

作者：李晓辉

联系方式：

1. 微信：Lxh_Chat

2. 邮箱：939958092@qq.com

ingress和route挺厉害的，能让http类的工作负载被公开。不过呢，它俩也有局限性，要是碰到容器集里那些非http服务，比如ssh协议的22号端口这种，就没办法搞定了。这种时候，就得靠k8s里的Service来出马了。要是想让这些服务对外能统一访问，把Service设置成loadbalance类型就行啦。

不过呢，设置成loadbalance类型，还得有外部的负载均衡器来配合。像公司里那种F5设备，或者公有云里各种负载均衡服务都能用。但我们课程里没那些，所以就用metallb在裸金属集群里搞定负载均衡服务啦。说到这，k8s里几种常见的服务类型也该好好唠唠了。

k8S中的服务类型

内部沟通

在集群内部，大家都是自家人，用 ClusterIP 类型的服务就足够了。它就像一个内部的通讯频道，只在集群内部有效，对外面的世界是完全封闭的，外面的人根本找不到它，这样也能保证内部的安全和隐私。

外部沟通

要是想和外面的世界交流，就得用 NodePort 或 LoadBalancer 类型的服务了。这两种类型就像是集群对外的“窗口”，能让外面的人访问到集群里的服务。

NodePort 是一种比较简单的方式，它会在集群的每个节点上开放一个固定的端口（范围是 30000 到 32767），外面的人可以通过这个端口找到集群里的服务。不过，这种方式有点像“明码标价”，端口号是固定的，可能会比较容易被发现，而且配置起来也有点麻烦，因为要手动管理端口号。

LoadBalancer 类型就更高级了，它就像是一个智能的“门卫”，会自动帮你管理流量，把外部的请求合理地分配到集群里的各个服务上。这种方式更灵活，也更安全，不过需要外部的负载均衡器来配合，比如公有云里的负载均衡服务或者公司里的 F5 设备。

负载均衡服务案例

在使用服务之前，我们需要先部署一个后端，不然前端接受请求后，不知道发给谁

由于http类型的服务部署简单，我们就用nginx演示，你也可以做做教材上的视频摄像头练习

```
cat > deployment-service.yml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: lxh-pod-backend
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:latest
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 8080
EOF
```

```
oc create -f deployment-service.yml
```

部署一个type: LoadBalancer的服务，此服务工作在80端口，80端口收到请求时，会转发给具有app: nginx标签的pod中的8080端口

```
cat > loadbalancer.yml <<-EOF
apiVersion: v1
kind: Service
metadata:
  name: loadbalance-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: LoadBalancer
EOF
```

在我们的环境中，metallb默认提供了192.168.50网段的外部IP

```
[student@workstation ~]$ oc create -f loadbalancer.yml
service/loadbalance-service created
[student@workstation ~]$ oc get -f loadbalancer.yml
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
loadbalance-service	LoadBalancer	172.30.214.47	192.168.50.20	80:30140/TCP	3s

我们来试试从负载均衡IP发起请求

```
[student@workstation ~]$ curl 192.168.50.20
<html>
  <body>
    <h1>Hello, world from nginx!</h1>
  </body>
</html>
```

本文档在线版本: <https://www.linuxcenter.cn>