QINGCLOUD | KUBESPHERE

Porter - An Open Source Load Balancer for Bare Metal Kubernetes

Xuetao Song, Fang Du

01

Introduction

QINGCLOUD | KUBESPHERE

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

QINGCLOUD | KUBESPHERE

Expose services to external

NodePort

Pros:

1.Convenient

Cons:

1.Hard to remember

2.Single point and the bottle-neck problem

Ingress

Pros:

1.Understandable

Cons:

1. No Ingress for ingress controller

2. Single point and the bottle-neck problem

3. L7 use only

LoadBalancer

Pros:

1.Convenient

2.No single point and the bottle-neck problem

Cons:

1. Available in cloud provider only

QINGCLOUD | KUBESPHERE

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

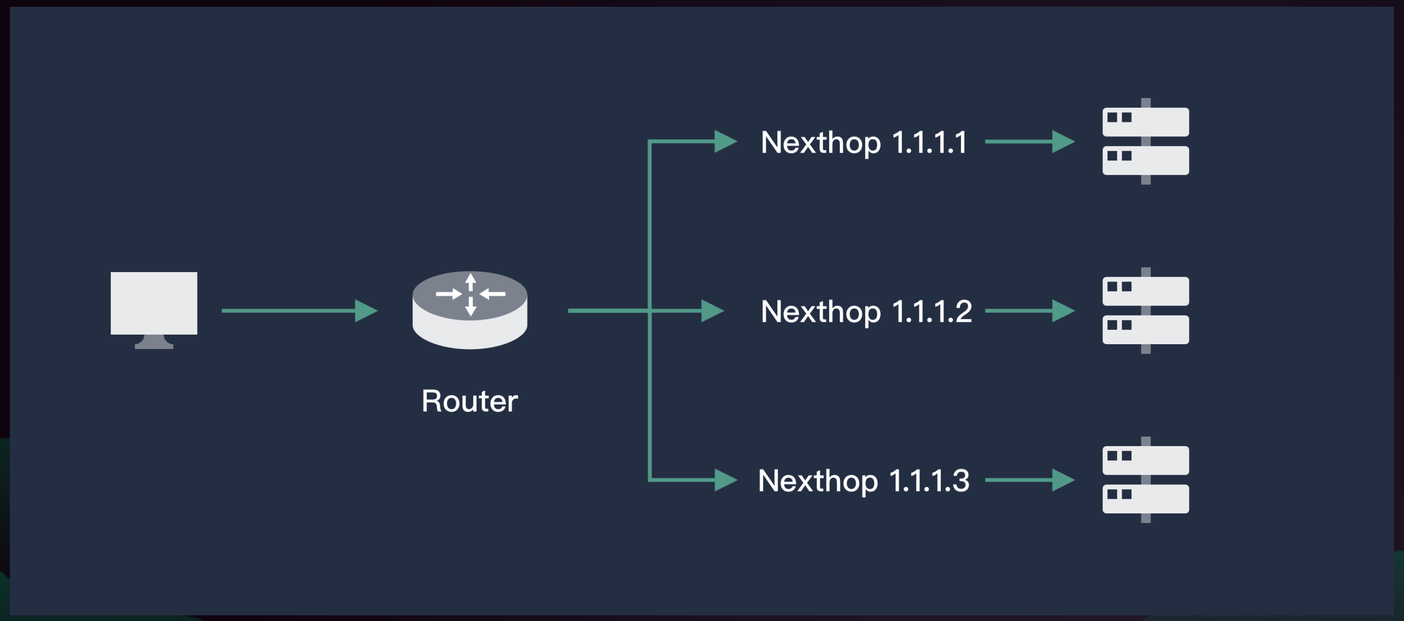
02

Fundamentals

QINGCLOUD | KUBESPHERE

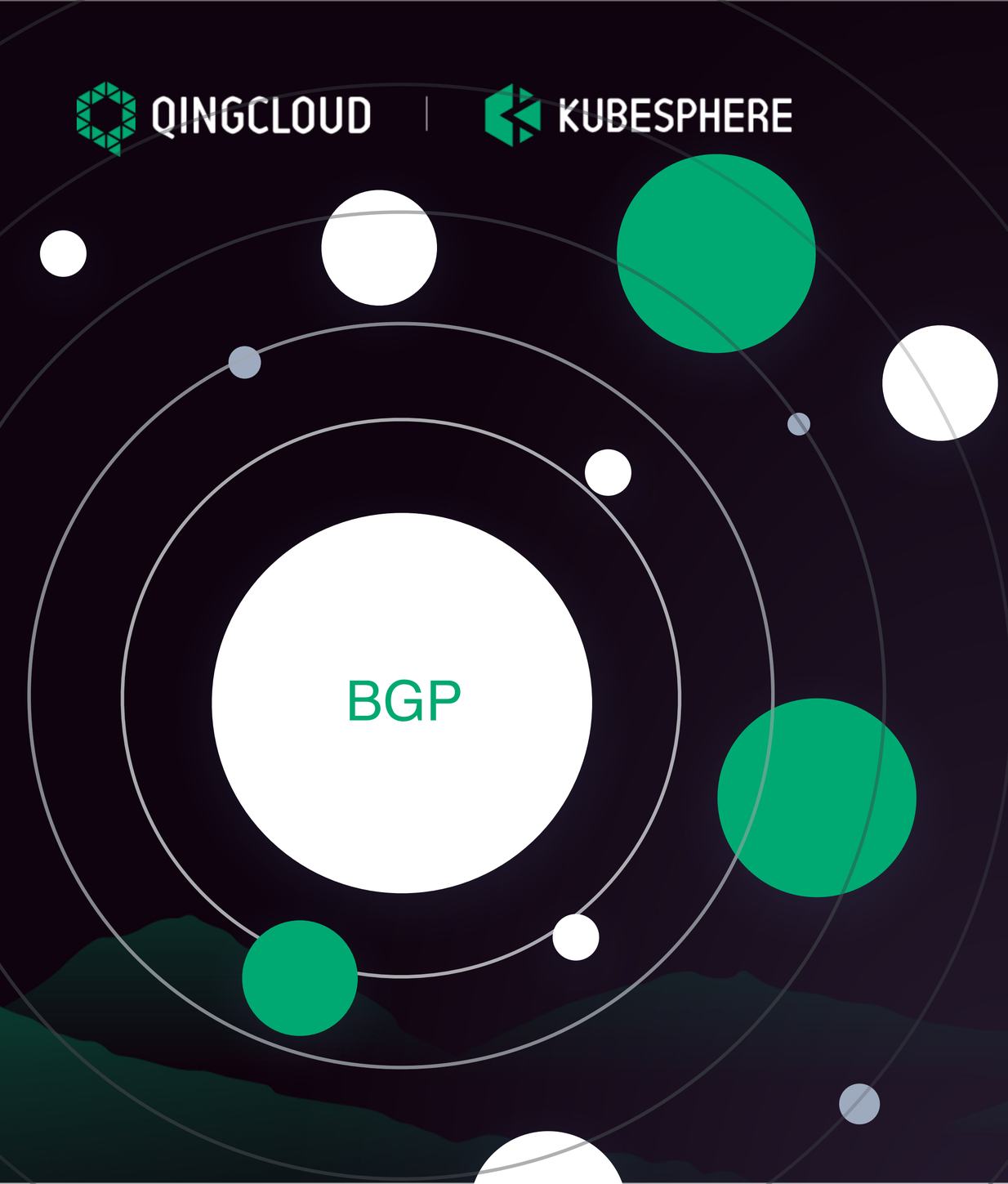
ECMP (Equal-cost multi-path)

