intervalTime: 10
            timeout: 5
            healthNum: 2
            unHealthNum: 2
            httpCheckPath: "/"
            httpCheckDomain: "staging-api.winrobot360.com"
            httpCheckMethod: HEAD
          scheduler: WRR
      - domain: "*.shadow-rpa.net"
        rules:
        - url: "/"
```

```
          forwardType: HTTPS
          session:
            enable: false
          healthCheck:
            enable: true
            intervalTime: 10
            timeout: 5
            healthNum: 2
            unHealthNum: 2
            httpCheckPath: "/"
            httpCheckDomain: "staging-boss.shadow-rpa.net"
```

`TkeServiceConfig` 创建完成后需要在nginx ingress Service中指定对应的 `TkeServiceConfig` 。通过 `service.cloud.tencent.com/tke-service-config` 注解将其关联起来：

```
  annotations:
    service.cloud.tencent.com/tke-service-config: online
```

service.cloud.tencent.com/direct-access

对nginx ingress Service添加此注解，开启CLB直通Pod模式。

对Service开启LB直通Pod模式前，需要先确认集群是否开启了直通Pod能力。查看ConfigMap `kube-system/tke-service-controller-config`

```
apiVersion: v1
data:
  GlobalRouteDirectAccess: "true"  # 如无此行，需新增此行，开启集群的直通Pod能力
  LOADBALANCER_CRD_SUPPORT: "true"
  REUSE_LOADBALANCER: "false"
kind: ConfigMap
metadata:
  name: tke-service-controller-config
  namespace: kube-system
```

确认集群开了直通Pod能力后，在对应的nginx ingress Service 中添加注解：

```
  annotations:
    service.cloud.tencent.com/direct-access: "true"
```

配置完成后可以在负载均衡中看到后端地址被指向为了pod地址

- 参考文档： https://cloud.tencent.com/document/product/457/41897

service.cloud.tencent.com/security-groups

在nginx ingress service中添加注解，配置安全组与是否启用默认放通：

```
  annotations:
    service.cloud.tencent.com/security-groups: "sg-xxxxxx,sg-xxxxxx" # 指定安全组，若值为 "" 或无此annotations则不绑定安全组
    service.cloud.tencent.com/pass-to-target: "true"  # 开启安全组默认放通
```

service.cloud.tencent.com/enable-grace-shutdown

开启Service优雅停机

更多信息参考： https://cloud.tencent.com/document/product/457/60064

开启gzip压缩

```
kubectl --namespace kube-system edit configmaps staging-ingress-nginx-controller

data:
## 添加下列两行
  use-gzip: "true"
  gzip-level: "7"
```

Ingress rules管理说明：

常用配置

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: staging。 # 指定nginx ingress实例名字
    kubernetes.io/ingress.rule-mix: "true"
    nginx.ingress.kubernetes.io/force-ssl-redirect: "true" # 开启http转发至https
```

```yaml
  name: staging-api-ng
  namespace: staging
spec:
  rules:
    - host: staging-api.yingdao.com
      http: &staging-api
        paths:
          - backend:
              service:
                name: xybot-gateway
                port:
                  number: 8079
            path: /
            pathType: Prefix
          - backend:
              service:
                name: xybot-404
                port:
                  number: 80
            path: /actuator
            pathType: Prefix
    - host: staging-api.winrobot360.com
      http: *staging-api
# 证书在nginx ingress Service中通过注解管理，此处无需指定证书相关的配置
```

使用snippet

使用 `nginx.ingress.kubernetes.io/configuration-snippet` 之前需要先开启nginx ingress实例的 `allow-snippet` 。此选项默认是关闭的

```
kubectl --namespace kube-system edit configmaps staging-ingress-nginx-controller
```

将 `allow-snippet-annotations: "false"` 改为 `true`

在ingress rules中使用 `nginx.ingress.kubernetes.io/configuration-snippet` 注解：

```yaml
metadata:
  annotations:
    kubernetes.io/ingress.class: staging
    kubernetes.io/ingress.rule-mix: "true"
    nginx.ingress.kubernetes.io/configuration-snippet: |
      if ($host = 'eryitest.winrobot360.com') {
```

```
7           return 301 https://eryitest.yingdao.com$request_uri;
8       }
```