LoadBalancer直连 Pod模式 Service获取真实源 IP Playbook (gr模式)

2025-07-11 18:15

目录

背景

本Playbook专门针对腾讯云容器服务(TKE)集群环境，指导您通过Orca Term终端实现CLB直连Pod模式的配置。

所有操作均在腾讯云Orca Term上执行，确保在TKE集群环境中，业务Pod能够获取客户端真实源IP。

本方案通过GlobalRouter网络模式实现，完全绕过NodePort转发，适用于需要真实客户端IP的应用场景（如安全审计、日志分析）。

前置条件

在Orca Term中开始操作前，请确保满足以下条件：

类别	要求	验证方式
集群环境	• TKE集群启用GlobalRouter网络模式 • 集群节点状态正常 • kubectl已配置访问权限	• 在TKE控制台确认网络模式 • `kubectl get nodes`检查节点状态 • `kubectl cluster-info`验证连接
镜像准备	• 业务镜像已推送至腾讯云镜像仓库 • 有权限拉取镜像	• 确认镜像地址格式：`<仓库>.tencentcloudcr.com/<命名空间>/<镜像>:<标签>` • 在Orca Term测试：`docker pull <镜像地址>`
访问权限	• Orca Term已绑定集群节点 • 拥有操作kubectl的权限 • 账户有创建CLB的配额	• 在Orca Term确认节点登录状态 • 尝试运行`kubectl get pods`验证权限 • 检查腾讯云账号余额和CLB配额
业务准备	• 已知业务服务端口 • 准备测试客户端	• 确认Deployment的containerPort • 准备可访问公网的设备（验证用）

操作流程

以下步骤均在腾讯云Orca Term中执行，专为TKE集群环境优化。

Step 1: 启用集群GlobalRoute直连能力

在Orca Term中配置集群级直连开关

```
# 1. 编辑ConfigMap
kubectl edit configmap tke-service-controller-config -n kube-system

# 2. 在vi编辑器中添加关键参数
# 定位到data字段，添加新行：
GlobalRouteDirectAccess: "true"

# 3. 保存退出
# 按ESC键，输入:wq保存（Orca Term使用标准vi操作）

# 4. 验证配置
kubectl get configmap tke-service-controller-config -n kube-system -o yaml |
grep GlobalRouteDirectAccess
```

预期输出：`GlobalRouteDirectAccess: "true"`

```
[root@VM-17-53-tlinux ~]# kubectl edit configmap tke-service-controller-config -n kube-system
Edit cancelled, no changes made.
[root@VM-17-53-tlinux ~]# kubectl get configmap tke-service-controller-config -n kube-system -o yaml | grep GlobalRouteDirectAccess
    GlobalRouteDirectAccess: "true"
```

关键点：此配置启用集群维度的直连能力，是后续操作的基础。

Step 2: 创建业务工作负载（Deployment）

在Orca Term中通过命令行创建业务Deployment：

1.创建 Deployment YAML 文件(deployment.yaml)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: real-ip-demo
  labels:
    app: real-ip-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: real-ip-app
  template:
    metadata:
      labels:
        app: real-ip-app
    spec:
      containers:
      - name: real-ip-container
        # 使用我的业务镜像
        image: vickytan-demo.tencentcloudcr.com/kestrelli/images:v1.0
        ports:
        - containerPort: 5000  # 替换为实际业务端口
```

2.部署工作负载

```
kubectl apply -f deployment.yaml
```

3.验证 Pod 状态

```
watch kubectl get pods -l app=real-ip-app
```

📌 关键配置

- `metadata.labels` 需与后续 Service 选择器匹配
- `containerPort` 需与业务实际端口一致
- `replicas`：根据业务需求调整副本数
- `image`：若有需要替换为您的腾讯云镜像仓库地址

验证要求：所有Pod状态为`Running`（按Ctrl+C退出watch）

```
Every 2.0s: kubectl get pods -l app=real-ip-app

NAME                             READY   STATUS    RESTARTS   AGE
real-ip-demo-8b55f64d-26wd4      1/1     Running   0          9m1s
real-ip-demo-8b55f64d-sbxhf      1/1     Running   0          9m1s
real-ip-demo-8b55f64d-ztqxr      1/1     Running   0          9m1s
```

Step 3: 创建直连Pod模式的Service

在Orca Term中创建LoadBalancer Service并启用直连模式：

1.创建 Service YAML 文件（service.yaml）

```yaml
apiVersion: v1
kind: Service
metadata:
  name: clb-direct-pod
  annotations:
    service.cloud.tencent.com/direct-access: "true"  # 关键注解：启用直连
    service.cloud.tencent.com/loadbalance-type: "OPEN"  # 公网CLB
spec:
  selector:
    app: real-ip-app  # 匹配Deployment标签
  type: LoadBalancer
  ports:
    - protocol: TCP
      port: 80        # Service端口
      targetPort: 5000  # 匹配Deployment端口
```

⚠️ 核心参数说明

- `annotations.service.cloud.tencent.com/direct-access: "true"`：启用 CLB 直连 Pod

2.部署 Service