Porter use ECMP to implement load balance



ECMP

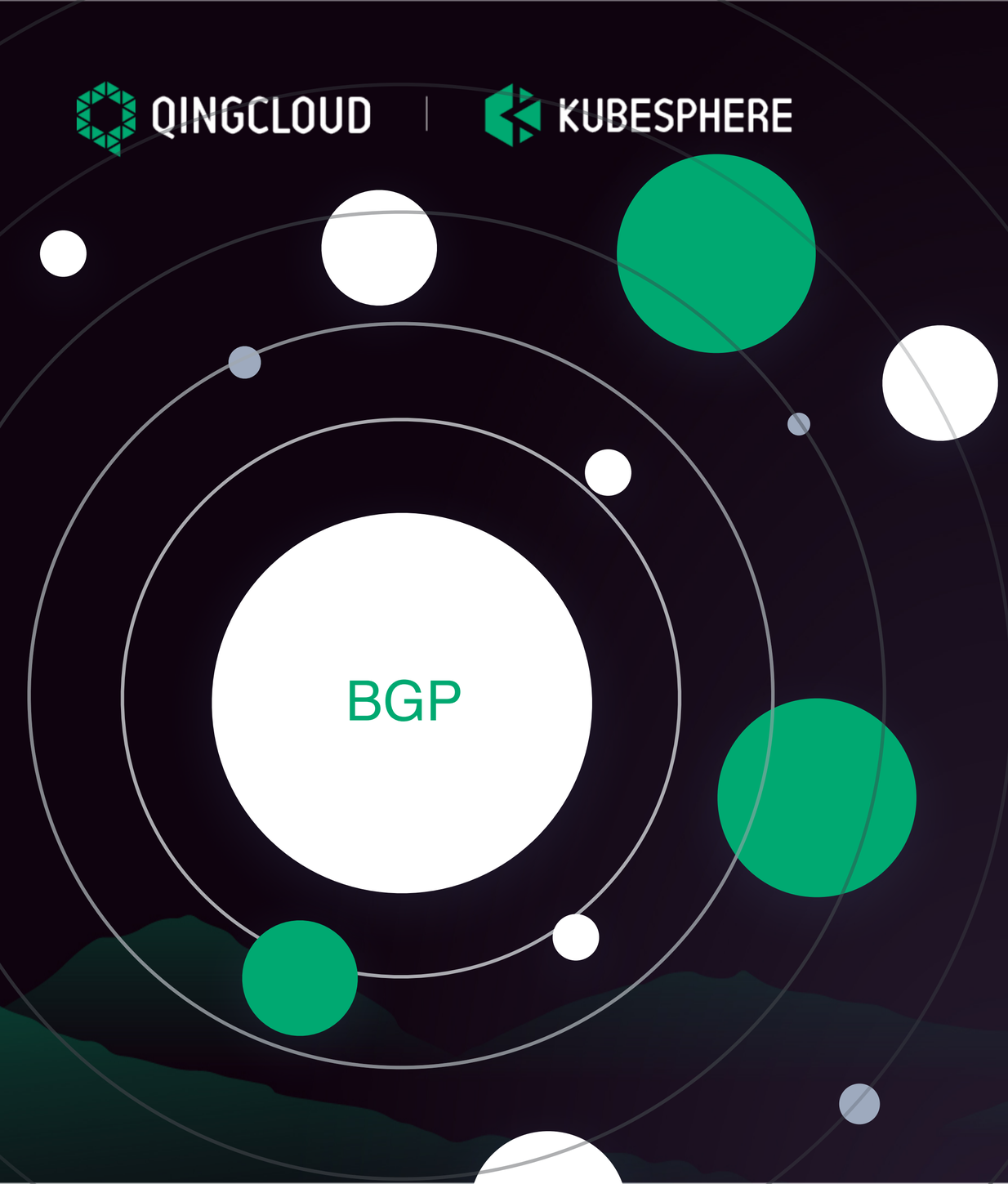
QINGCLOUD | KUBESPHERE



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.

