Porter - An Open Source Load Balancer for Bare Metal Kubernetes

Xuetao Song, Fang Du
Introduction
Service IP is an internal virtual-IP

In default mode (ClusterIP and kube-proxy in iptables)

- service can be used only in cluster
- the address of service is not bound to any device
# Expose services to external

<table>
<thead>
<tr>
<th>NodePort</th>
<th>Ingress</th>
<th>LoadBalancer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros:</strong></td>
<td><strong>Pros:</strong></td>
<td><strong>Pros:</strong></td>
</tr>
<tr>
<td><strong>Cons:</strong></td>
<td><strong>Cons:</strong></td>
<td><strong>Cons:</strong></td>
</tr>
<tr>
<td>1. Hard to remember</td>
<td>1. No Ingress for ingress controller</td>
<td>1. Available in cloud provider only</td>
</tr>
<tr>
<td>2. Single point and the bottle-neck problem</td>
<td>2. Single point and the bottle-neck problem</td>
<td></td>
</tr>
<tr>
<td>3. L7 use only</td>
<td></td>
<td>3. L7 use only</td>
</tr>
</tbody>
</table>
Porter is a porter to bring LoadBalancer to on-premise Kubernetes

1. Traffic load balance
   BGP dynamic configure

2. IP Access Management

3. Open Source
   Kubernetes Native
Fundamentals

02
ECMP (Equal-cost multi-path)

Porter uses ECMP to implement load balance.
BGP (Border Gateway Protocol)

BGP is the routing protocol for the Internet. Much like the post office processing mail, BGP picks the most efficient routes for delivering Internet traffic.
BGP ECMP

BGP support multipath, multipath allows you to install multiple internal BGP paths and multiple external BGP paths to the forwarding table. Selecting multiple paths enables BGP to load-balance traffic across multiple links.
2. Leaf publishes routes to spine via BGP
   1.1.1.1/32 nexthop
   <leaf1 ip>
   <leaf2 ip>

3. Spine publishes routes to border
   1.1.1.1/32 nexthop
   <spine1 ip>
   <spine2 ip>

1. Controller creates routes in its BGP server and sync to leaf
   1.1.1.1/32 nexthop
   192.168.0.2
   192.168.0.6

Node1 192.168.0.2
Node2 192.168.0.6

curl http://1.1.1.1
Dynamic Configure with zero downtime

When endpoints changed -> change the route

When bgp neighbours changed -> update the BGP server
Kubernetes Native

Using CRD to configure BGP
Using CRD to configure IPAM
Using Annotations to configure address of services

We can use kubectl or k8s client-go to configure porter
Usage
#!/bin/bash

#Apply yamls

kubectl apply -f https://github.com/kubesphere/porter/releases/download/v0.1.0/porter.yaml

#Configure neighbors

kubectl edit configmaps -n porter-system bgp-cfg
Sample Config

```json
[global.config]
  as = 65000
  router-id = "192.168.98.111"
  port = 17900

[port-remote-config]
  using-port-forward = true

[neighbors]
  [neighbors.config]
    neighbor-address = "192.168.98.5"
    peer-as = 65001

  [neighbors.add-paths.config]
    send-max = 8
```

Add a peer and using 17900 as BGP port
apiVersion: network.kubesphere.io/v1alpha1
kind: EIP
name: eip-sample
spec:
  address: 192.168.2.2
disable: false

Add a ip with address 192.168.2.2 to ippool
Create a service using porter

```yaml
kind: Service
apiVersion: v1
metadata:
  name: mylbapp
  annotations:
    lb.kubesphere.io/v1alpha1: porter
spec:
  selector:
    app: mylbapp
  type: LoadBalancer
  ports:
    - name: http
      port: 8088
      targetPort: 80
```

Using annotation to tell system to use porter for load-balancing
Plan

1. More flexible IP configure

2. Using other protocols or methods as optional way for users who do not support BGP

3. Support BGP policy to control route advertisement
Thank you

Q&A

https://github.com/kubesphere/porter