

TKE Ingress 获取真实源 IP (gr模式实现clb非直连业务 pod)

2025-07-03 16:52

腾讯云负载均衡器（CLB 七层）默认会将客户端真实源 IP 放至 HTTP Header 的 X-Forwarded-For 和 X-Real-IP 字段。当服务流量在经过 Service 四层转发后会保留上述字段，后端通过 Web 服务器代理配置或应用代码方式获取到客户端真实源 IP

通过容器服务控制台 获取源 IP 步骤如下：

1.创建一个工作负载（Deployment）（以nginx为例）

• 基本信息

资源对象

• 节点管理

• 命名空间

• 工作负载

• Pod

• 服务与路由

• 配置管理

• 存储

• 资源对象浏览器

DeploymentStatefulsetDaemonsetJobCronjob

新建监控Workload Map

命名空间kube-system名称只能搜索

<input type="checkbox"/>	名称	Labels	Selector	镜像	运行/期望Po...	Request/Li...	操作
<input type="checkbox"/>	coredns	addonmanage... app.kubernete... k8s-app:kube-... ...	k8s-app:kub...	ccr.ccs.tencentyu n.com/tkeimages/ coredns:1.11.1	2/2	CPU : 0.1/ 无 限制 内存 : 30/ 2048 Mi	更新Pod数量 更新Pod配置
<input type="checkbox"/>	csi-cbs-controller	app.kubernete... app:cbs-csi-...	app:cbs-csi-...	ccr.ccs.tencentyu n.com/tkeimages/ csi-provisioner:v2.0.8 ccr.ccs.tencentyu	1/1	CPU : 0.5/ 7 核 内存 : 250/	更新Pod数量 更新Pod配置

名称

nginx

最长63个字符，只能包含小写字母、数字及分隔符("-"), 且必须以小写字母开头，数字或小写字母结尾

描述

请输入描述信息，不超过1000个字符

命名空间

default

Labels

新增
标签键名称不超过63个字符,仅支持英文、数字、"/"、"."且不允许以("/")开头。支持使用前缀，更多说明[查看详情](#)
标签键值只能包含字母、数字及分隔符("-", "_", "."), 且必须以字母、数字开头和结尾

Annotations

新增
Annotations键名称不超过63个字符，仅支持英文、数字、"/"、"."且不允许以("/")开头。支持使用前缀，更多说明[查看详情](#)
Annotations值为字符串类型无长度限制。为保证可读性和可移植性，建议将值限制为较短字符串并避免使用特殊字符（如换行、空格等）。

OS类型

Linux

切换容器OS类型将会初始化配置

数据卷（选填）

添加数据卷
为容器提供存储，目前支持临时路径、主机路径、云硬盘数据卷、文件存储NFS、配置文件、PVC，还需挂载到容器的指定路径中。[使用指引](#)

实例内容器

container-1

+ 添加容器

实例内容器

nginx

+ 添加容器

名称

nginx

最长63个字符，只能包含小写字母、数字及分隔符("-"), 且不能以分隔符开头或结尾

镜像 ⓘ

nginx

选择镜像

镜像版本 (Tag)

不填默认为 latest

选择镜像版本

镜像拉取策略

Always

IfNotPresent

Never

总是从远程拉取该镜像

此时镜像使用nginx官方镜像，后期可以直接从镜像仓库中推送的镜像及对应版本进行替换

Deployment

Statefulset

Daemonset

Job

Cronjob

新建

监控

Workload Map

default

<input type="checkbox"/>	名称	Labels	Selector	镜像 ⓘ	运行/期望Pod...
<input type="checkbox"/>	nginx	k8s-app:nginx qcloud-app:nginx	k8s-app:nginx qcloud-app:ng...	eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:v1.0	1/1

此时保证nginx的运行和期望pod的数量满足预期或在pod管理中保证状态为运行（running）

Pod管理

详情

修订历史

应用性能

事件

日志

YAML

监控

销毁重建

多个过滤标签用回车键分隔

<input type="checkbox"/>	实例名称	状态 卣	实例所在节点IP	实例IP	Request/Limits	运行时间 ⓘ
▶ <input type="checkbox"/>	nginx-848fd654c5-lgpjb 卣	● Running	10.15.17.101 卣	172.18.0.69 卣	CPU : 0.25/ 0.5 核 内存 : 256/ 1024 Mi	0d 3h 12m

