The bridge to possible

Kubernetes (K8s) Infrastructure Connectivity

Network Designs for the Modern Data Center

Shangxin Du, Technical Marketing Engineer, Cloud Networking
Camillo Rossi, Technical Leader, Cloud Networking

Cisco Webex App

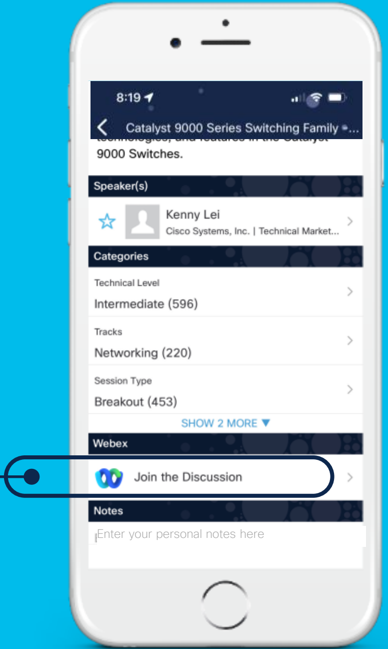
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

**Webex spaces will be moderated
until February 24, 2023.**



8:19

Catalyst 9000 Series Switching Family ▪...

9000 Switches.

Speaker(s)

☆ Kenny Lei
Cisco Systems, Inc. | Technical Market...

Categories

Technical Level
Intermediate (596)

Tracks
Networking (220)

Session Type
Breakout (453)

SHOW 2 MORE ▼

Webex

Join the Discussion

Notes

Enter your personal notes here

Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

Agenda

- **Kubernetes Refresh**
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

Kubernetes Refresh

Kubernetes - pod

- A pod is the scheduling unit in Kubernetes. It is a logical collection of one or more containers which are always scheduled together.
- The set of containers composed together in a pod share an IP.

```
[root@k8s-01-p1 ~]# kubectl get pod  --namespace=kube-system
NAME                                      READY     STATUS    RESTARTS   AGE
aci-containers-controller-1201600828-qsw5g   1/1       Running   1          69d
aci-containers-host-lt9kl                 3/3       Running   0          72d
aci-containers-host-xnwkr                 3/3       Running   0          58d
aci-containers-openvswitch-0rjbw          1/1       Running   0          58d
aci-containers-openvswitch-7j1h5          1/1       Running   0          72d
```

Kubernetes – Deployment

- Deployments are a collection of pods providing the same service
- You describe the desired state in a Deployment object, and the Deployment controller will change the actual state to the desired state at a controlled rate for you
- For example you can create a deployment that declare you need to have 2 copies of your front-end pod.

```
[root@k8s-01-p1 ~]# kubectl get deployment --namespace=kube-system
NAME                         DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
aci-containers-controller    1         1         1            1           72d
```

Kubernetes – Services

- A service tells the rest of the Kubernetes environment (including other pods and Deployments) what services your application provides.
- While pods come and go, the service IP addresses and ports remain the same.
- Kubernetes automatically load balance the load across the replicas in the deployment that you expose through a Service
- Other applications can find your service through Kubernetes service discovery.
  - Every time a service is create a DNS entry is added to kube-dns

```
[root@k8s-01-p1 ~]# kubectl get svc --namespace=kube-system
NAME        CLUSTER-IP   EXTERNAL-IP   PORT(S)         AGE
kube-dns    11.96.0.10   <none>        53/UDP,53/TCP   72d
```

Kubernetes – External Services

- If there are external IPs that route to one or more cluster nodes, Kubernetes services can be exposed on those external IPs.
- Traffic that ingresses into the cluster with the external IP (as destination IP), on the service port, will be routed to one of the service endpoints.
- External IPs are not managed by Kubernetes and are the responsibility of the cluster administrator.

```
[root@k8s-01-p1 ~]# kubectl get svc front-end --namespace=guest-book
NAME         CLUSTER-IP    EXTERNAL-IP    PORT(S)         AGE
front-end    11.96.0.33    11.3.0.2       80:30002/TCP    3m
```

Kubernetes - Annotations

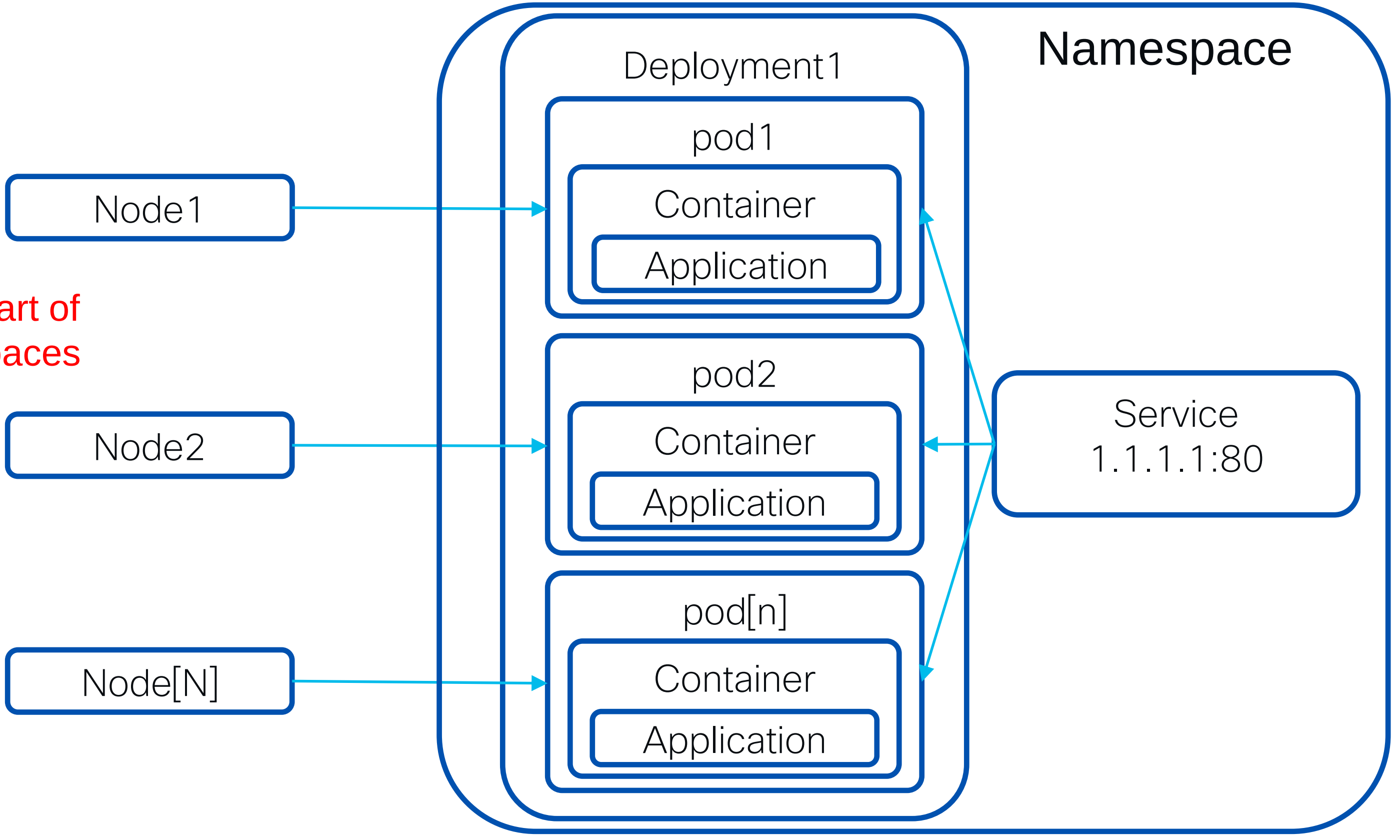
- Similar to labels but are NOT used to identify and select object

```
[root@k8s-01-p1 ~]# kubectl describe node k8s-01-p1 | more
Name:                  k8s-01-p1
Role:
Labels:                      beta.kubernetes.io/arch=amd64
                       beta.kubernetes.io/os=linux
                       kubernetes.io/hostname=k8s-01-p1
                       node-role.kubernetes.io/master=
Annotations:                 node.alpha.kubernetes.io/ttl=0
                       opflex.cisco.com/pod-network-ranges={"V4":[{"start":"11.2.0.130","end":"11.2.1.1"}]}
                       opflex.cisco.com/service-endpoint={"mac":"66:85:9a:e9:ef:2f","ipv4":"11.5.0.3"}
                       volumes.kubernetes.io/controller-managed-attach-detach=true
```

Kubernetes – Namespace

- Groups everything together:
  - Pod
  - Deployment
  - Volumes
  - Services
  - Etc….

All Together: A K8S Cluster



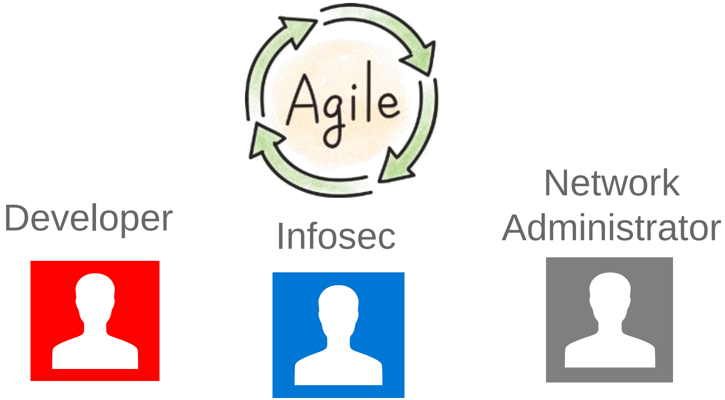
A node can be part of
Several Namespaces

Agenda

- Kubernetes Refresh
- **Kubernetes Network Challenges**
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

Operations and Visibility

- Skills gap between network and Kubernetes admins
- Visibility
  - Encapsulated traffic between K8s nodes hides the POD-to-POD Communications
- Governance of network policies



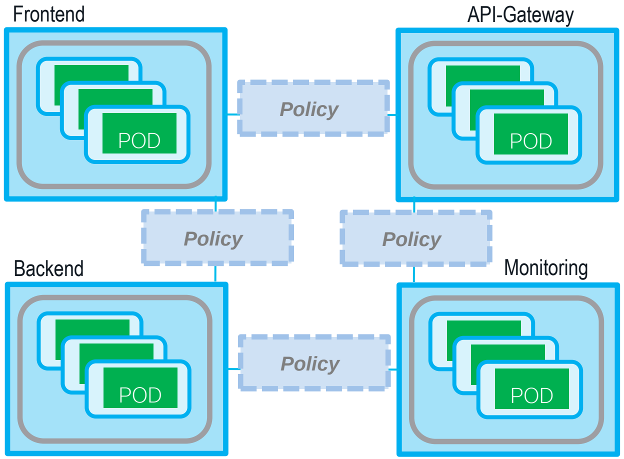
Developer

Infosec

Network Administrator

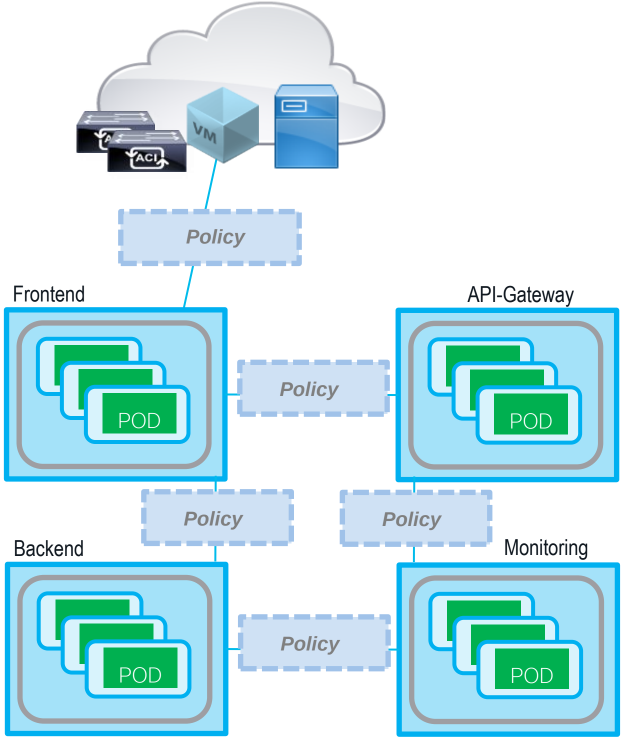
Segmentation

- Secure K8s **infrastructure**:
  - network isolation for kube-system and other infrastructure related objects (i.e. heapster, hawkular, etc.)
- Network isolation between **namespaces**



Communications outside of the Cluster

- Non-Cluster endpoints communicating with Cluster:
  - Exposing external services, how? NodePort? LoadBalancer?
  - Scaling-out ingress controllers?
- Cluster endpoints communicating with non-cluster endpoints:
  - POD access to external services and endpoints
- Cluster accessing shared resources like Storage



“Outsourcing the issue” - Container Networking Interface

CNI

- A generic plugin-based networking solution for application containers on Linux
- The spec defines a container as being a Linux network namespace
- The plugin must connect containers to networks and is responsible for IPAM and DNS configurations.

Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- **Design Kubernetes Network on ACI fabric**
  - **ACI CNI**
  - BGP-based CNI
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

ACI-CNI
Solution Overview

Why ACI-CNI for Application Container Platforms

Turnkey solution for node and container connectivity

Flexible policy: Native platform policy API and ACI policies

Hardware-accelerated: Integrated load balancing and Source NAT

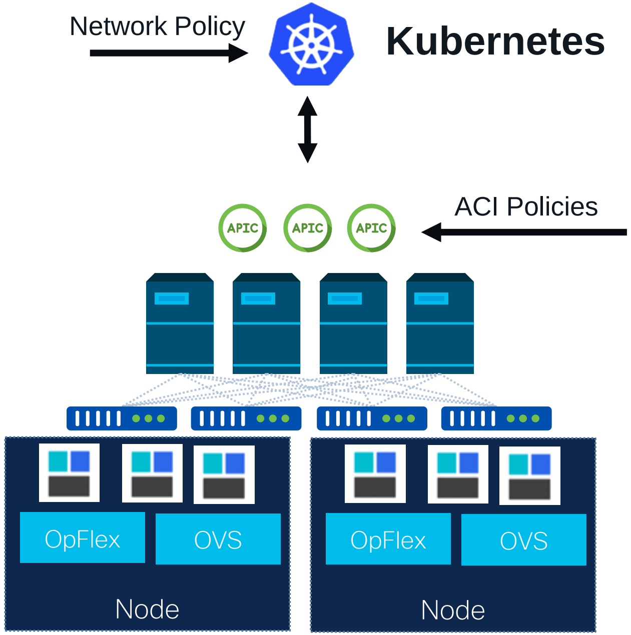
Visibility: Live statistics in APIC per container and health metrics

Enhanced Multitenancy and unified networking for containers, VMs, bare metal

Fast, easy, secure and scalable **networking** for your Application Container Platform

Cisco ACI CNI plugin features

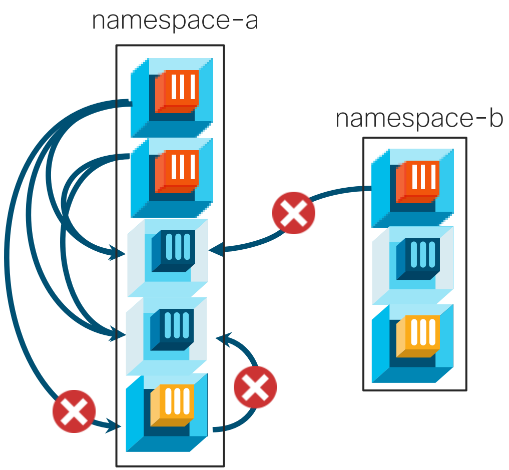
- IP Address Management for Pods and Services
- Distributed Routing and Switching with integrated VXLAN overlays implemented fabric wide and on Open vSwitch
- Distributed Firewall for implementing Network Policies
- EPG-level segmentation for K8s objects using annotations
- Consolidated visibility of K8s networking via VMM Integration



ACI-CNI:
Key Features

Support for Network Policy in ACI

- Specification of how selections of pods are allowed to communicate with each other and other network endpoints.
- Network namespace isolation using defined labels
  - directional: allowed ingress pod-to-pod traffic
  - filters traffic from pods in other projects
  - can specify protocol and ports (e.g. tcp/80)

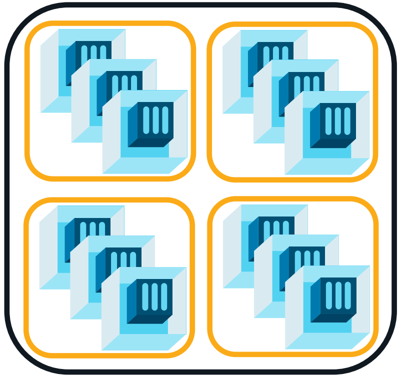


Policy applied to namespace: namespace-a

```
kind: NetworkPolicy
apiVersion: extensions/v1beta1
metadata:
  name: allow-red-to-blue-same-ns
spec:
  podSelector:
    matchLabels:
      type: blue
  ingress:
  - from:
    - podSelector:
        matchLabels:
          type: red
```

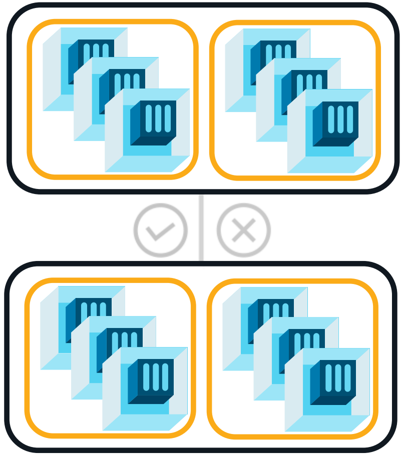
Mapping Network Policy and EPGs

Cluster Isolation



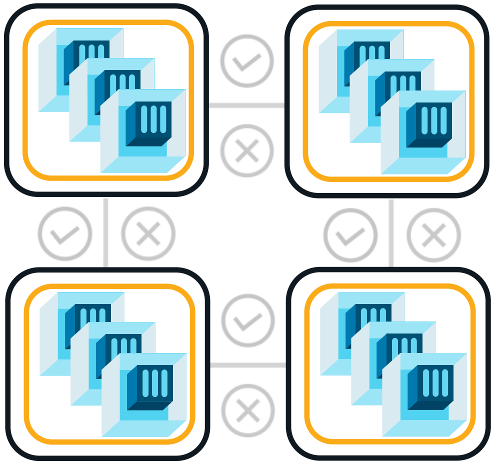
Single EPG for entire cluster.
(Default behavior)
No need for any internal contracts.

Namespace Isolation



Each namespace is mapped to its own EPG.
Contracts for inter-namespace traffic.

Deployment Isolation



Each deployment mapped to an EPG
Contracts tightly control service traffic

Key Map | EPG | NetworkPolicy | ✓ | ✗ Contract

Automated LoadBalancing

- Create a service of type “LoadBalancer” (as per K8s standard)
- ACI CNI will:
  - Allocate an external IP from a user-defined subnet
  - Deploy a Service Graph with PBR redirection to LoadBalance the traffic between any K8s Nodes that have PODs for the exposed service

```
cisco@k8s-01:~/demo/guestbook1$ kubectl --namespace=guestbook get svc frontend
NAME        CLUSTER-IP    EXTERNAL-IP   PORT(S)         AGE
frontend    10.37.0.124   10.34.0.5     80:32677/TCP    5h
```

extern_ip

POD SNAT

- POD Initiated traffic can be natted to an IP address selected by the user
  - SNAT IP: Single IP or Range
  - Ability to apply SNAT Policy at different levels:
    - Cluster Level: connection initiated by any POD in any Namespaces is natted to the selected SNAT IP
    - Namespace: connection initiated by any POD in the selected Namespaces is natted to the selected SNAT IP
    - Deployment: connection initiated by any POD in the selected Deployment is natted to the selected SNAT IP
    - LoadBalanced Service: connection initiated by any POD mapped to an external Service of Type LoadBalance are natted to the external Service IP.

Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- **Design Kubernetes Network on ACI fabric**
  - ACI CNI
  - **BGP-based CNI**
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

BGP Based Integration Benefits – why?

1. **Relies on a well-established protocol (BGP)**
2. **Unified networking:** Node, Pod and Service endpoints are accessible from an L3OUT providing easy connectivity across and outside the fabric
3. **(Limited) Security:** ability to use external classification to secure communications to Node/Pod/Service Subnets (no /32 granularity)
4. **High performance:** low-latency connectivity without egress routers if no Overlay are used
5. **Hardware-assisted load balancing:** ECMP up to 64 paths/Nodes
6. **Any Fabric/Hypervisor/Bare Metal:** allows to mix form factors together

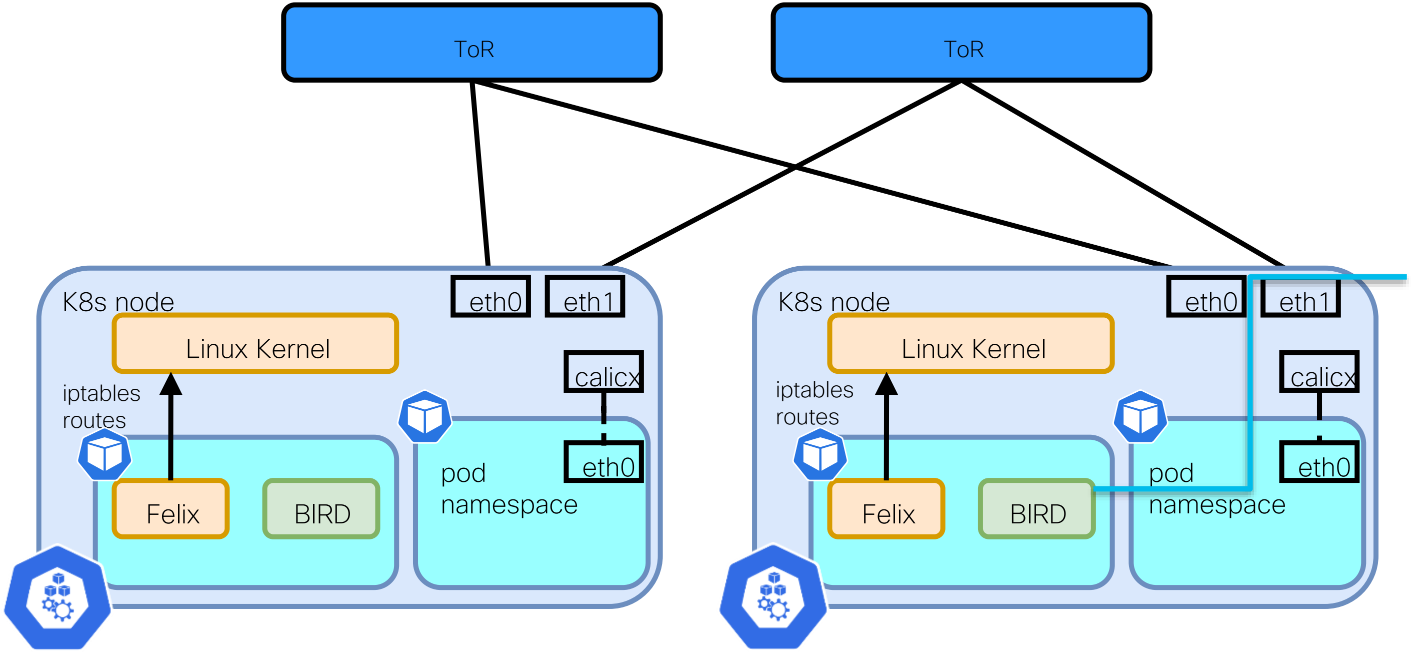
Configuration Goals

- Every Nodes, PODs and Service IPs Subnets will be advertised to the Fabric
  - K8s nodes is allocated one or more subnets from the POD Supernet. Each subnets is advertised to the Fabric as well
- Exposed Services will be advertised to the Fabric as host routes from every nodes that has a running POD associated to the service.
- K8s nodes use the Fabric as default gateway for ease of cluster bootstrapping (No need to have BGP running at config time)

Calico

Calico

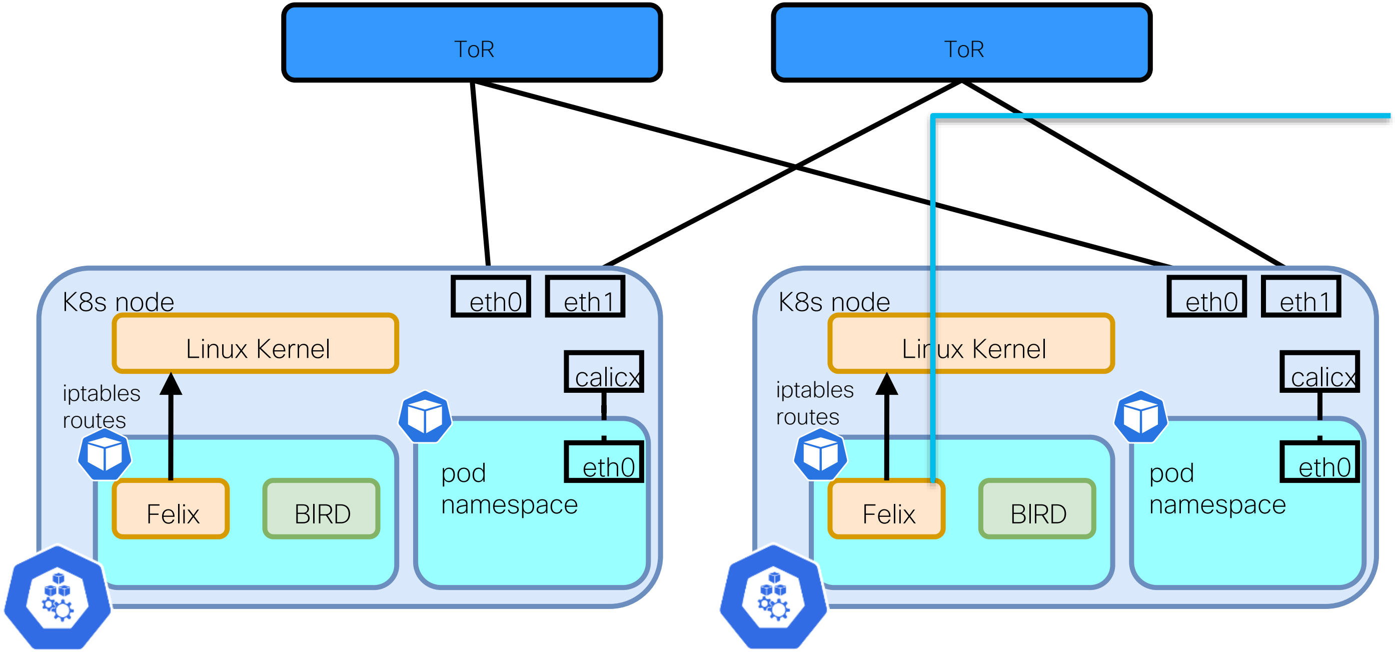
A CNI plugin of Kubernetes



BIRD: It is a routing daemon responsible for peering with other K8s nodes and exchanging routes of pod network and service network for inter-node communication.

Calico

A CNI plugin of Kubernetes

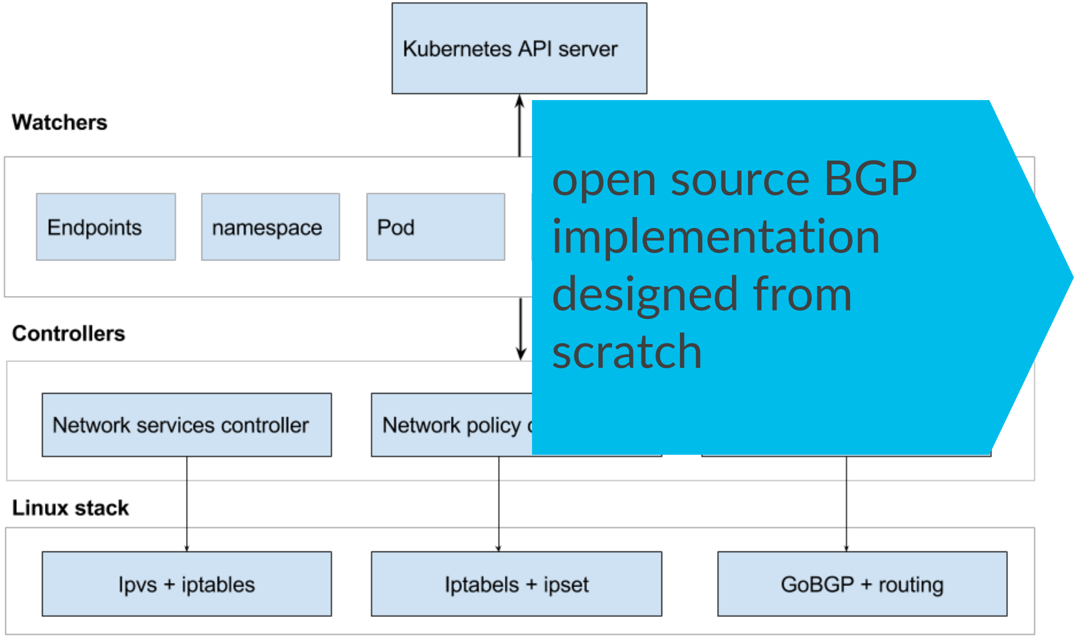


Felix: Running in same pod as BIRD, programs routes and ACLs (iptables) and anything required on Calico node to provide connectivity for the pods scheduled on that node