QINGCLOUD | KUBESPHERE

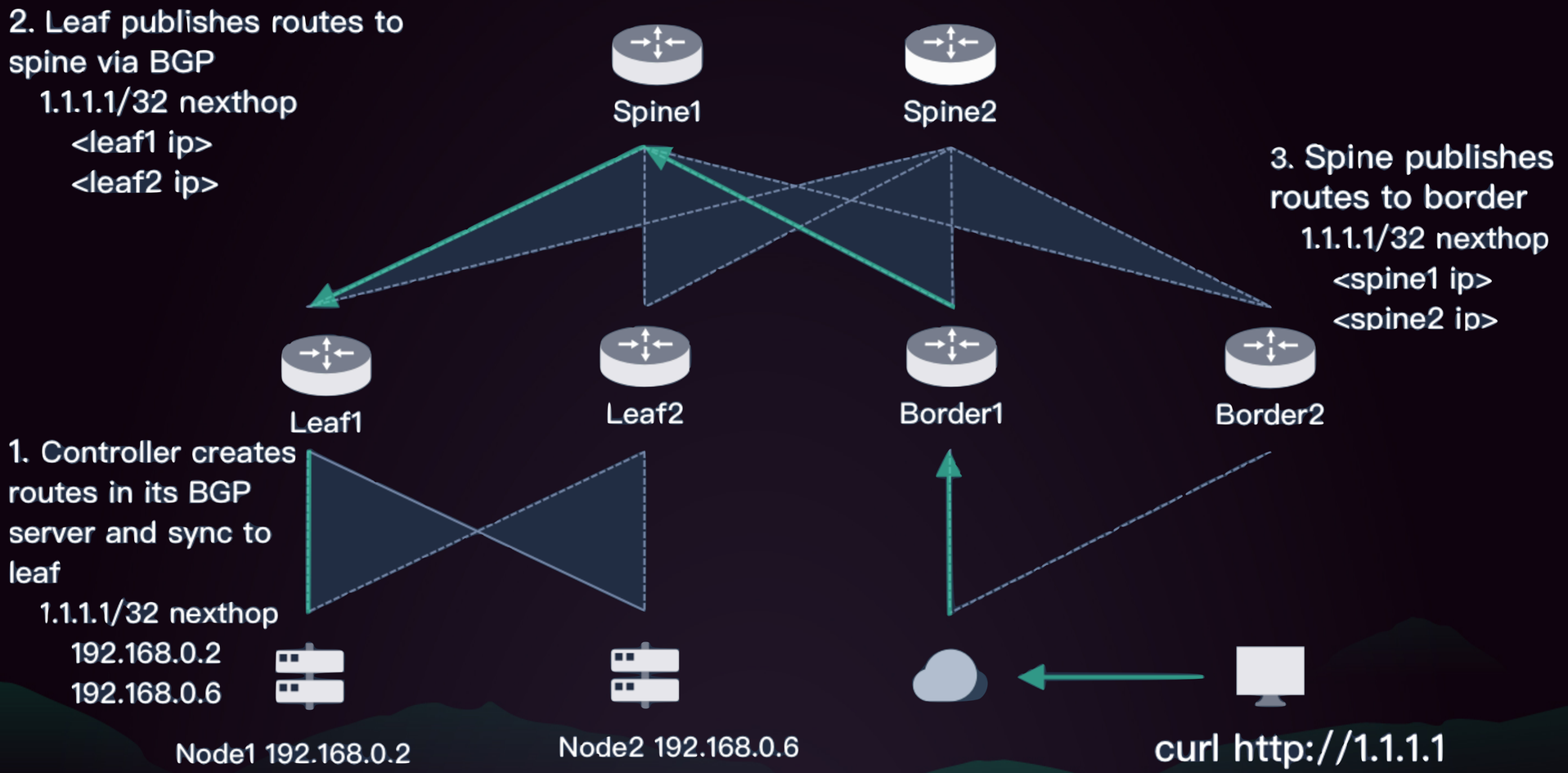


BGP ECMP

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

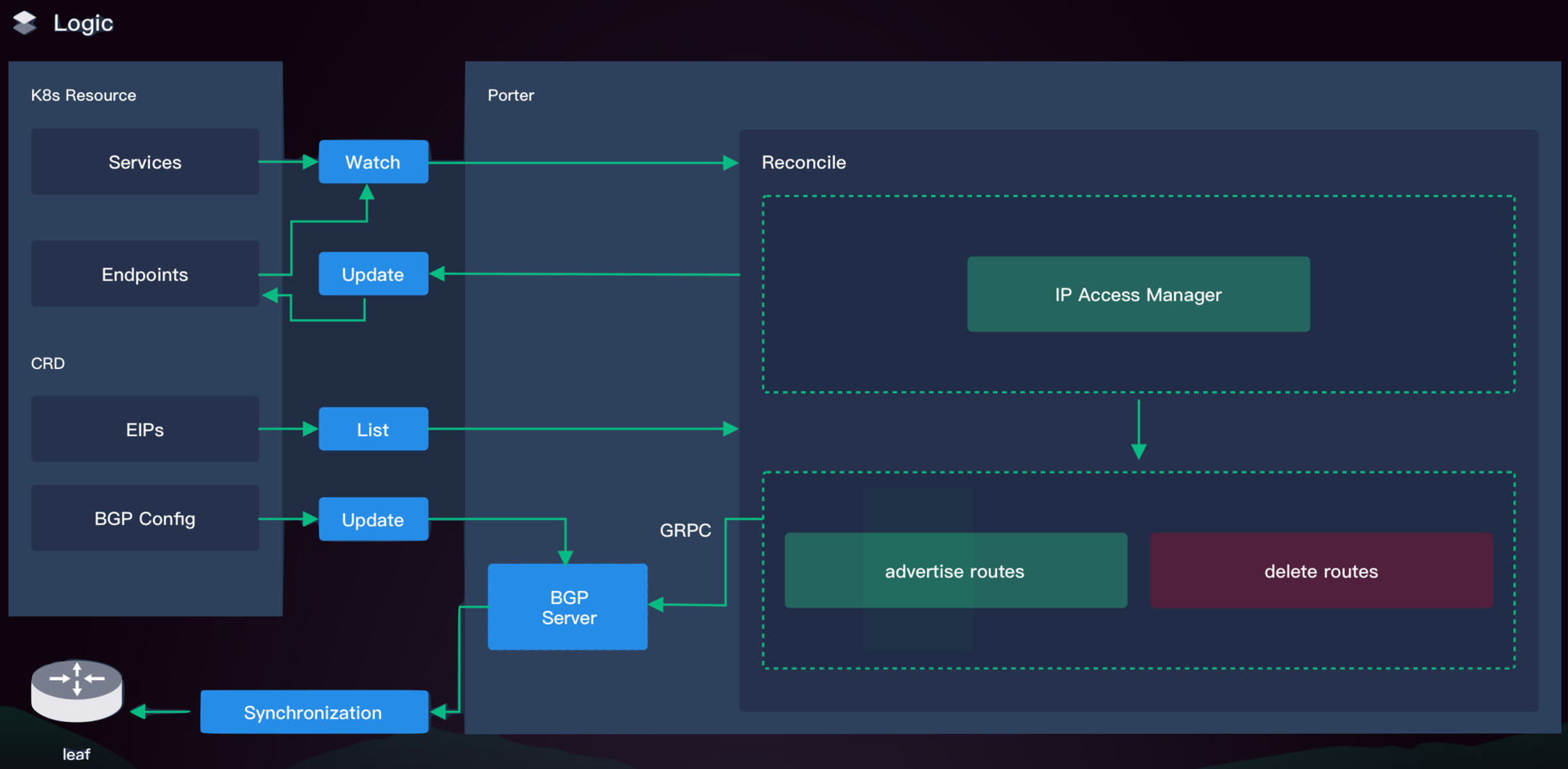
QINGCLOUD | KUBESPHERE

How BGP works



QINGCLOUD | KUBESPHERE

Workflow



QINGCLOUD | KUBESPHERE

Dynamic Configure with zero downtime

When endpoints changed -> change the route

When bgp neighbours changed -> update the BGP server

QINGCLOUD | KUBESPHERE

Kubernetes Native