2 . 在服务与路由中调整service配置和创建Ingress

访问设置 (Service)

Service ☒ 启用

Service Name 不填默认与工作负载名称相同

服务访问方式 ☐ 仅在集群内访问 ☒ 主机端口访问 ☐ 公网LB访问 ☐ 内网LB访问 [如何选择](#)

即NodePort类型，提供一个主机端口映射到容器的访问方式，支持TCP&UDP，可用于业务定制上层LB转发到Node。

如您需要公网通过HTTP/HTTPS协议或根据URL转发，您可以在Ingress页面使用Ingress进行路由转发，[查看详情](#)

端口映射

协议①	容器端口①	主机端口①	服务端口①	名称
TCP	80	范围：30000~32767	80	最长63个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

[添加端口映射](#)

在此操作中选择svc的主机端口访问，选择的容器端口和服务端口为80端口，后期可根据Dockerfile文件中的端口号进行更换

基本信息

Ingress名称 nginxtest

最长63个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

描述 请输入描述信息，不超过1000个字符

命名空间 default

Ingress类型

应用型 CLB

Nginx Ingress Controller

Others

[详细对比](#)

应用型负载均衡器（支持HTTP/HTTPS）

转发规则相关配置

重定向 ☒ 无 ☐ 手动 ☐ 自动

自定义监听端口 ☐

HTTP 与 HTTPS协议的默认监听端口分别为 80 和 443，若需要使用非标准端口，可根据需求进行自定义

转发配置

协议	监听端口	域名①	路径	后端服务①	服务端口
HTTP	80	默认为IPv4 IP	eg: /	nginx	80

[添加转发规则](#)

此时转发配置，后端服务选择svc的服务名和选择的对应服务端口

Service		Ingress		NginxIngress	
新建		default		名称只能搜索一个关键字，Label格式要求:	
名称	类型	访问入口	后端服务	创建时间	操作
nginxtest	lb-jnuecsco 公网LB(按量计费)	203.195.143.248	http://203.195.143.248/-->nginx:80	2025-06-27 10:06:15	更新转发配置 编辑yaml 删除

3.登录腾讯云OrcaTerm

在云服务器中选择合适的服务器，在操作中选择登录

ID名称	监控	状态	可用区	实例类型	实例配置	主IPv4地址	主IPv6地址	实例计费模式	网络计费模式	所属项目	操作
ins-3rfwey9w 9ke_cls- d59frouz_worker1		运行中 未付安全策略	东京一区	标准型SAS	8核 16GB 0Mbps 系统盘: 通用型SSD云 硬盘 网络: 9ke-8-demo	10.0.1.142 (内)		按量计费 2025-04-29 16:49:22创建		默认项目	登录 更多

登录

腾讯云产品

云服务器 (CVM) ins-3rfwey9w

连接协议

☒ 免密连接 (TAT) ☐ 终端连接 (SSH)

用户名

root

☒ 保存登录信息到连接配置，下次快速登录 [如何快速登录](#)

登录

其他登录方式 [VNC登录](#)

可以选择免密登录腾讯云OrcaTerm或者选择终端登录

登录

×

腾讯云产品

云服务器 (CVM) ins-3rfwey9w

连接协议

☐ 免密连接 (TAT) ☒ 终端连接 (SSH)

连接网络

连接端口

内网 10.0.1.142

▼

36000

验证方式

☒ 密码验证 ☐ 密钥验证

🔑 输入密码

🔑 使用托管密码

用户名

密码

root

.....

×

👁

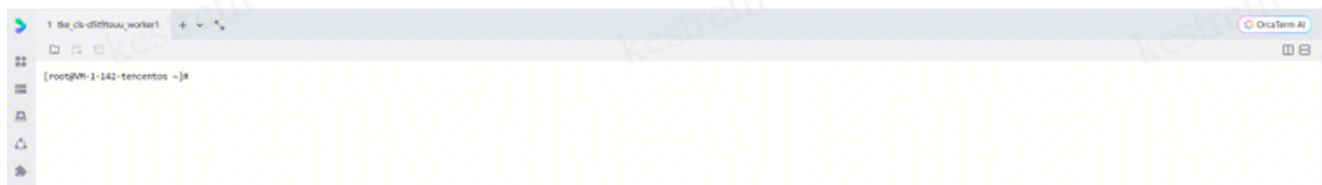
[忘记密码?](#)