Kube-Router

Kube Router

A CNI plugin of Kubernetes



open source BGP implementation designed from scratch

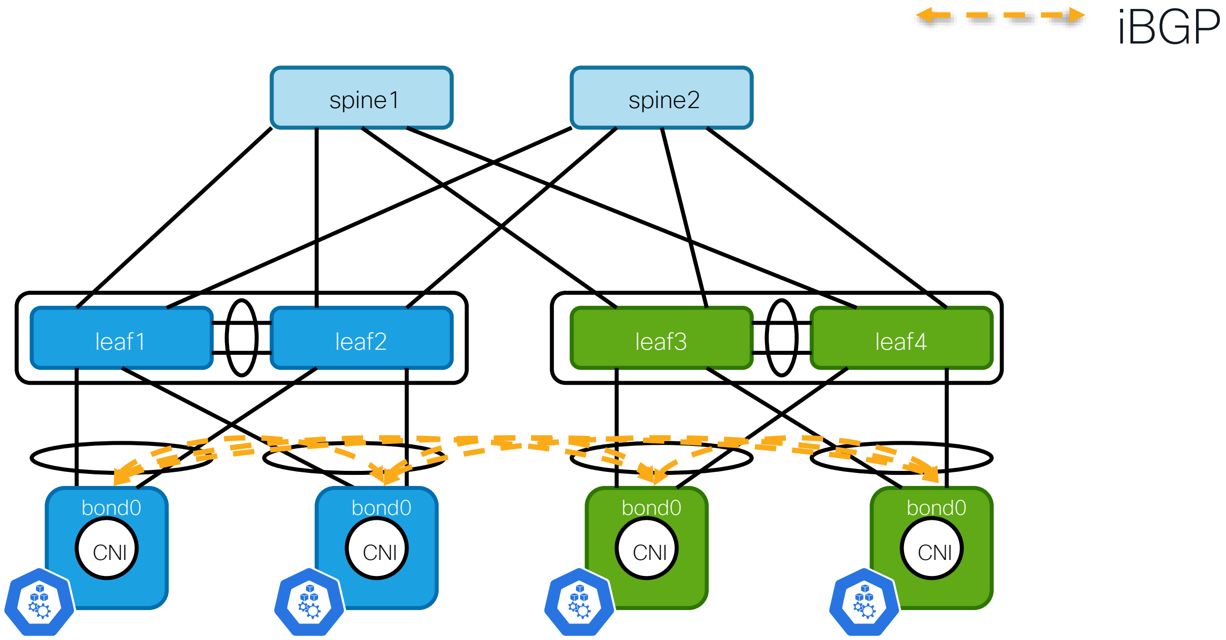
- Kube-router is built around concept of watchers and controllers.
- GoBGP Runs inside the KubeRouter POD
    - Injects learned routes into local Node routing table

Non-Fabric Integrated CNI

A tale of scalability issues and limitations

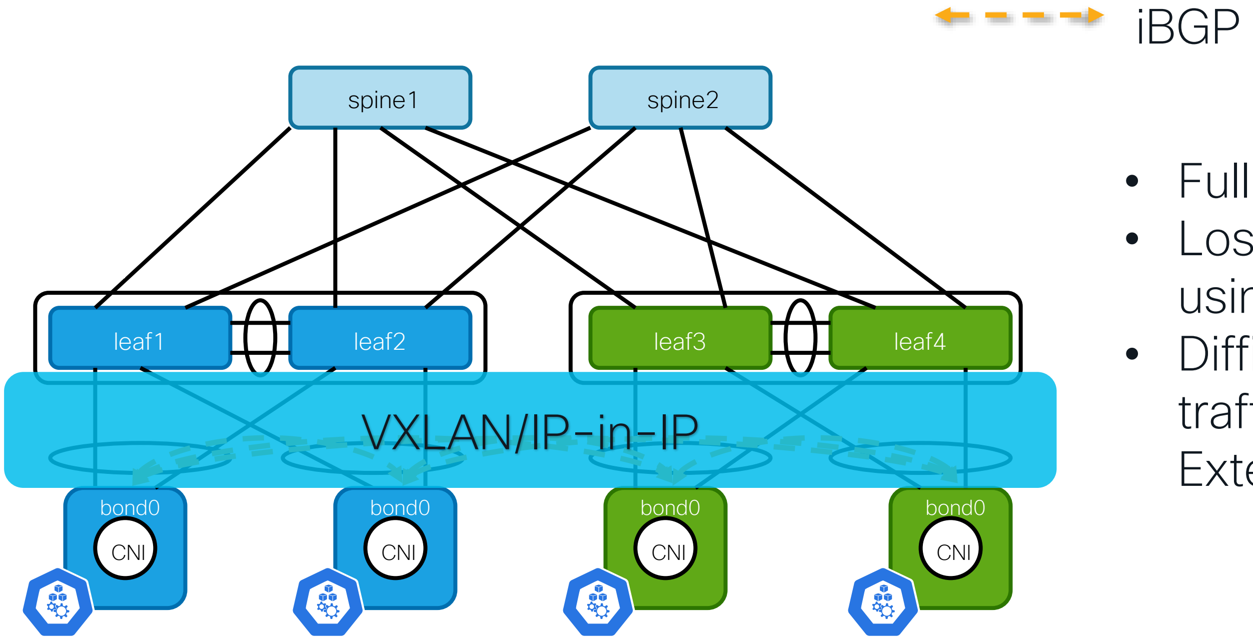
Network Architecture

Full mesh



Network Architecture

Full mesh data plane



- Full mesh does not scale!
- Losing visibility when using software overlay
- Difficult to LoadBalance traffic between ExternalServices

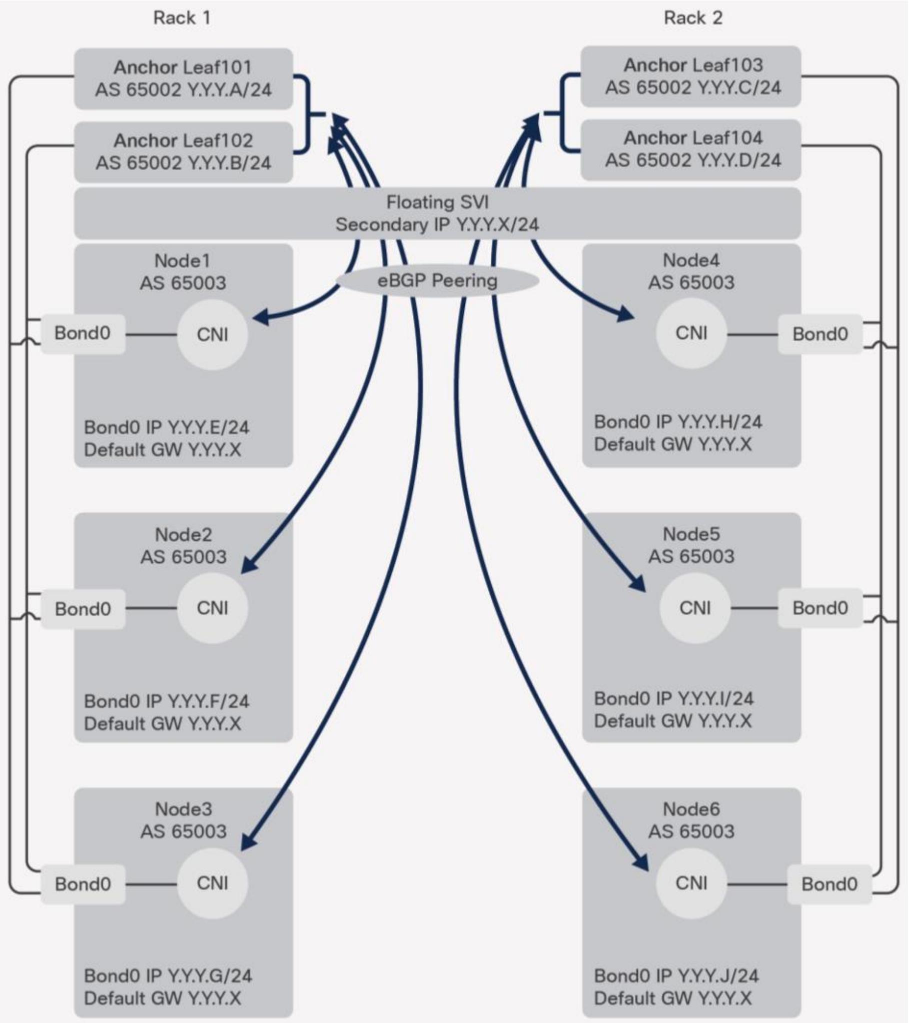
Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- **Design Kubernetes Network on ACI fabric**
  - ACI CNI
  - BGP-based CNI
  - **Design BGP network on ACI**
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

ACI BGP Based Architecture

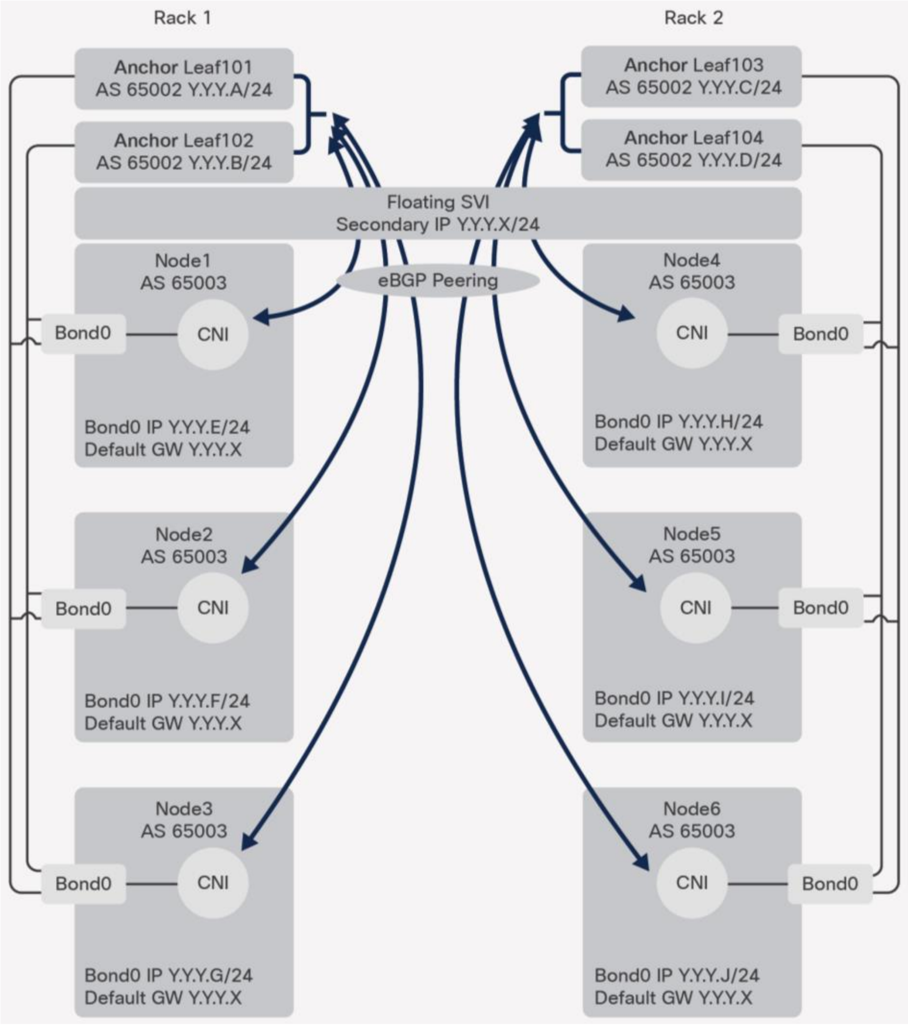
Architecture

- Each K8s Node will peer with a pair of border leaves
- Single AS for the whole cluster
  - Simpler ACI config (can use a subnet for passive peering)



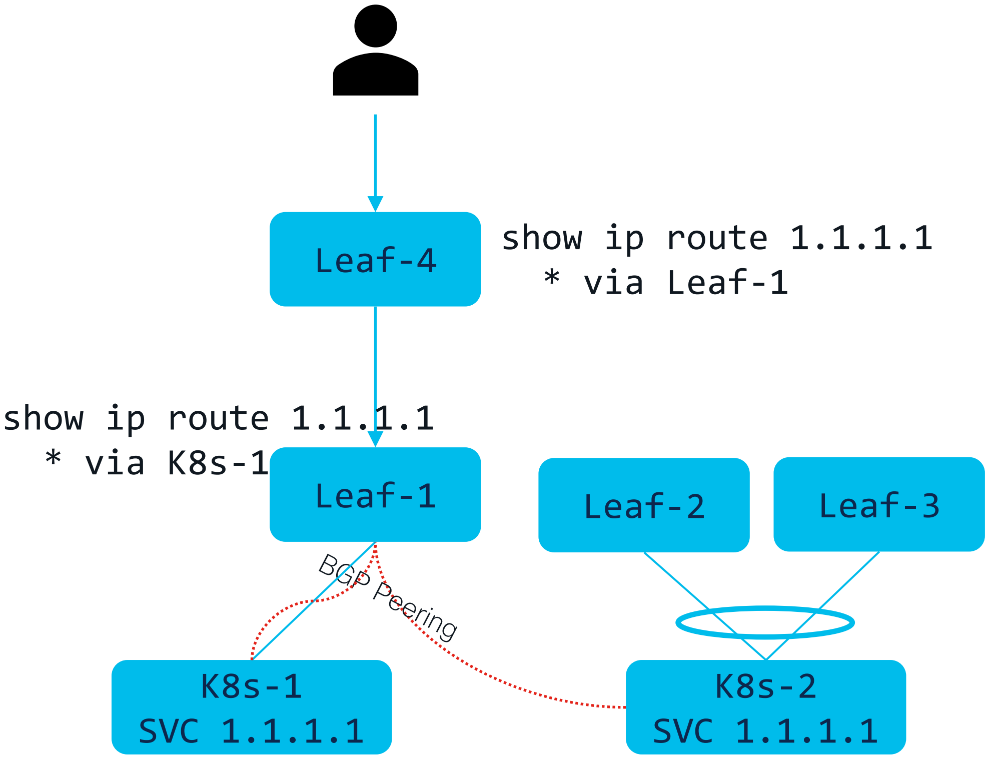
L3OUT Design

- K8s Nodes are connected to an L3OUT via vPC
  - External EPGs can be used to classify the traffic coming from the cluster
- Floating L3OUT
  - VM Mobility
  - Ability to mix BareMetal and VMs running on any hypervisor



ACI Best Practice: Peer to local ToR

- If some K8s nodes are connected to **Anchor** and some to **Non-Anchor** Leaves and are advertising the same Service IP only the one connected to the **Anchor** Leaves are selected as valid next hop.
- This happens because the Next-Hop cost is higher for to Non-Anchor Leaves connected K8s Nodes.
- We are working to address this in an upcoming ACI release



```
show ip route 1.1.1.1
  * via Leaf-1
```

```
show ip route 1.1.1.1
  * via K8s-1
```

ACI BGP Tuning

The following tunings are required:

- AS override and Disable Peer AS Check: To support having a single AS per cluster without the presence of Route Reflectors or Full Mesh inside the cluster
- BGP Graceful Restart
- BGP timers tuned to 1s/3s for quick eBGP node down detection for.
  - Note: There is currently an issue with Calico(1) and you should use 10s/30s
- Increase Max BGP ECMP path to 64 for better load balancing

ACI BGP Hardening (Optional)

- Enabled BGP password authentication
- Set the maximum AS limit to one
- Configure BGP import route control to accept only the expected subnets from the Kubernetes cluster:
  - Pod subnet(s)
  - Node subnet(s)
  - Service subnet(s)
- Set a limit on the number of received prefixes from the nodes.