```
kubectl apply -f service.yaml
```

3.验证 Service 配置

```
kubectl describe svc clb-direct-pod
```

```
[root@VM-17-53-tlinux ~]# kubectl describe svc clb-direct-pod
Name:                     clb-direct-pod
Namespace:                default
Labels:                   service.cloud.tencent.com/loadbalance-type=OPEN
Annotations:              service.cloud.tencent.com/client-token: 162de8ad-b893-4e49-b682-60335b39def0
                          service.cloud.tencent.com/direct-access: true
                          service.cloud.tencent.com/loadbalance-type: OPEN
                          service.cloud.tencent.com/sync-begin-time: 2025-07-10T17:40:55+08:00
                          service.cloud.tencent.com/sync-end-time: 2025-07-10T17:40:56+08:00
                          service.kubernetes.io/loadbalance-id: lb-mhaytcha
Selector:                 app=real-ip-app
Type:                     LoadBalancer
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       172.19.253.69
IPs:                      172.19.253.69
LoadBalancer Ingress:     159.75.192.214 (VIP)
Port:                     <unset>  80/TCP
TargetPort:               5000/TCP
NodePort:                 <unset>  31052/TCP
Endpoints:                172.19.0.5:5000,172.19.0.69:5000,172.19.0.70:5000
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:
  Type    Reason                Age                    From                Message
  ----    ------                ----                   ----                -------
  Normal  EnsureServiceSuccess  2m13s (x3 over 2m17s)  service-controller  Sync Success. ReturnCode: S2000
```

确认以下输出：

- Annotations包含 `service.cloud.tencent.com/direct-access: "true"`
- `LoadBalancer Ingress`显示公网IP

Step 4: 验证真实源IP获取

1. 获取CLB公网IP

```
CLB_IP=$(kubectl get svc clb-direct-pod -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
echo "测试地址: http://$CLB_IP"
```

```
[root@VM-17-53-tlinux ~]# CLB_IP=$(kubectl get svc clb-direct-pod -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
[root@VM-17-53-tlinux ~]# echo "测试地址: http://$CLB_IP"
测试地址: http://159.75.192.214
```

2. Orca Term中快速测试（需业务支持IP回显）

```
curl -s http://$CLB_IP
```

```
[root@VM-17-53-tlinux ~]# curl -s http://$CLB_IP
{"headers":{"Accept":"*/*","Host":"159.75.192.214","User-Agent":"curl/7.61.1"},"message":"Here are your request headers","method":"GET","remote_addr":"172.19.0.65"}
```

若业务不支持支持IP回显，可直接curl+CLB公网IP

```
[root@VM-17-53-tlinux ~]# curl 159.75.192.214
{"headers":{"Accept":"*/*","Host":"159.75.192.214","User-Agent":"curl/7.61.1"},"message":"Here are your request headers","method":"GET","remote_addr":"172.19.0.65"}
```

3. 若有需要，可在外部设备验证

```
echo "请在外部设备执行:"
echo "  curl http://$CLB_IP"
echo "或浏览器访问 http://$CLB_IP"
```

预期结果：

- 回显内容包含客户端真实公网IP（非节点IP）
- 示例输出：`"remote_addr":"172.19.0.65"`

```
"message":"Here are your request headers","method":"GET","remote_addr":"172.19.0.65"}
```

验证技巧：

- 在手机5G网络下访问，确认IP与公网IP一致
- 对比`kubectl get nodes -o wide`显示的节点IP，确保不同

故障排查（Orca Term环境特供版）

现象	原因	解决方案
ConfigMap保存失败	vi操作不熟练	使用`kubectl patch`命令替代： `kubectl patch cm tke-service-controller-config -n kube-system --patch '{"data":{"GlobalRouteDirectAccess":"true"}}'`
Pod状态异常	镜像拉取失败	1. `kubectl describe pod <pod-name>`查看事件 2. 在Orca Term手动拉取：`docker pull <镜像>` 3. 检查镜像仓库权限
Service无公网IP	配额不足或注解错误	1. `kubectl describe svc`查看事件 2. 确认注解`direct-access: "true"`存在 3. 检查腾讯云账号CLB配额
访问返回节点IP	直连未生效	三重检查： 1. ConfigMap中GlobalRouteDirectAccess=true 2. Service注解direct-access=true
Orca Term连接断开	会话超时	1. 使用`tmux`创建持久会话 2. 关键操作前刷新Orca Term连接

清理资源

在Orca Term中释放资源避免费用：

1. 删除Service（保留Deployment可复用）

```
kubectl delete svc clb-direct-pod
```

2. 删除Deployment

```
kubectl delete deploy real-ip-demo
```

3. 可选：重置ConfigMap

```
kubectl patch cm tke-service-controller-config -n kube-system --patch
'{"data":{"GlobalRouteDirectAccess":"false"}}'
```

TKE环境最佳实践

1.Orca Term操作优化：
- 使用`watch`命令实时监控资源状态（如`watch -n 2 kubectl get pods`）
- 用`alias k=kubectl`简化命令输入
- 重要操作前创建屏幕快照（Orca Term截图功能）

2.安全建议：
- 为CLB配置安全组规则，限制访问源IP
- 定期轮转镜像仓库访问凭证
- 生产环境使用独立服务账号操作kubectl

3.性能监控：

```
# 直连模式性能检查
kubectl top pods -l app=real-ip-app
# 查看CLB监控指标（腾讯云控制台）
```