☒ 保存登录信息与登录凭证, 下次快速登录 [如何快速登录 >](#)

登录

其他登录方式 [VNC登录](#) ⓘ

进入腾讯云OrcaTerm



此时检查腾讯云OrcaTerm中是否存在docker, 如果存在, 需卸载旧版本docker, 输入指令如下

```
1 sudo yum remove docker \
2   docker-client \
3   docker-client-latest \
4   docker-common \
5   docker-latest \
6   docker-latest-logrotate \
7   docker-logrotate \
8   docker-engine
```

如果不存在docker，需要安装docker

安装docker的步骤如下：

```
1  安装必要的依赖包：
2  sudo yum install -y yum-utils device-mapper-persistent-data lvm2
3  添加Docker官方仓库：
4  sudo yum-config-manager --add-repo
   https://download.docker.com/linux/centos/docker-ce.repo
5  安装Docker引擎：
6  sudo yum install -y docker-ce docker-ce-cli containerd.io
7  启动Docker服务并设置开机自启：
8  sudo systemctl start docker
9  sudo systemctl enable docker
```

4. 创建一个完整的Flask Web服务

Flask Web服务可以打印所有请求头并响应这些头信息，同时提供依赖包文件和Dockerfile用于部署

创建Flask Web服务操作步骤如下：

创建一个名为app.py的文件，包含Flask应用代码：

```

1  import logging
2
3  # 配置日志记录
4  logging.basicConfig(
5      level=logging.INFO,
6      format='%(asctime)s - %(levelname)s - %(message)s',
7      handlers=[
8          logging.StreamHandler(),
9          logging.FileHandler('app.log')
10     ]
11 )
12
13 app = Flask(__name__)
14
15 @app.route('/', methods=['GET', 'POST', 'PUT', 'DELETE', 'PATCH'])
16 def handle_request():
17     """处理所有HTTP方法请求，打印并返回请求头"""
18     # 获取所有请求头
19     headers = dict(request.headers)
20
21     # 打印请求头到控制台和日志文件
22     logging.info("Received request with headers:")
23     for header, value in headers.items():
24         logging.info(f"{header}: {value}")
25
26     # 返回JSON格式的请求头
27     return jsonify({
28         "message": "Here are your request headers",
29         "headers": headers,
30         "method": request.method
31     })
32
33 if __name__ == '__main__':
34     app.run(host='0.0.0.0', port=5000, debug=True)
35

```

创建一个requirements.txt文件，列出所有需要的Python包：

```

1  Flask==2.3.2
2  gunicorn==20.1.0

```

创建一个Dockerfile用于容器化部署：


```

1  # 使用官方Python运行时作为基础镜像
2  FROM python:3.9-slim
3
4  # 设置工作目录
5  WORKDIR /app
6
7  # 复制依赖文件并安装
8  COPY requirements.txt .
9  RUN pip install --no-cache-dir -r requirements.txt
10
11 # 复制应用代码
12 COPY app.py .
13
14 # 暴露端口
15 EXPOSE 5000
16
17 # 运行应用
18 CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]

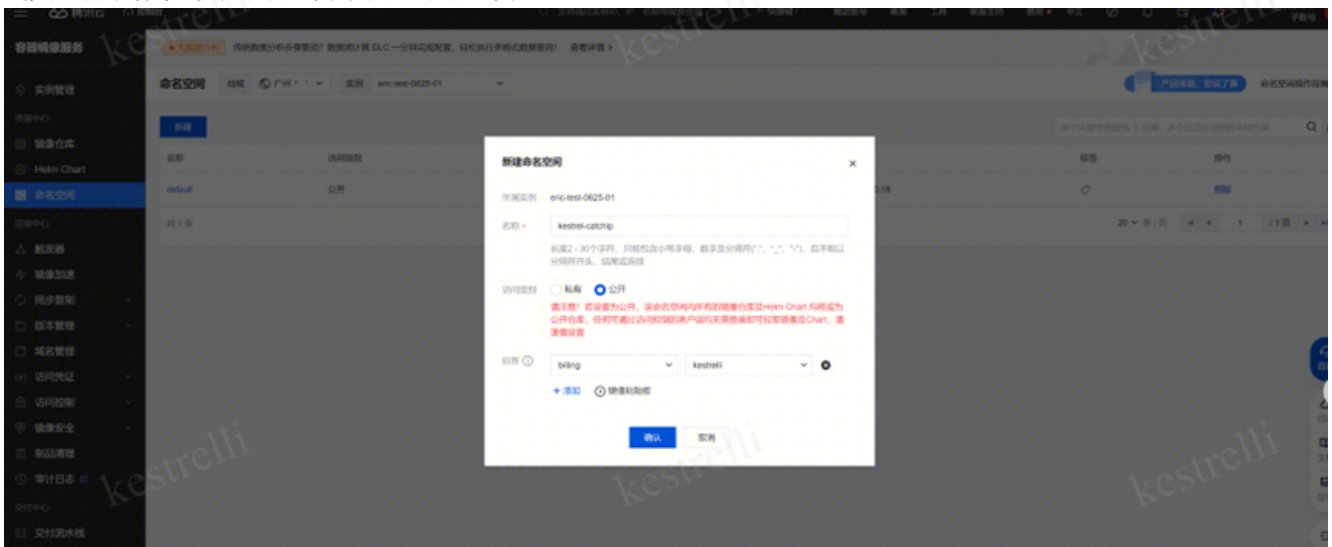
```