CNI Configuration Examples: Calico

Calico BGP Config

The following Calico configurations objects are required

- One or more IPPool with all overlays disabled
- BGPConfiguration with:
  - nodeToNodeMeshEnabled set to “false”
  - List of serviceClusterIPs and serviceExternalIPs subnets to enabled host routes advertisement for those subnets
- BGPPeer to define the BGP Peer the K8s nodes connects to
- A Secret, Role and RoleBinding to pass the BGP Password to the Calico BGP Process

Calico IPPool Config – Cont.

```
apiVersion:
crd.projectcalico.org/v1
kind: IPPool
metadata:
name: default-ipv4-ippool
spec:
  blockSize: 26
  cidr: 192.168.3.0/24
  ipipMode: Never
  nodeSelector: all()
  vxlanMode: Never
```

blockSize: 26 → How to split the POD subnet between nodes

cidr: 192.168.3.0/24 → POD Subnet

ipipMode: Never → Disable IP in IP

nodeSelector: all() → Allocate this Subnet to all the nodes

vxlanMode: Never → Disable VXLAN Overlay

Calico BGPConfiguration – Cont.

```
apiVersion:
crd.projectcalico.org/v1
kind: BGPConfiguration
metadata:
  name: default
spec:
  asNumber: 65003
  listenPort: 179
  logSeverityScreen: Info
  nodeToNodeMeshEnabled: false
  serviceClusterIPs:
  - cidr: 192.168.4.0/24
  serviceExternalIPs:
  - cidr: 192.168.5.0/24
```

K8s Cluster BGP AS

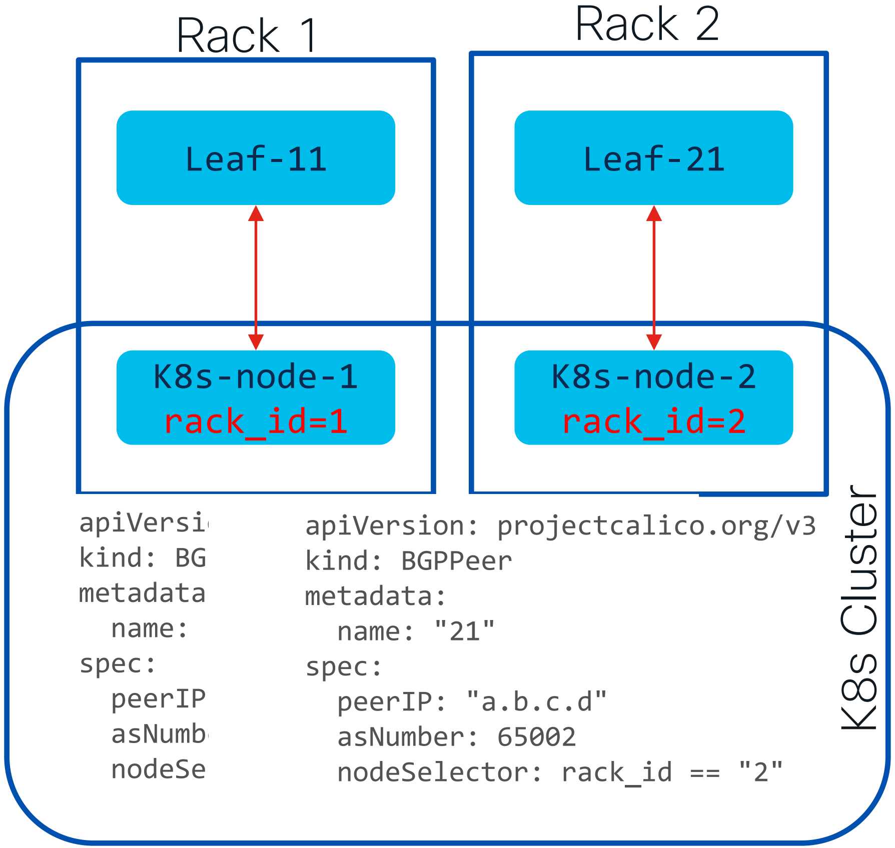
BGP Port

Disable iBGP Full Mesh Peering

Allow Calico to Advertise the Cluster and External service Subnets

How do I peer with the “local” TORs?

- Use a Node label to identify the location of the K8s Node, for example the rack id
- Configure the BGPPeer resource with a nodeSelector matching the label of the K8s Nodes
- The result will be that the peering is happening only between K8s Nodes and leaves with a matching rack id



```
apiVersi
kind: BG
metadata
  name:
spec:
  peerIP
  asNumb
  nodeSe
```

```
apiVersion: projectcalico.org/v3
kind: BGPPeer
metadata:
  name: "21"
spec:
  peerIP: "a.b.c.d"
  asNumber: 65002
  nodeSelector: rack_id == "2"
```

CNI Configuration Examples: Kube-Router

Kube-Router eBGP Config

- Most of the configuration is applied at the Kube-Router DaemonSet level, for our design we need the following options

```
--run-router=true                                --advertise-external-ip
--run-firewall=true                              --advertise-loadbalancer-ip
--run-service-proxy=true                         --advertise-pod-cidr=true
--bgp-graceful-restart=true                      --enable-ibgp=false
--bgp-holdtime=3s                                --enable-overlay=false
--kubeconfig=/var/lib/kube-router/kubeconfig     --enable-pod-egress=false
--cluster-asn=<BGP AS>                           --override-nexthop=true
```

How do I peer with the “local” TORs?

- Kube-Router expects the K8s nodes to be annotated with the peer (leaf) IP, AS and password so we simply need to annotate the K8s nodes accordingly for example:

```
kube-router.io/peer.ips=<leaf-1_IP>,<leaf-2_IP>
kube-router.io/peer.asns=<ACI AS>,<ACI AS>
kube-router.io/peer.passwords=<MD5 Pass>,<MD5 Pass>
```

- Annotating any node with the above config will result in such node to peer with “leaf-1” and ”leaf-2”

Agenda

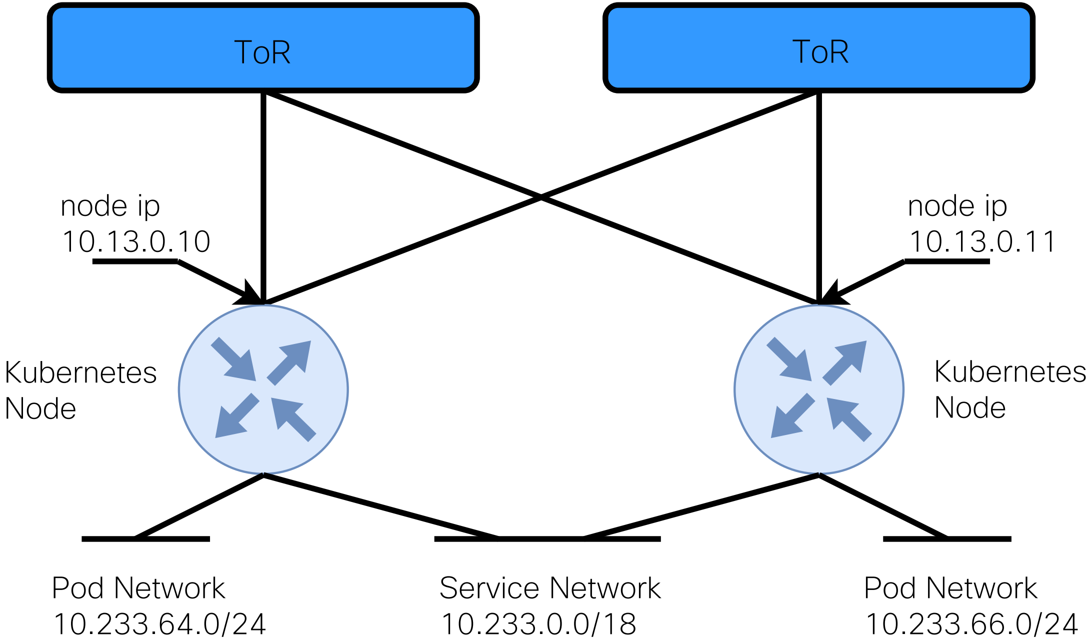
- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- Design Kubernetes Network on NX-OS fabric
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- **Design Kubernetes Network on NX-OS fabric**
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

BGP-Based CNI

Simplified



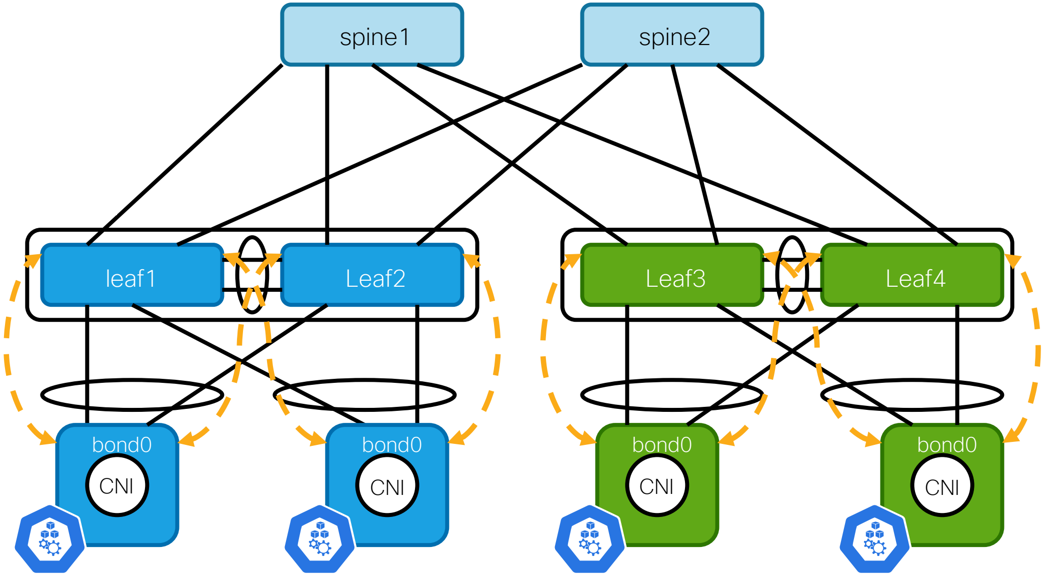
- Each Kubernetes node has one node IP
- one or more ranges of IP addresses (CIDRs) for pod networks
- a shared network for the whole Kubernetes cluster which is called the service network.

Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- **Design Kubernetes Network on NX-OS fabric**
  - **Design BGP network on IP Fabric**
  - Design BGP network on VXLAN EVPN Fabric
  - Visualization with NDFC

Network Architecture

Peer with Switch



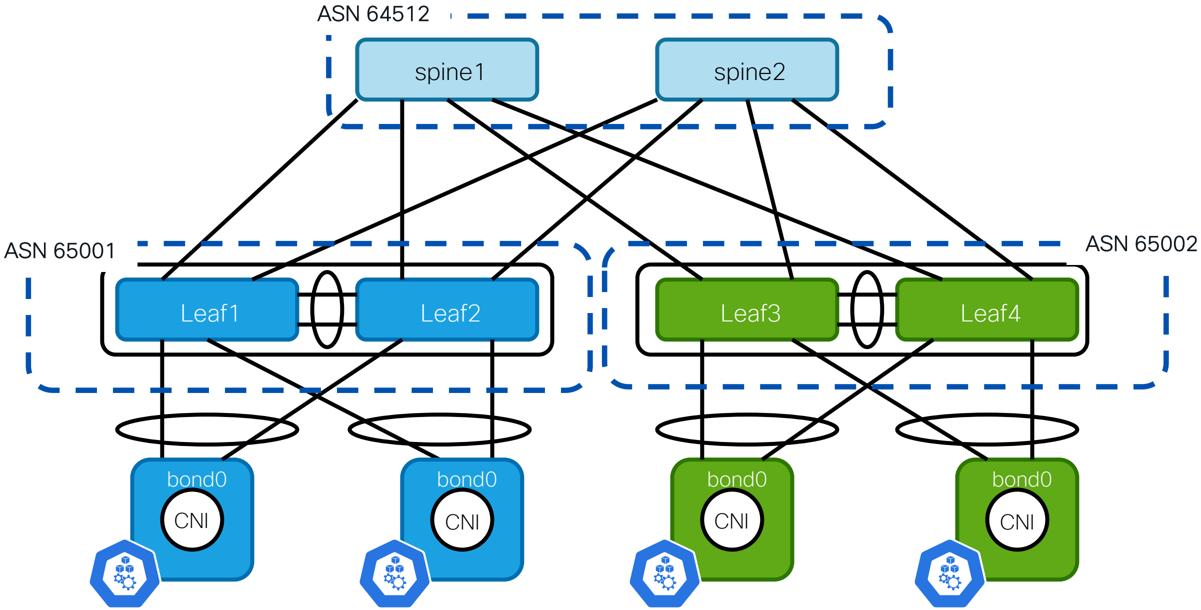
⇠ - - ⇢ iBGP

- Scalable approach, the leaf switches become Route-Reflector
- Data is transported with the original header

```
apiVersion: projectcalico.org/v3
kind: IPPool
metadata:
  name: default-pool
spec:
  blockSize: 24
  cidr: 10.233.64.0/20
  ipipMode: Never
  nodeSelector: all()
  vxlanMode: Never
```