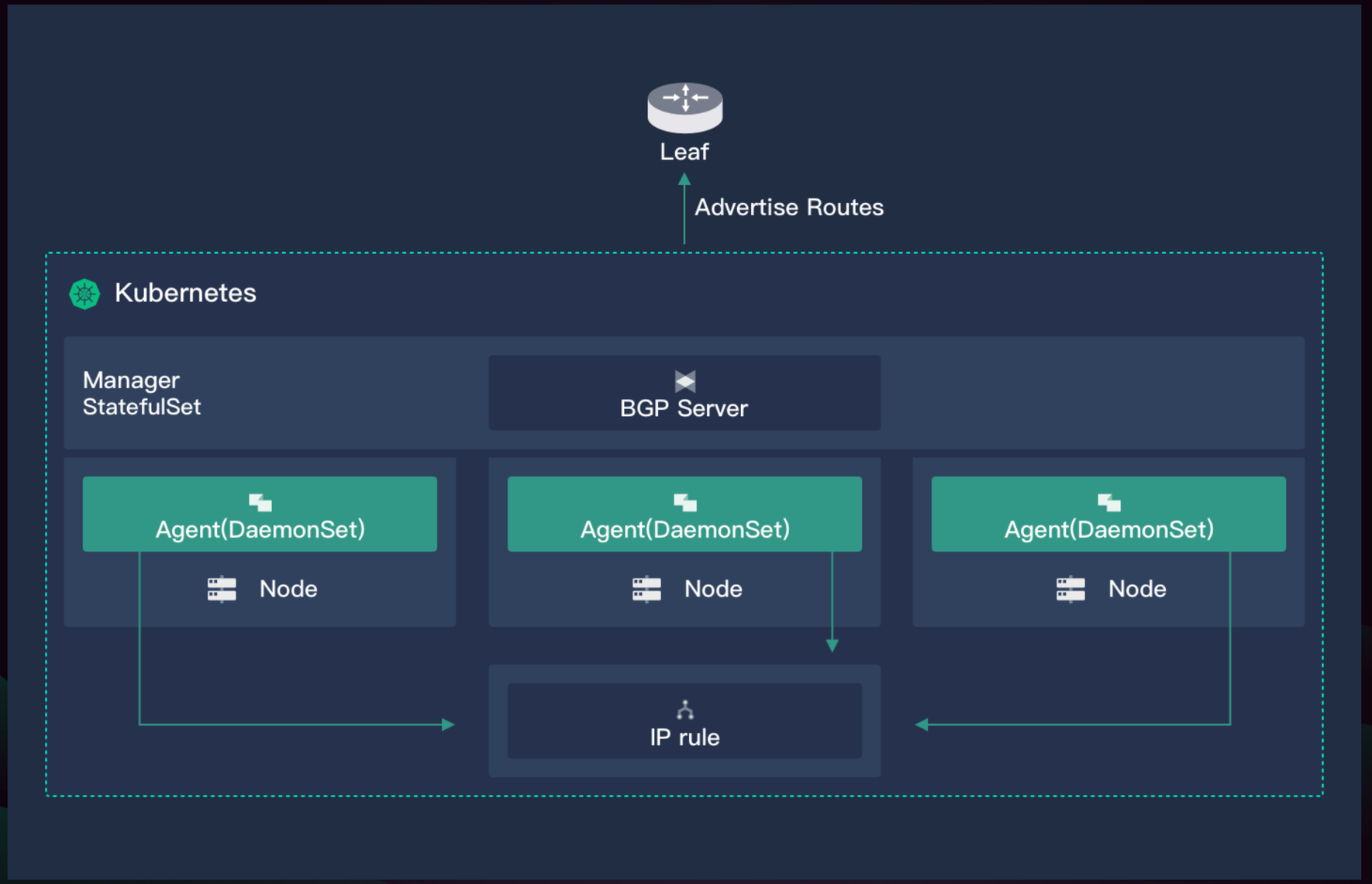
Using CRD to configure BGP

Using CRD to configure IPAM

Using Annotations to configure address of services

We can use kubectl or k8s client-go to configure porter

QINGCLOUD | KUBESPHERE



03

Usage

QINGCLOUD | KUBESPHERE

```
#!/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
```

QINGCLOUD | KUBESPHERE

Sample Config

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

QINGCLOUD | KUBESPHERE

Add an ip to pool

```yaml
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

QINGCLOUD | KUBESPHERE

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

QINGCLOUD | KUBESPHERE

Plan

1. More flexible ip configure
2. Using other protocols or methods as optional way for users do not support BGP
3. Support BGP policy to control route advertisement

QINGCLOUD | KUBESPHERE

Thank you

Q&A

https://github.com/kubesphere/porter