5.Docker部署及镜像推送

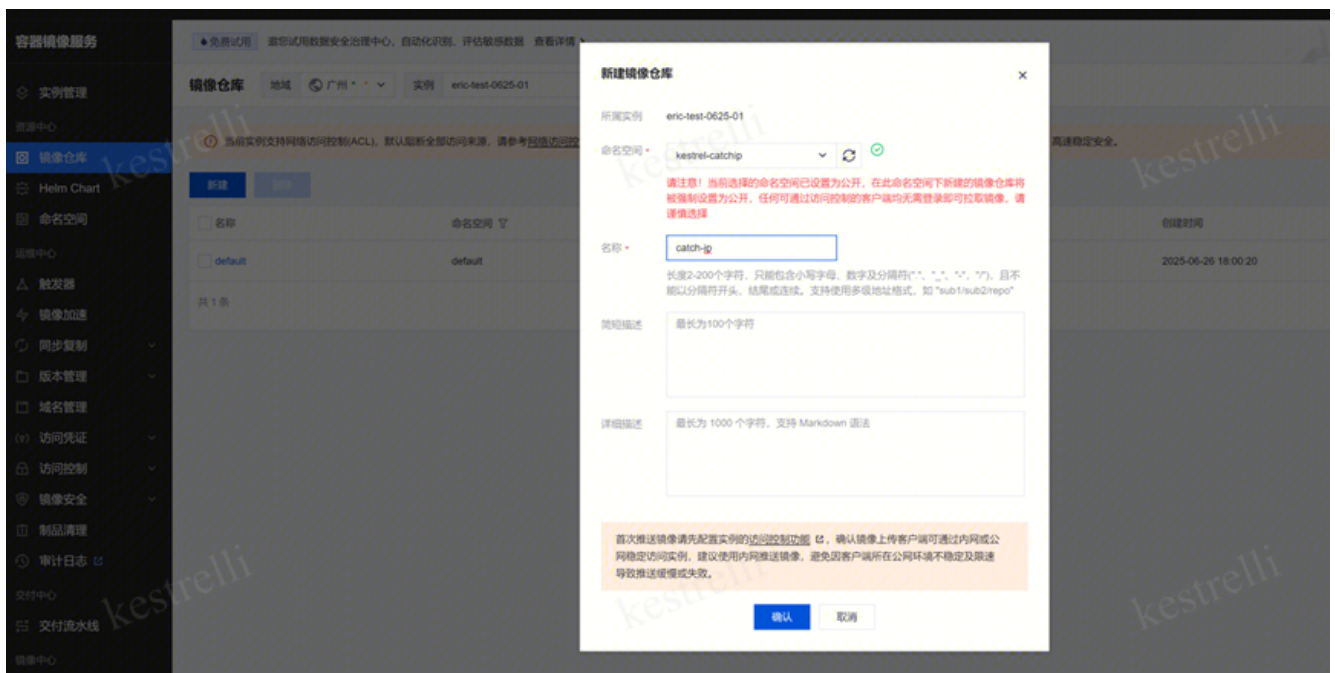
构建Docker镜像：docker build -t flask-headers .