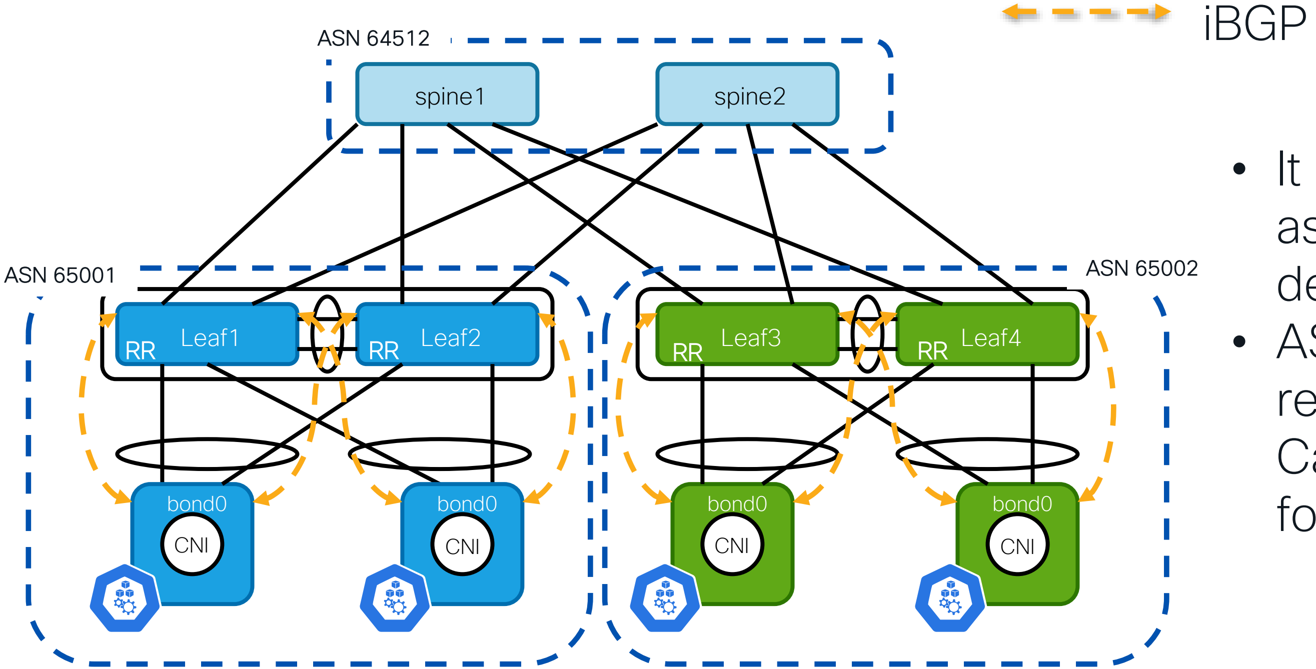
Network Architecture

Deploy Over IP Fabric



Network Architecture

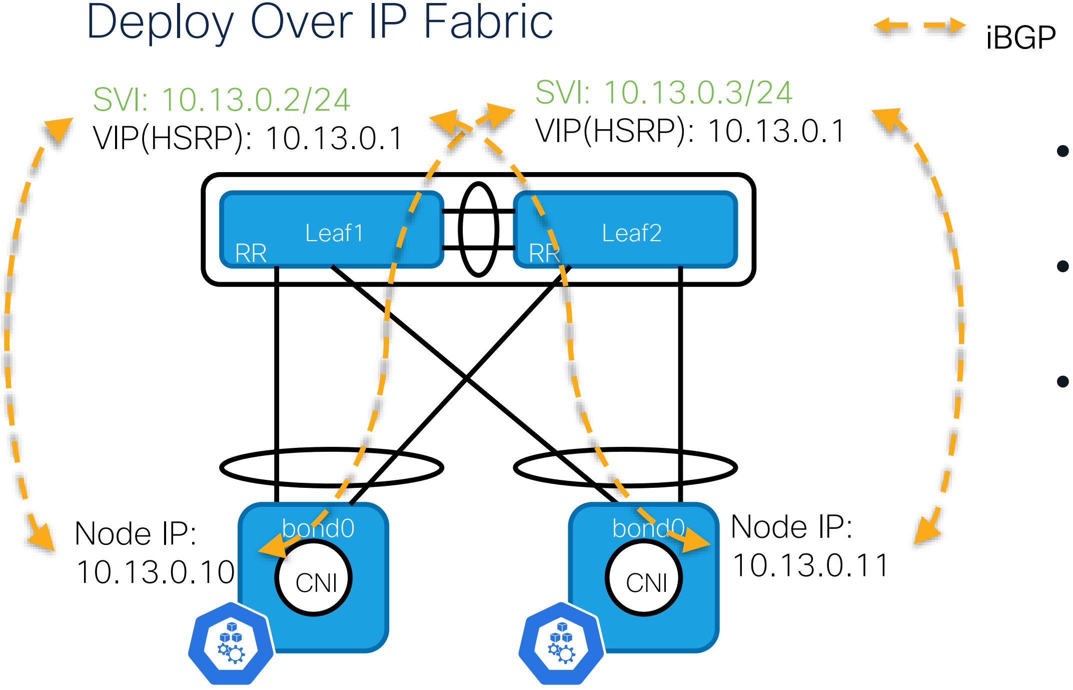
Deploy Over IP Fabric



- It is usually referred to as AS-Per-Rack design.
- AS-Per-Rack is recommended by Calico, but exclusively for IP Fabric(RFC 7938)

Network Architecture

Deploy Over IP Fabric



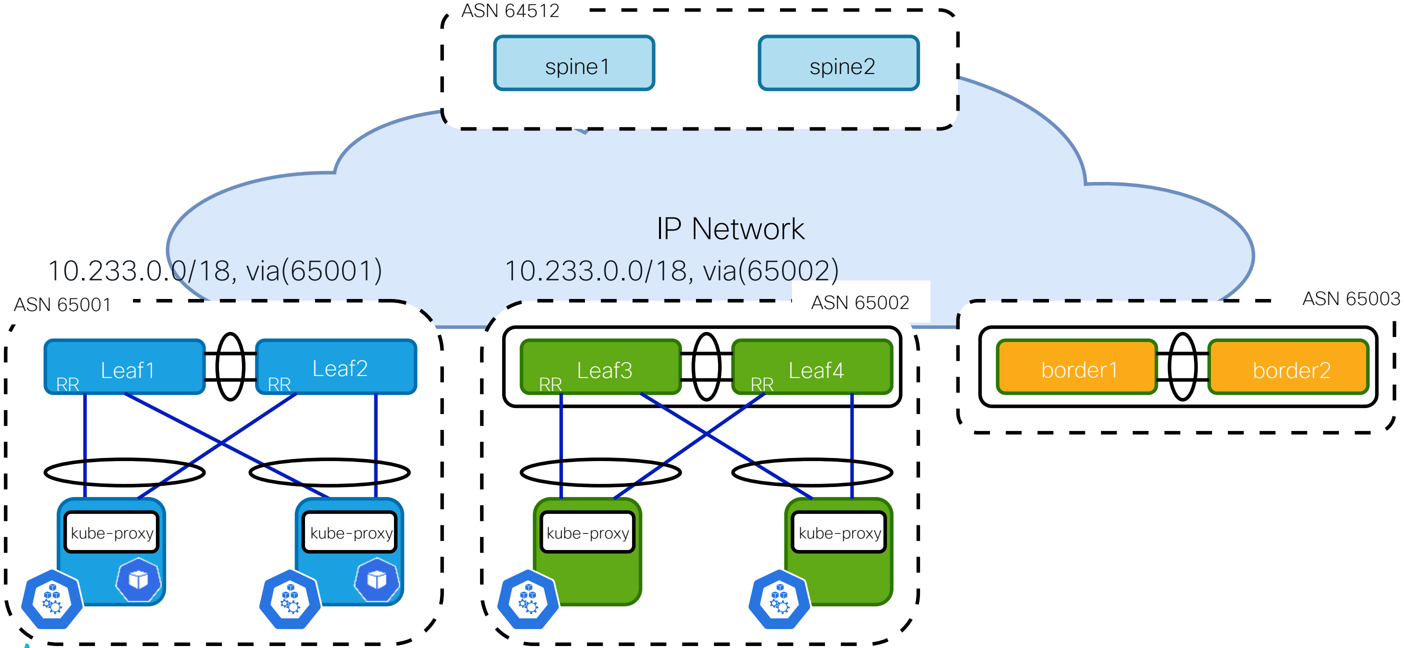
- HSRP/VRRP is used for gateway redundancy
- Kubernetes nodes peer with the primary IP address of SVI
- The node subnets are advertised into BGP to provide nodes reachability

Deploy over IP Fabric

Service Traffic

Service Subnet:
10.233.0.0/18

```
10.233.0.0/18, ubest/mbest: 4/0
    *via 10.4.0.37, [20/0], 2d10h, bgp-64512, external, tag 65001
    *via 10.4.0.45, [20/0], 2d10h, bgp-64512, external, tag 65001
```



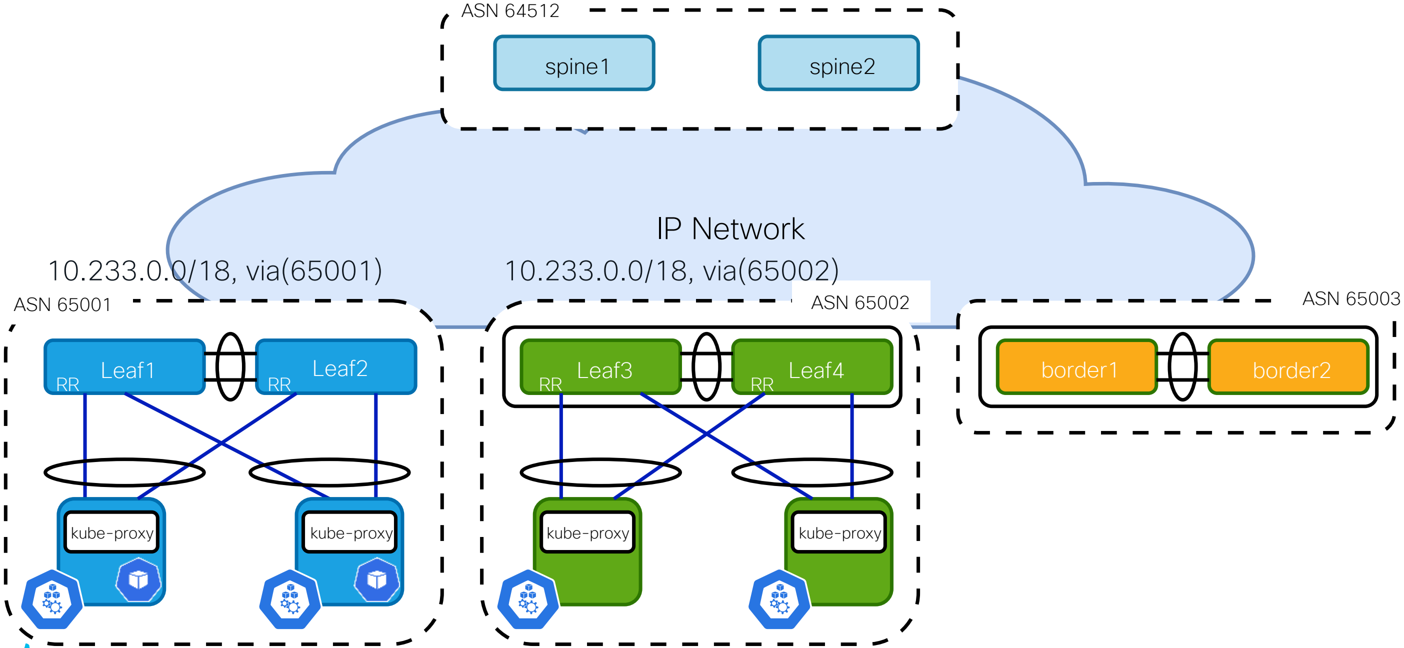
Deploy over IP Fabric

Service Traffic

Service Subnet:
10.233.0.0/18

```
router bgp 64512
  bestpath as-path multipath-relax
```

```
10.233.0.0/18, ubest/mbest: 4/0
    *via 10.4.0.21, [20/0], 2d10h, bgp-64512, external, tag 65002
    *via 10.4.0.29, [20/0], 2d10h, bgp-64512, external, tag 65002
    *via 10.4.0.37, [20/0], 2d10h, bgp-64512, external, tag 65001
    *via 10.4.0.45, [20/0], 2d10h, bgp-64512, external, tag 65001
```



Deploy over IP Fabric

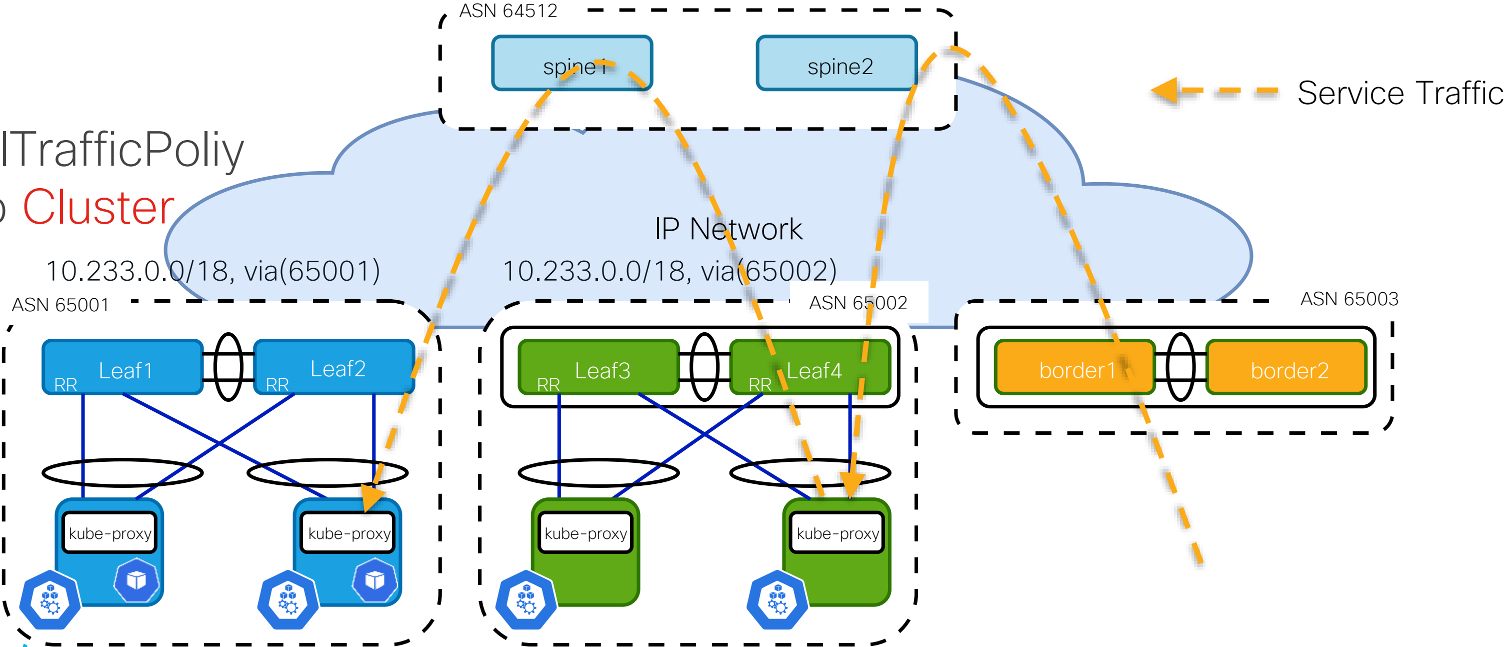
Sub-optimal service traffic

```
router bgp 64512
  bestpath as-path multipath-relax
```

Service Subnet:
10.233.0.0/18

```
10.233.0.0/18, ubest/mbest: 4/0
    *via 10.4.0.21, [20/0], 2d10h, bgp-64512, external, tag 65002
    *via 10.4.0.29, [20/0], 2d10h, bgp-64512, external, tag 65002
    *via 10.4.0.37, [20/0], 2d10h, bgp-64512, external, tag 65001
    *via 10.4.0.45, [20/0], 2d10h, bgp-64512, external, tag 65001
```

K8s externalTrafficPoliy is set to Cluster



Deploy over IP Fabric

Avoid Second Hop of Service Traffic

```
router bgp 64512
  bestpath as-path multipath-relax
```

