

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

2025-07-03 15:55

使用 TKE Ingress 获取真实源 IP

网络模式为VPC-CNI

如图所示

节点和网络信息

节点数量

3个 [查看资源用量](#)

默认操作系统

tlinux4\_x86\_64\_public\_uefi [✎](#)

系统镜像来源

公共镜像 - 基础镜像

节点hostname命名模式

自动命名 [✎](#)

节点网络

[vpc-pdnvwkgx](#) [🔗](#)

容器网络插件

VPC-CNI 共享网卡多IP

是否固定Pod IP

否

容器子网

子网ID	子网名称	子网CIDR	可用区	剩余IP	可用区可扩展Pod数
subnet-6c0yoz...	kestrel	10.15.17.0/24	广州四区	222	222

Service CIDR

172.17.0.0/17

Kube-proxy 转发模式

ipvs

ClusterIP增强 ①

未开启

注册节点能力 ①

☐

咨询

动态

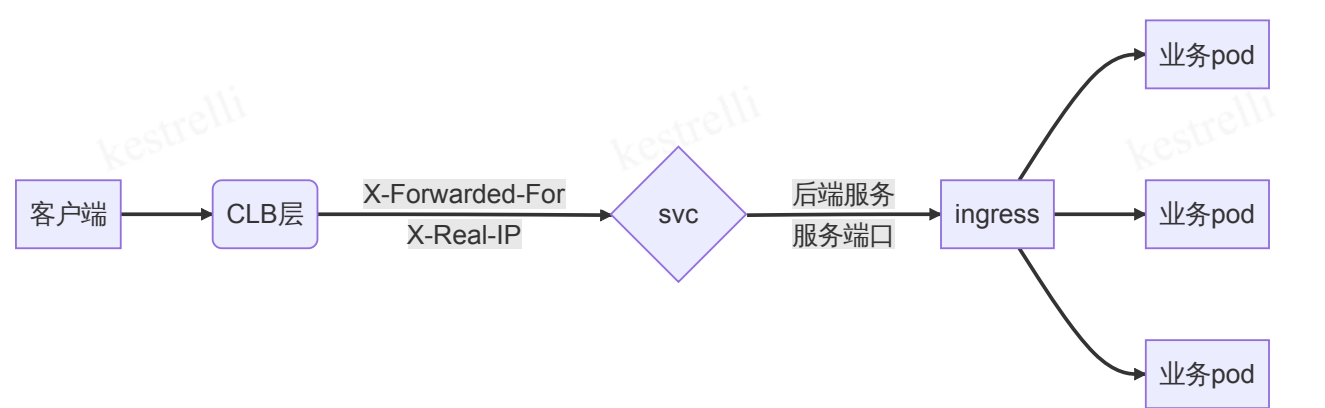
文档

反馈

业务访问链路：

客户端-> LB类型svc->LB类型ingress -> nodport -> 业务pod

转发流程如图所示



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)

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 ①

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

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 ①	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	实例所在节点IP	实例IP	Request/Limits	运行时间
	<input type="checkbox"/> nginx-848fd654c5-lgpbj	Running	10.15.17.101	172.18.0.69	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	容器端口	主机端口	服务端口	名称
TCP	80	范围：30000~32767	80	最长63个字符，只能包含

添加端口映射

后期可根据Dockerfile文件中的暴露端口号进行容器端口替换

基本信息

Ingress名称

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

转发配置中，后端服务选择service的服务名和对应服务端口

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

选择合适的云服务器进行登录

ID名称	图标	状态	可用区	实例类型	实例规格	主IPv4地址	主IPv6地址	实例计费模式	网络计费模式	创建时间	操作
ins-3fwoybw the_os- d59rou_worke1		运行中 三台安全实例	华东-一	标准型SAP	8核 16GB SA7ops 系统盘: 通用型SSD云 硬盘: 云硬盘 网络: 独享	10.0.1.142 (内)		按量计费		2025-04-29 16:49:22创建	默认设置 登录 更多

1 the\_os-d59rou\_worke1

root@the-1-142-tescentos ~#

检查腾讯云OrcaTerm中是否存在docker，如果存在，需卸载旧版本docker

```
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Web服务

创建一个app.py的文件，包含应用代码：

```
1  from flask import Flask, request, jsonify
2  import logging
3
4  # 配置日志记录
5  logging.basicConfig(
6      level=logging.INFO,
7      format='%(asctime)s - %(levelname)s - %(message)s',
8      handlers=[
9          logging.StreamHandler(),
10         logging.FileHandler('app.log')
11     ]
12 )
13
14 app = Flask(__name__)
15
16 @app.route('/', methods=['GET', 'POST', 'PUT', 'DELETE', 'PATCH'])
17 def handle_request():
18     """处理所有HTTP方法请求，打印并返回请求头"""
19     # 获取所有请求头
20     headers = dict(request.headers)
21
22     # 打印请求头到控制台和日志文件
23     logging.info("Received request with headers:")
24     for header, value in headers.items():
25         logging.info(f"{header}: {value}")
26
27     # 返回JSON格式的请求头
28     return jsonify({
29         "message": "Here are your request headers",
30         "headers": headers,
31         "method": request.method
32     })
33
34 if __name__ == '__main__':
35     app.run(host='0.0.0.0', port=5000, debug=True)
36
```



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"]
19
```

## 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，按照快捷指令推送镜像到腾讯云仓库

**快捷指令**

**登录腾讯云容器镜像服务 Docker Registry**

```
docker login eric-test-0625-01.tencentcloudcr.com --username 100042975125 --passw...
```

当前临时登录指令有效期为 1 小时，如需长期保持登录状态，请创建并使用 [访问凭证](#)

**从 Registry 拉取镜像**

```
docker pull eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip:[tag]
```

其中 [tag] 请根据您的需要拉取镜像的具体版本镜像替换，如 latest。更多命令说明，请参考官方文档：[docker pull](#)

**向 Registry 中推送镜像**

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

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

其中 [imageId] 请替换为您所要推送的实际镜像ID，或使用本地镜像的完整路径，[tag] 请替换为您期待的镜像版本。更多命令说明，请参考官方文档：[docker tag](#) [docker push](#)

创建时间	操作
2025-06-26 21:08:00	<a href="#">快捷指令</a>
2025-06-26 18:00:20	<a href="#">快捷指令</a>

20 / 多页

按指令操作完成镜像推送

```
[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
```

版本号如图



## 6.修改service及工作负载配置

**访问设置(Service)**

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](#)

则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①	名称
TCP	5000	32515	80	80-tcp-3ayf48tfxya

[添加端口映射](#)

**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	/	nginx	80

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

**实例内容**

nginx [+ 添加容器](#)

名称 nginx

镜像① eric-test-0625-01.tencentcloudcr.com/kestrel-catchip/catch-ip [选择镜像](#)

镜像版本 (Tag) v1.0 [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



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转

名称	类型 了	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	<a href="http://203.195.143.248/">http://203.195.143.248/</a> -->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，通过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"}
```