此时docker镜像已经完成构建，推送镜像到腾讯云仓库

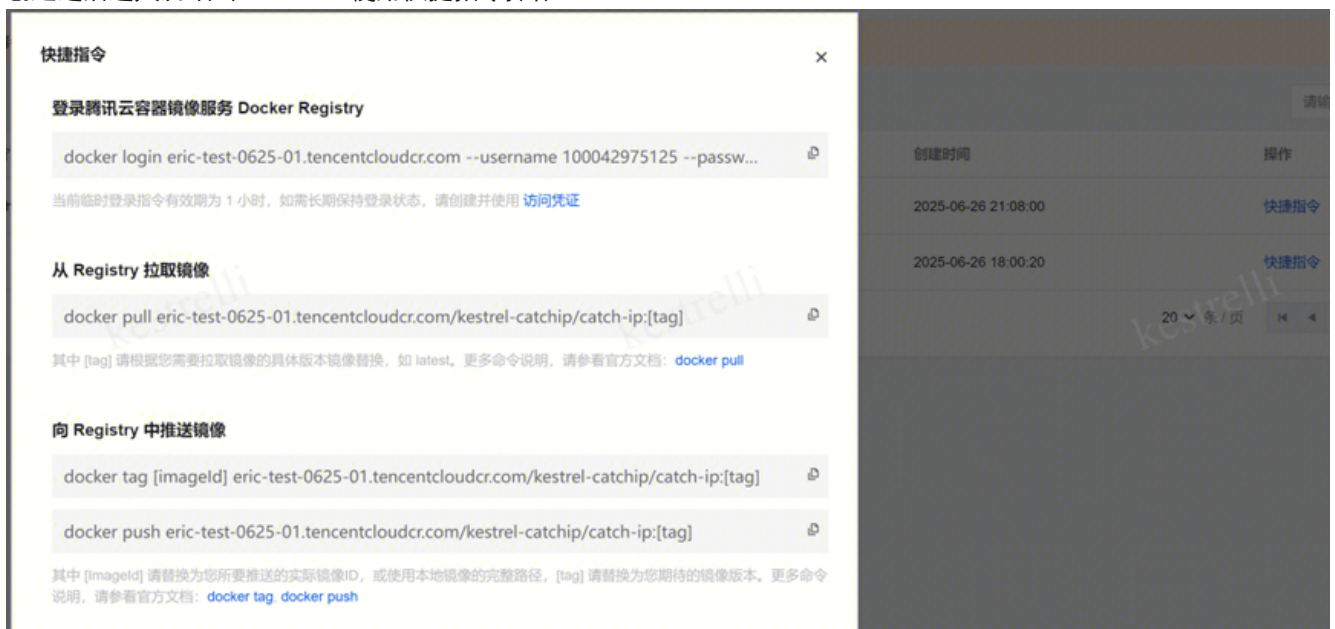
进入容器镜像服务，命名空间，新建命名空间



进入镜像仓库，新建镜像仓库



创建之后进入腾讯云OrcaTerm使用快捷指令操作



选取快捷指令

选择登录腾讯云容器镜像服务 Docker Registry的指令进行操作

```
[root@VM-16-20-tencentos ~]# docker login eric-test-0625-01.tencentcloudcr.com --username 100042975125 --password ey3hbGc10i35Uz11NiIsIntpZCI6Iiw5IA6UkdaSjpwM1RC01FSMkc6N1pHTJo3Tk5aojUzN0M6MjEzZmZc50TQlCjVvcGlyYXRvcjVpb1I6IjEwMDA0Mjk3NTEyNSIsImV4cCI6MTc1MDk5NjgwM5wibmJmIjoXNzUwOTkzMjAxLC7pYXQ10jE3NTA5OTMyMDF9.dheyY2aoB6vkuVgKn1NSYvr0PpGfPyBioztyAUJ0jdZwqbwin2TaSaX5BNzPQT9RrWEeoB93fw5a21S7G5Ma06bcXu_PZ47XM-SKmEcgd9-mnXWbnXU8DgbwFAxNfK8Ed4v8-csb8-1J1PvPe5DvxRmH8NtmXJ23XJAS1-VEa-Gyp_G0sXv04REDAgTKItynb8ru0LpApMrPV1PMCHUDAMT53KKVUMCxaPQVTsc4g3yI5nIes43w1SW1YEDGkq9gkexY10SV5GhFu8FJKyUmzg
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

此时表示已经成功登录到了腾讯云容器镜像仓库，之后向 Registry 中推送镜像

- 1 `docker tag [imageId] eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:[tag]`
- 2 `docker push eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:[tag]`

(其中 [Imageld] 请替换为所要推送的实际镜像ID，或使用本地镜像的完整路径，[tag] 替换为期待的镜像版本)