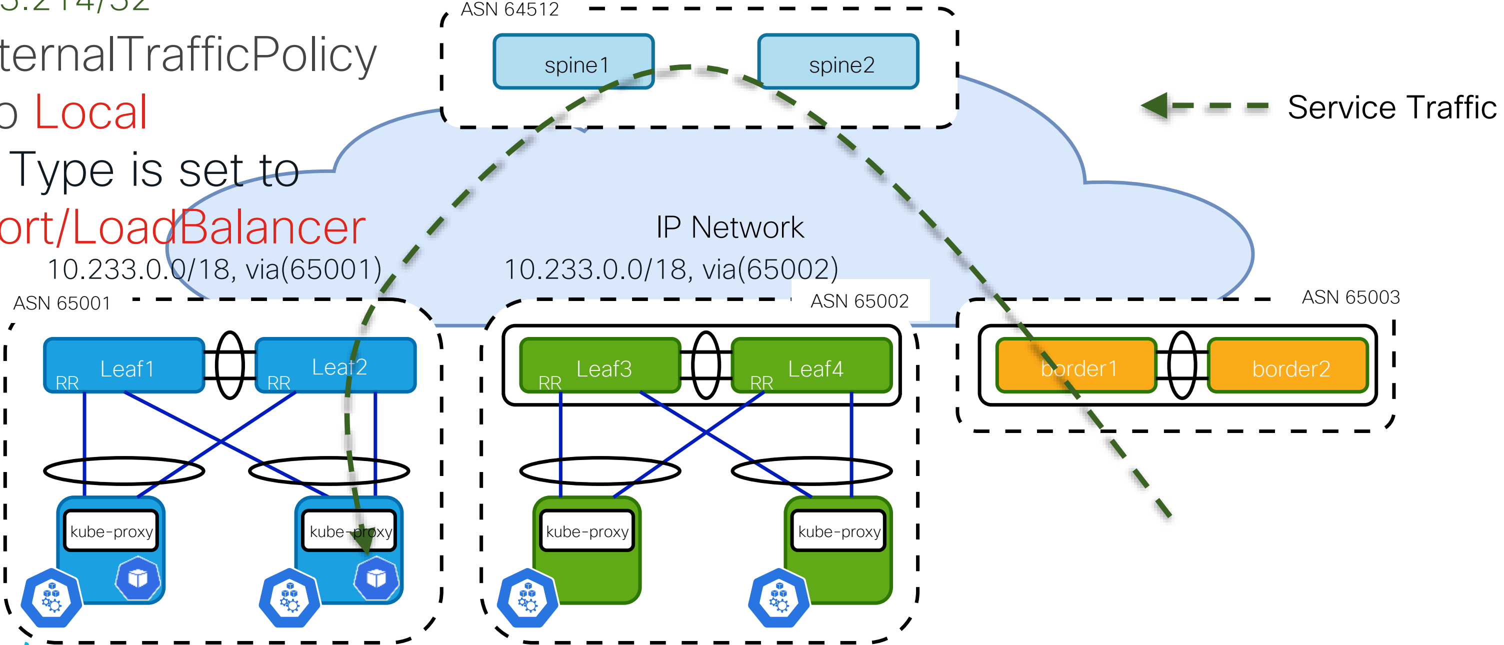
```
10.233.63.214/32, ubest/mbest: 2/0
    *via 10.4.0.37, [20/0], 2d10h, bgp-64512, external, tag 65001
    *via 10.4.0.45, [20/0], 2d10h, bgp-64512, external, tag 65001
```

Service Subnet:
10.233.0.0/18

Service ip:
10.233.63.214/32

K8s externalTrafficPolicy is set to Local

Sevice Type is set to NodePort/LoadBalancer



Exposing Services

A note on “externalTrafficPolicy”

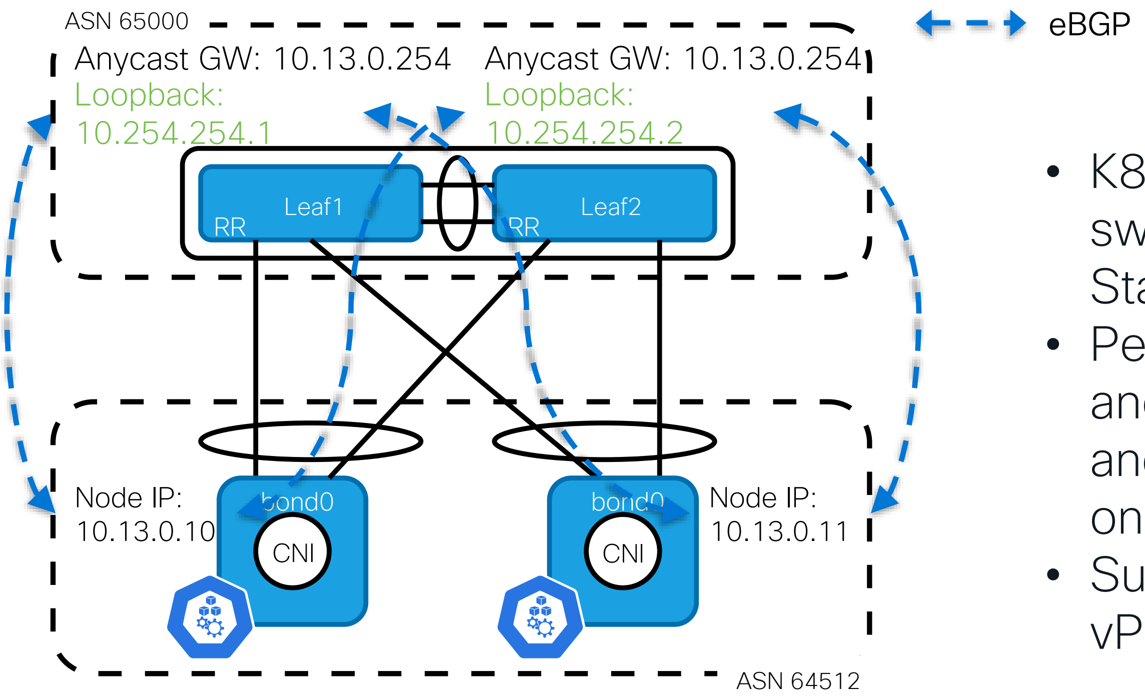
- Denotes if this Service desires to route to node-local or cluster-wide endpoints.
- externalTrafficPolicy == Cluster
  - Pros: overall good load-bala us
  - Cons: potential second h ng additional latency
- externalTrafficPolic
  - Pros: avoid the urce IP is preserved
  - Cons: poten ed workload spreading
    - Pods can be sp nly with topologySpreadConstraints

This applies to IP Fabric, VXLAN EVPN Fabric and ACI

Agenda

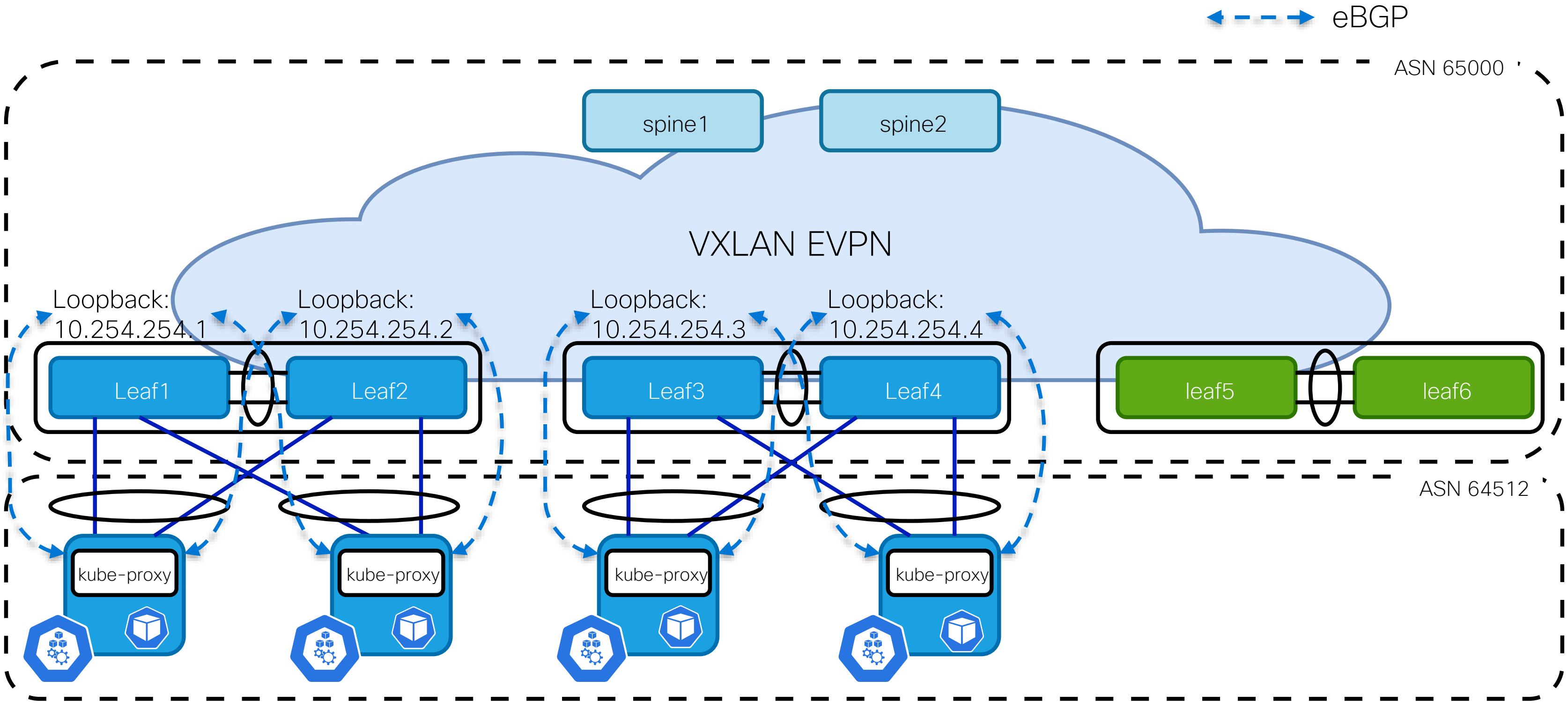
- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- **Design Kubernetes Network on NX-OS fabric**
  - Design BGP network on IP Fabric
  - **Design BGP network on VXLAN EVPN Fabric**
  - Visualization with NDFC

Connecting K8s nodes to Leaf Switches



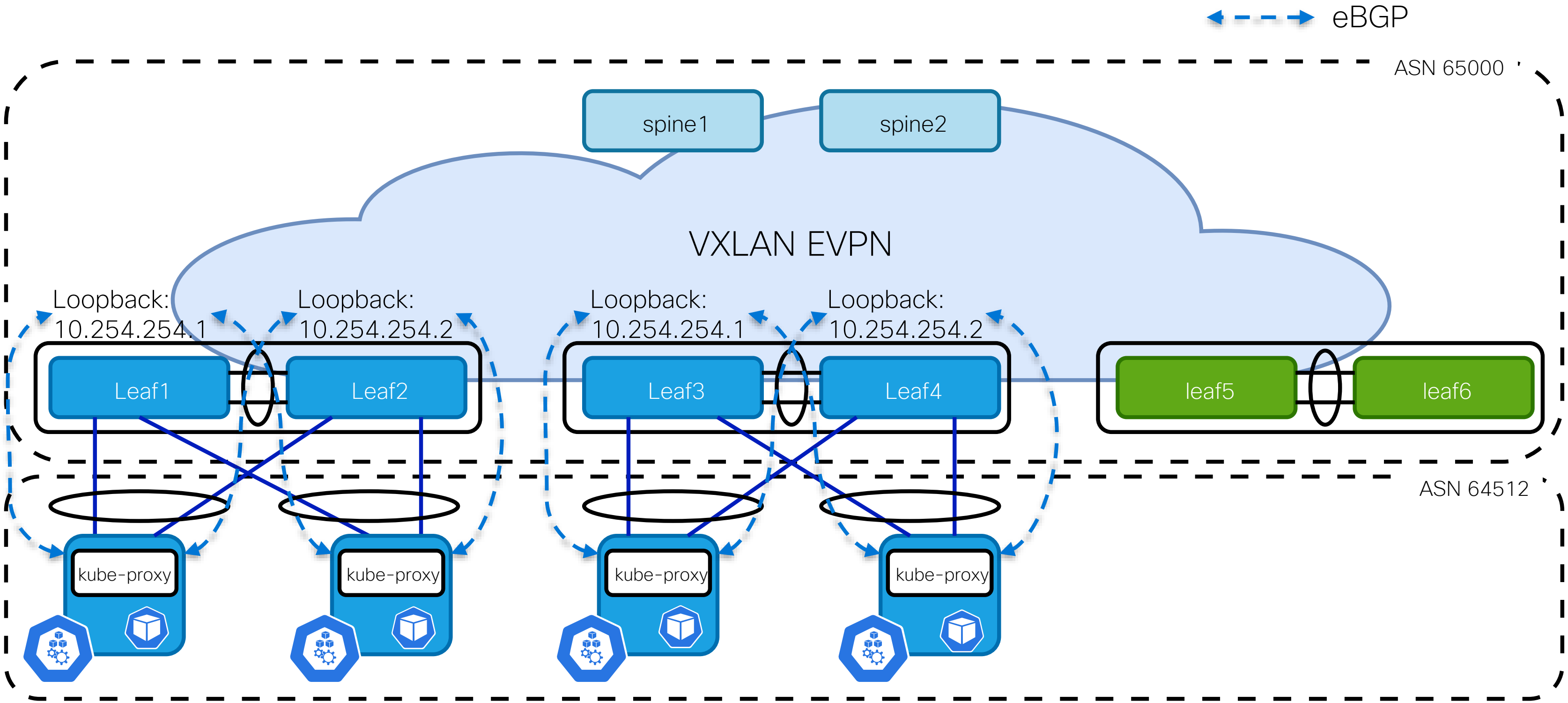
- K8s nodes connect to Leaf switches using VPC or Active-Standby
- Peering eBGP between K8s nodes and leaf switches using node IP and localized loopback addresses on each leaf switches
- Suggest peering iBGP between vPC pair in the user VRF

As-Per-Cluster design



As-Per-Cluster design

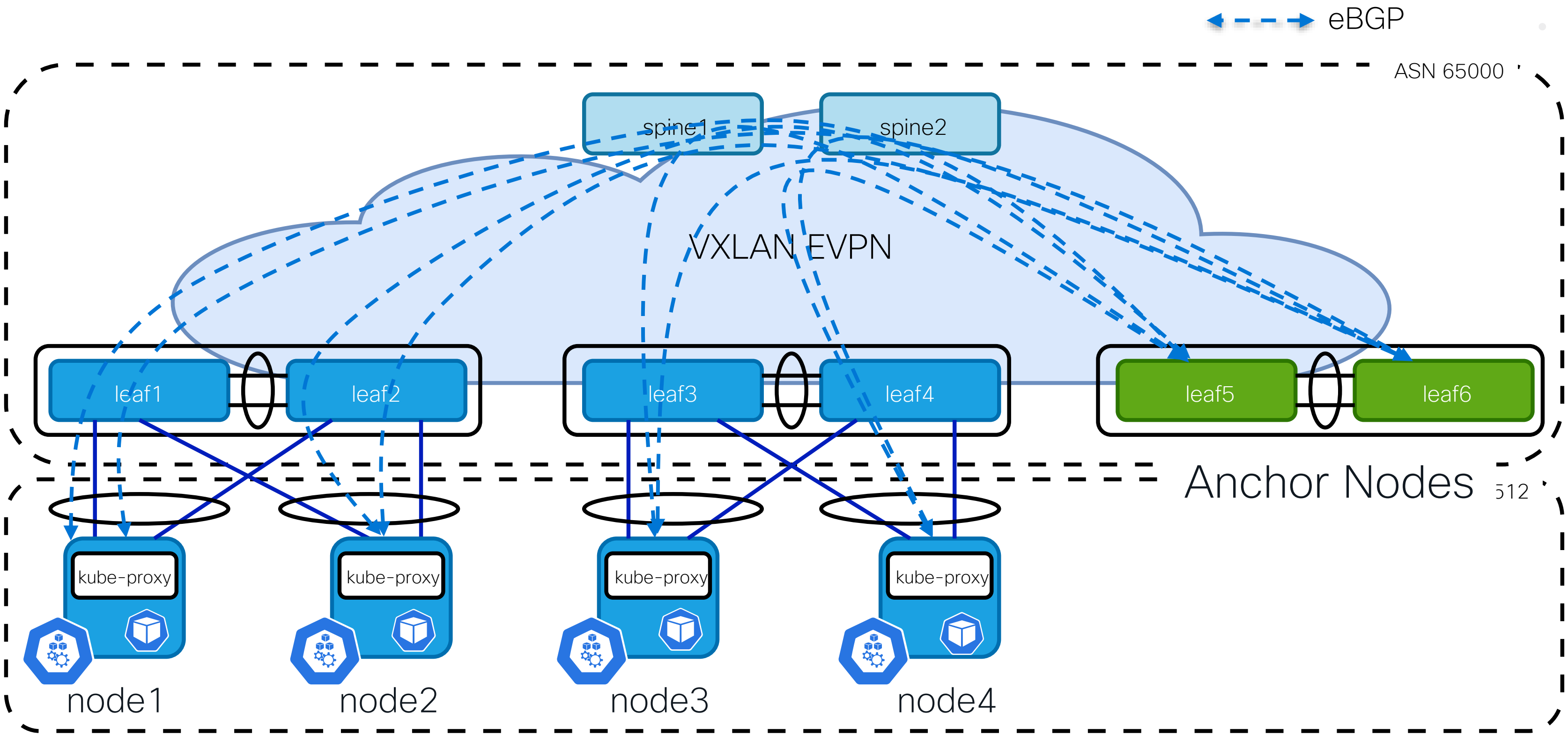
Use same loopback addresses



As-Per-Cluster design

- Using single AS number per cluster reduces the complexity of bootstrap K8s node
- Loopback addresses are local to leaf switches
  - It does not need to be advertised to EVPN address family
    - But you will need iBGP peering between vPC peer switches
  - The same loopbacks can be used on all pairs of leaf switches
- Minimum BGP configuration can be tuned on Calico
  - `disable-peer-as-check` and `as-override` are needed on leaf switches

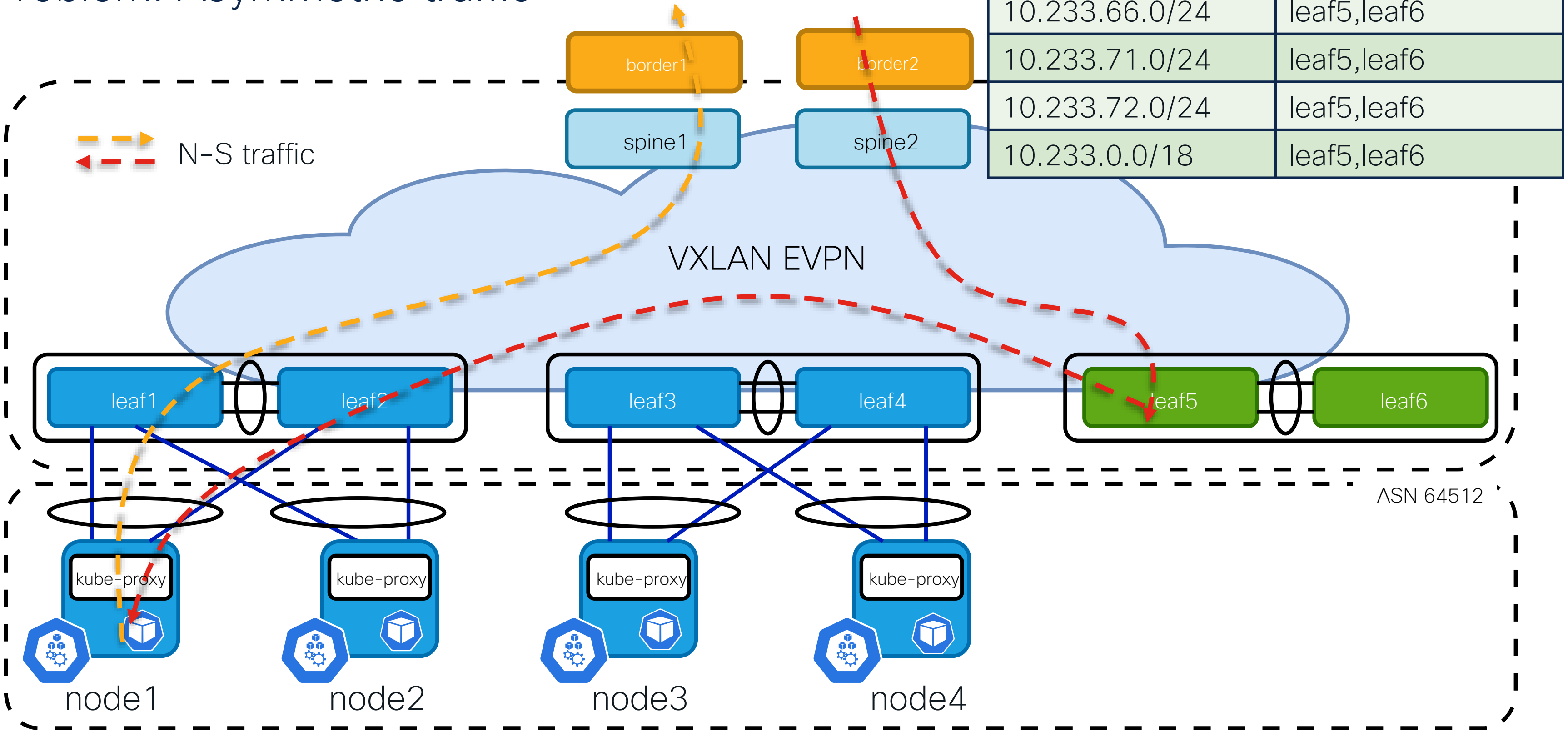
Centralized Routing Peering



Centralized Routing Peering

Problem: Asymmetric traffic

prefix	nexthop
10.233.64.0/24	leaf5,leaf6
10.233.66.0/24	leaf5,leaf6
10.233.71.0/24	leaf5,leaf6
10.233.72.0/24	leaf5,leaf6
10.233.0.0/18	leaf5,leaf6



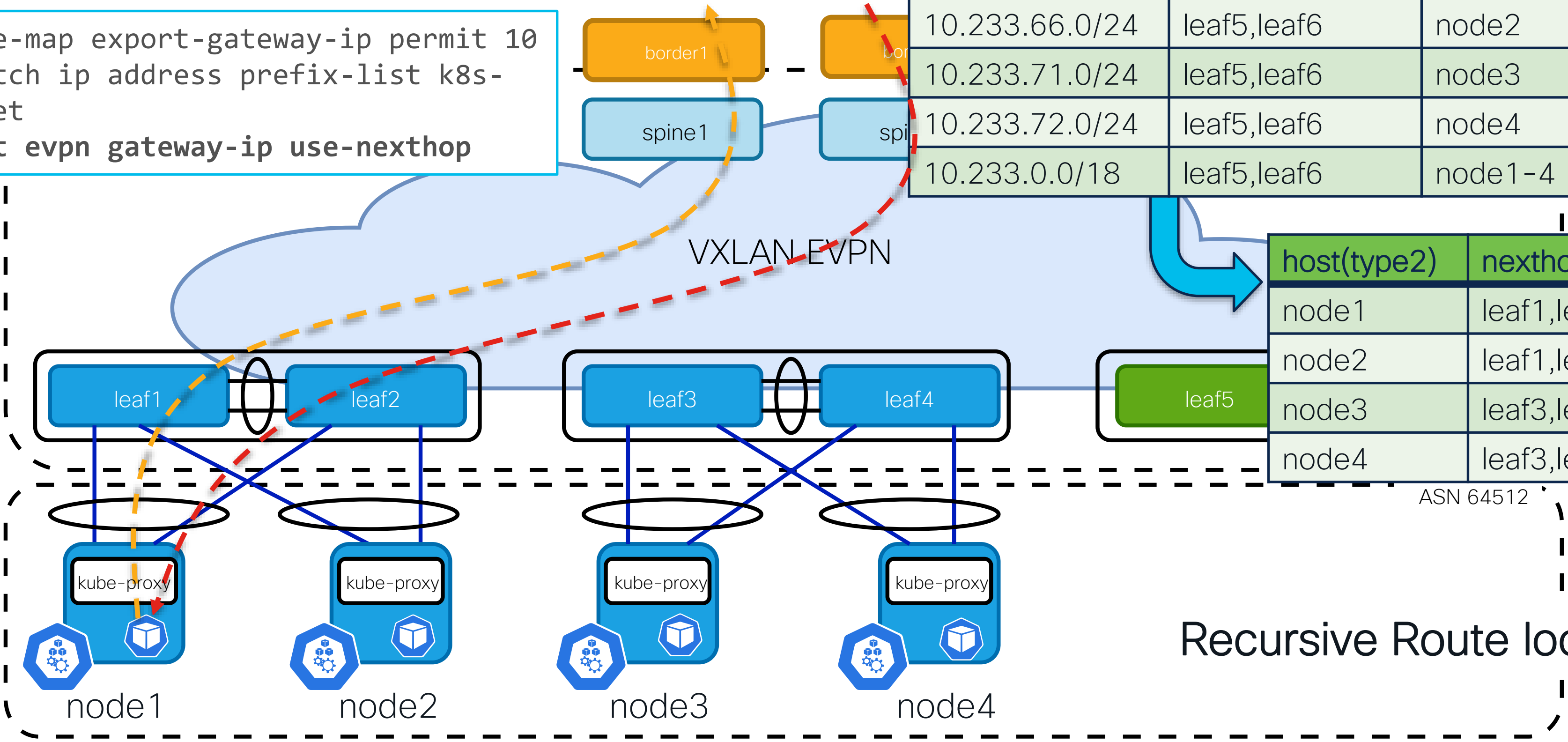
Centralized Routing Peering

Solution

```
route-map export-gateway-ip permit 10
  match ip address prefix-list k8s-
subnet
  set evpn gateway-ip use-nexthop
```

prefix	nexthop	gateway
10.233.64.0/24	leaf5,leaf6	node1
10.233.66.0/24	leaf5,leaf6	node2
10.233.71.0/24	leaf5,leaf6	node3
10.233.72.0/24	leaf5,leaf6	node4
10.233.0.0/18	leaf5,leaf6	node1-4

host(type2)	nexthop
node1	leaf1,leaf2
node2	leaf1,leaf2
node3	leaf3,leaf4
node4	leaf3,leaf4

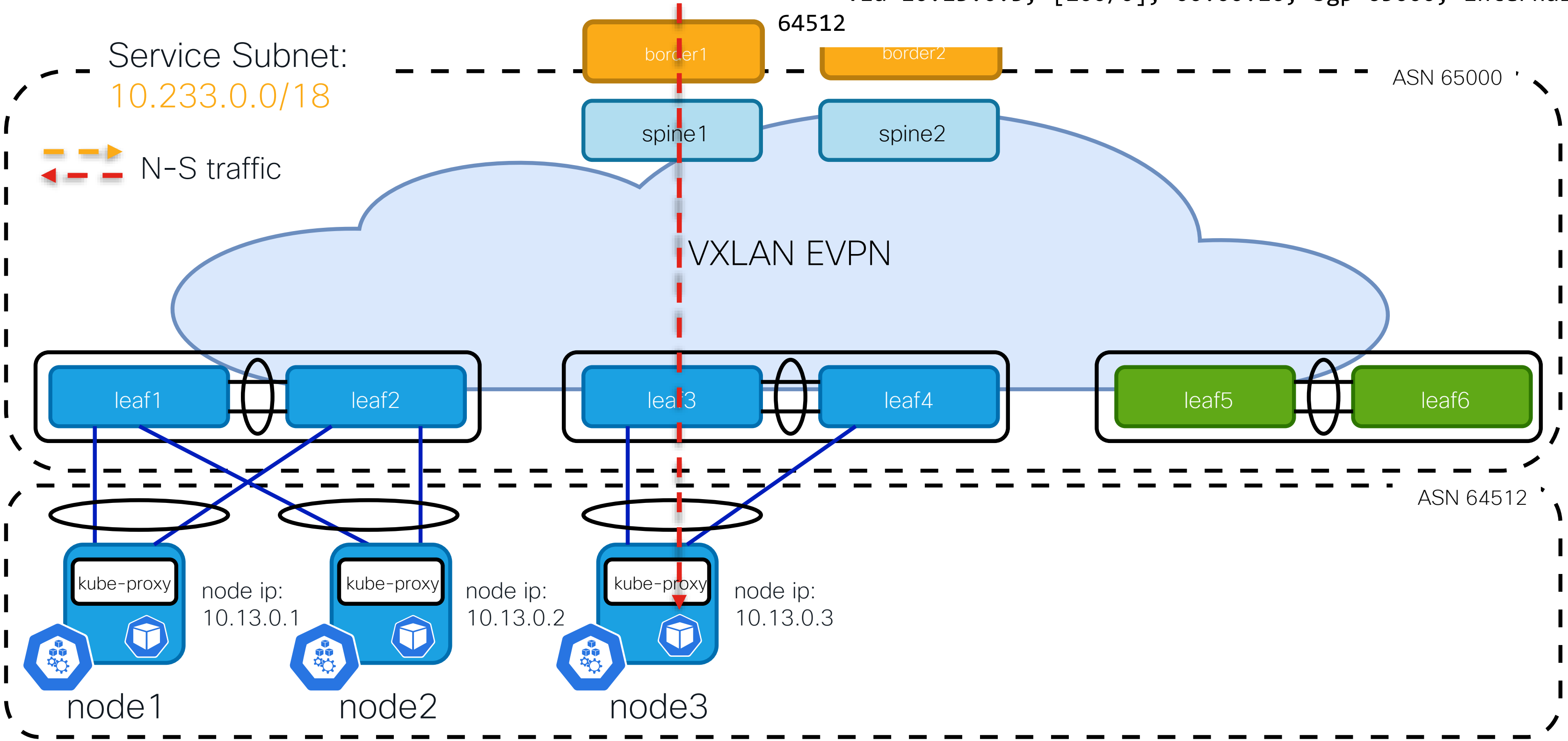


Recursive Route lookup

Centralized Routing Peering

Service Traffic

```
10.233.0.0/18, ubest/mbest: 4/0
    *via 10.13.0.3, [200/0], 00:00:16, bgp-65000, internal, tag
64512
```