```
[root@VM-16-20-tencentos ~]# docker tag c37a5a2efd24 eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:v1.0
[root@VM-16-20-tencentos ~]# docker push eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:v1.0
The push refers to repository [eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip]
767616119c19: Pushed
275caaa2a12a: Pushed
ee1c98146918: Pushed
f2125cd07823: Pushed
f0d3b09df3b4: Pushed
b9a50396677d: Pushed
be76169c9d2c: Pushed
7fb72a7d1a8e: Pushed
v1.0: digest: sha256:ac4dde76727c9e04987bfd03225e34bbdb2f8e946e317ee8284a3abcd37e36b4 size: 1990
```

此时表示镜像以推送成功到指定命名空间的镜像仓库中



6.修改service及工作负载配置

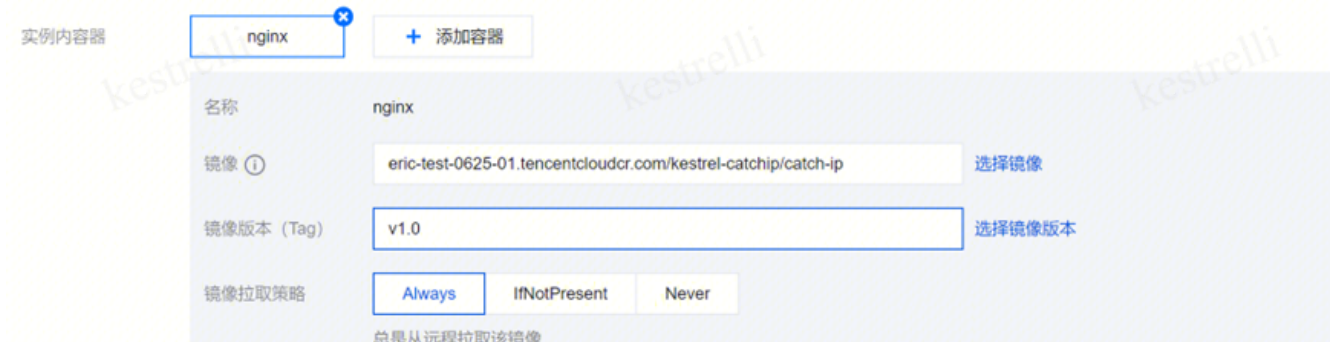
根据个人情况修改service的容器端口映射（以Dockerfile文件暴露服务端口为准，本次以5000端口为例）



Ingress对应更新转发配置



根据推送到镜像仓库的镜像和对应的版本进行工作负载中更新Deployment的pod配置



7. 通过TKE Ingress 获取真实源 IP

Service		Ingress	NginxIngress		
新建		default		名称只能搜索一个关键字, Label	
名称	类型	访问入口	后端服务	创建时间	
nginxtest	lb-jnuecsco 公网LB(按量计费)	203.195.143.248	http://203.195.143.248/-->nginx:80	2025-06-27 10:06:15	

在服务与路由中根据Ingress中的后端服务对应超链接点击访问，可在X-Forwarded-For或X-Real-Ip获取客户端真实IP



或者通过Ingress复制访问入口，通过cmd命令行输入curl+访问入口IP，同样可以在X-Forwarded-For或X-Real-Ip获取客户端真实IP

```
C:\Users\Tencent_Go>curl 203.195.143.248
{"headers":{"Accept":"*/*","Connection":"keep-alive","Host":"203.195.143.248","User-Agent":"curl/8.13.0","X-Client-Proto":"http","X-Client-Proto-Ver":"HTTP/1.1","X-Forwarded-For":"111.206.96.147","X-Forwarded-Proto":"http","X-Real-Ip":"111.206.96.147","X-Stgw-Time":"1751025556.325"},"message":"Here are your request headers","method":"GET"}
```