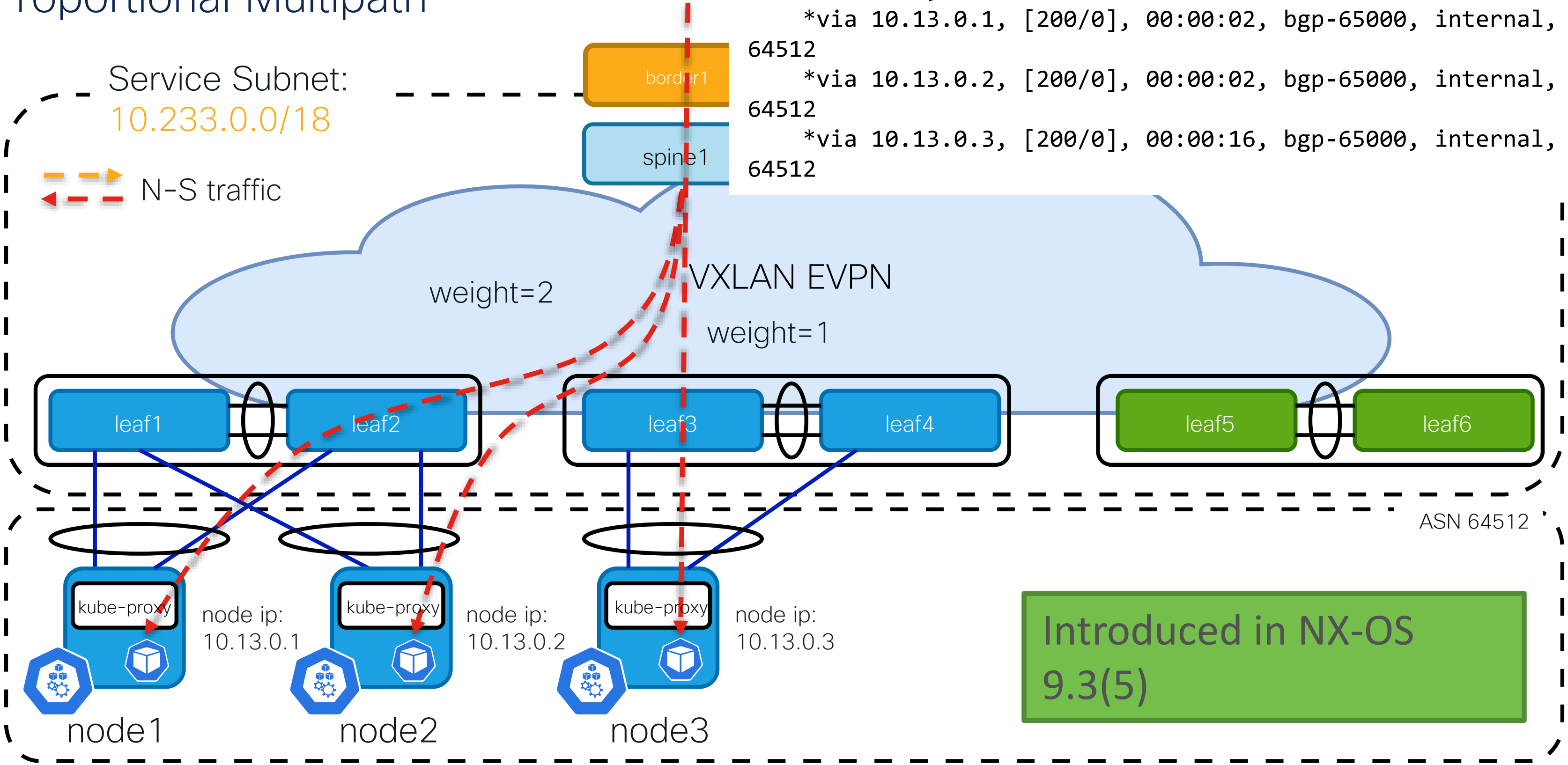
Service Subnet:
10.233.0.0/18

N-S traffic

Centralized Routing Peering

Proportional Multipath

```
10.233.0.0/18, ubest/mbest: 4/0
    *via 10.13.0.1, [200/0], 00:00:02, bgp-65000, internal, tag
64512
    *via 10.13.0.2, [200/0], 00:00:02, bgp-65000, internal, tag
64512
    *via 10.13.0.3, [200/0], 00:00:16, bgp-65000, internal, tag
64512
```

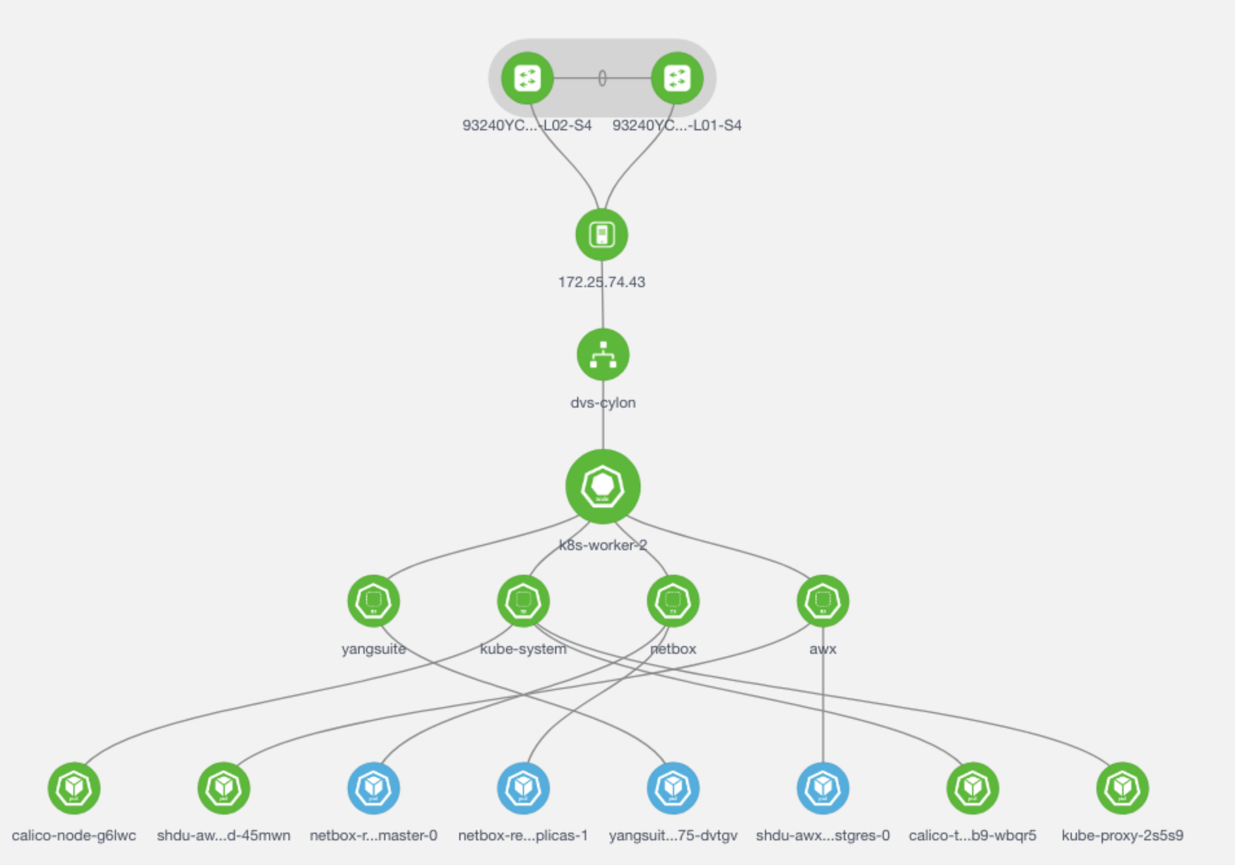


Introduced in NX-OS 9.3(5)

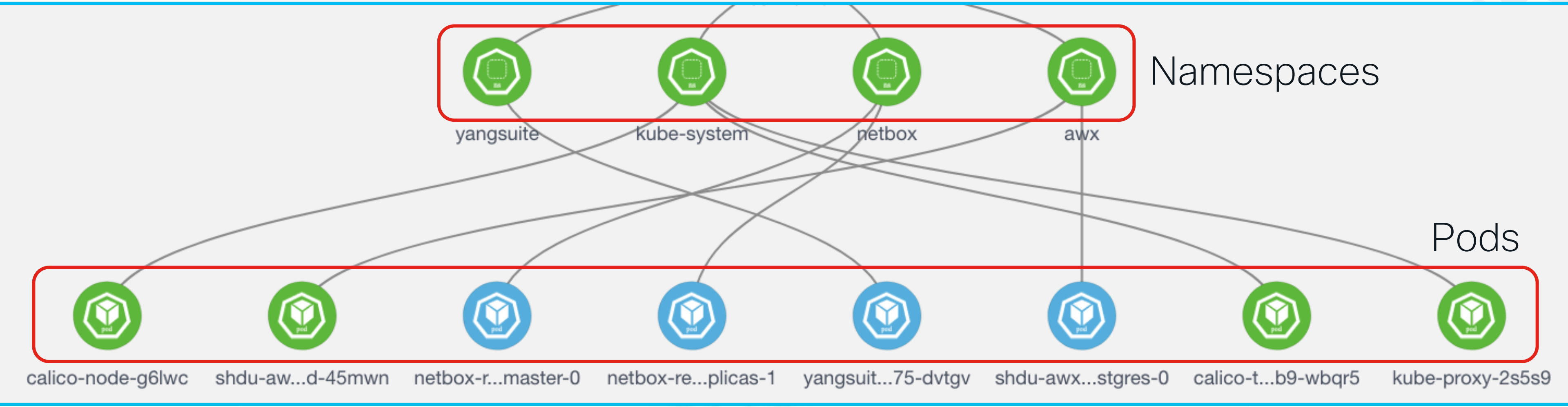
Agenda

- Kubernetes Refresh
- Kubernetes Network Challenges
- Design Kubernetes Network on ACI fabric
  - ACI CNI
  - BGP-based CNI
  - Design BGP network on ACI
- **Design Kubernetes Network on NX-OS fabric**
  - Design BGP network on IP Fabric
  - Design BGP network on VXLAN EVPN Fabric
  - **Visualization with NDFC**

Kubernetes Visualization with NDFC

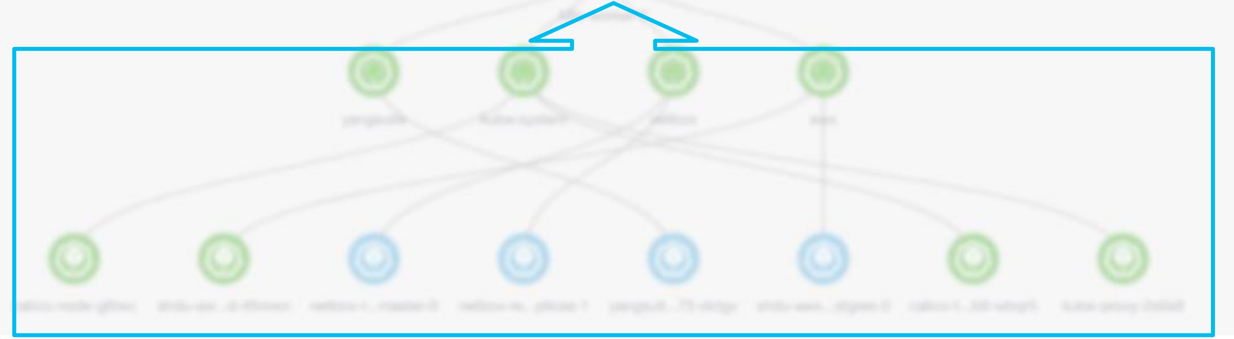


Kubernetes Visualization with NDFC

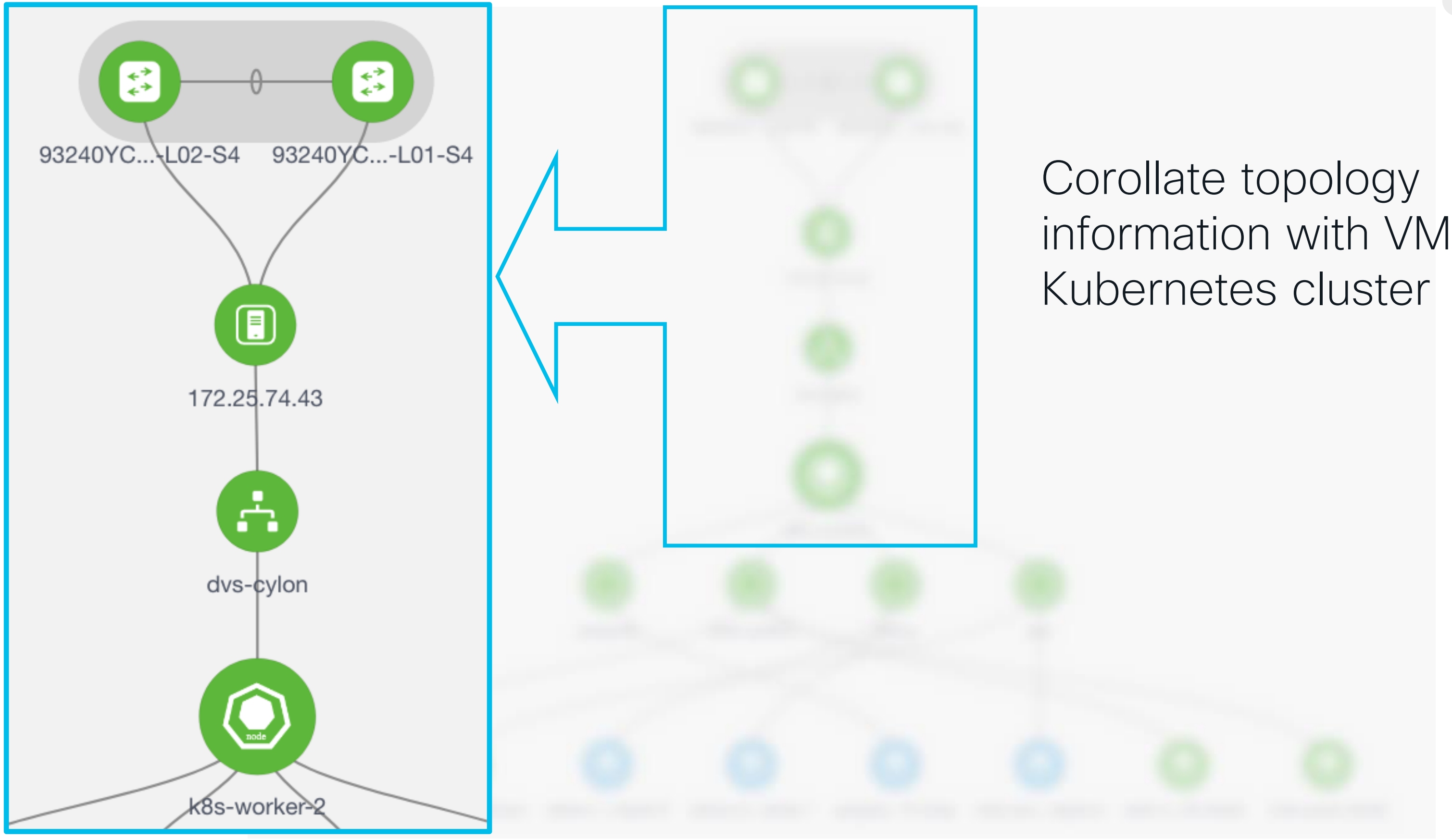


Namespaces

Pods



Kubernetes Visualization with NDFC



Corollate topology information with VM and Kubernetes cluster

Summary

- Choose ACI-CNI or BGP-based CNI based your business requirements
- Greenfield Calico network does not require L2 extension
- The best practice is peering BGP neighborship with local switches
- Centralized Route Peering can simplify the configuration of Calico
  - But does require additional consideration to optimize traffic
- All the necessary features are shipped today on ACI and NX-OS

Reference

- Cisco Application Centric Infrastructure K8s Design White Paper
  - https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-743182.html
- Cisco ACI CNI and Kubernetes Integration
  - https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/kb/b_Kubernetes_Integration_with_ACI.html
- Cisco NX-OS Calico Network Design White Paper
  - https://www.cisco.com/c/en/us/td/docs/dcn/whitepapers/cisco-nx-os-calico-network-design.html
- Configuring Proportional Multipath for VNF
  - https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/93x/vxlan/configuration/guide/b-cisco-nexus-9000-series-nx-os-vxlan-configuration-guide-93x/b-cisco-nexus-9000-series-nx-os-vxlan-configuration-guide-93x_appendix_011010.html

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.

The bridge to possible

Thank you

cisco Live!
ALL IN