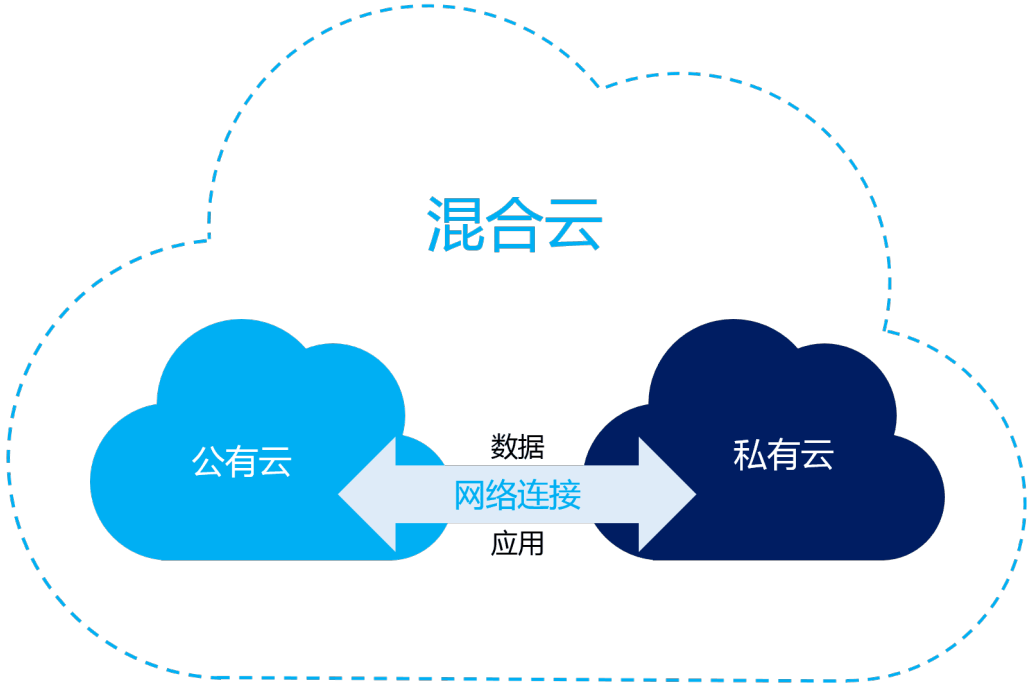

KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Hybridnet: Let Underlay & Overlay Network Coexist in Your K8s Cluster!

Bruce Ma & Liang Fang, AntGroup

Background: Hybrid Cloud Infrastructure

- **Single IaaS Vendor**
  - Container network is deeply coupled to IaaS abilities
  - Extreme virtualization performance
- **Challenges on Hybrid Cloud Infrastructure**
  - Consistent functionality among heterogeneous infrastructures
  - Agile & stable delivery capabilities
  - Unified perspective for ops & maintenance
  - High performance



What We Need in Hybrid Cloud Scenarios

- Unified network models to reduce cognitive & maintenance costs
- Hiding the complexity of heterogeneous infra from users
- High performance networking
- Minimizing dependence on low-level networking technologies
- Deep integration with K8s, providing dual-stack, IP retain and other advanced IPAM capabilities

Target

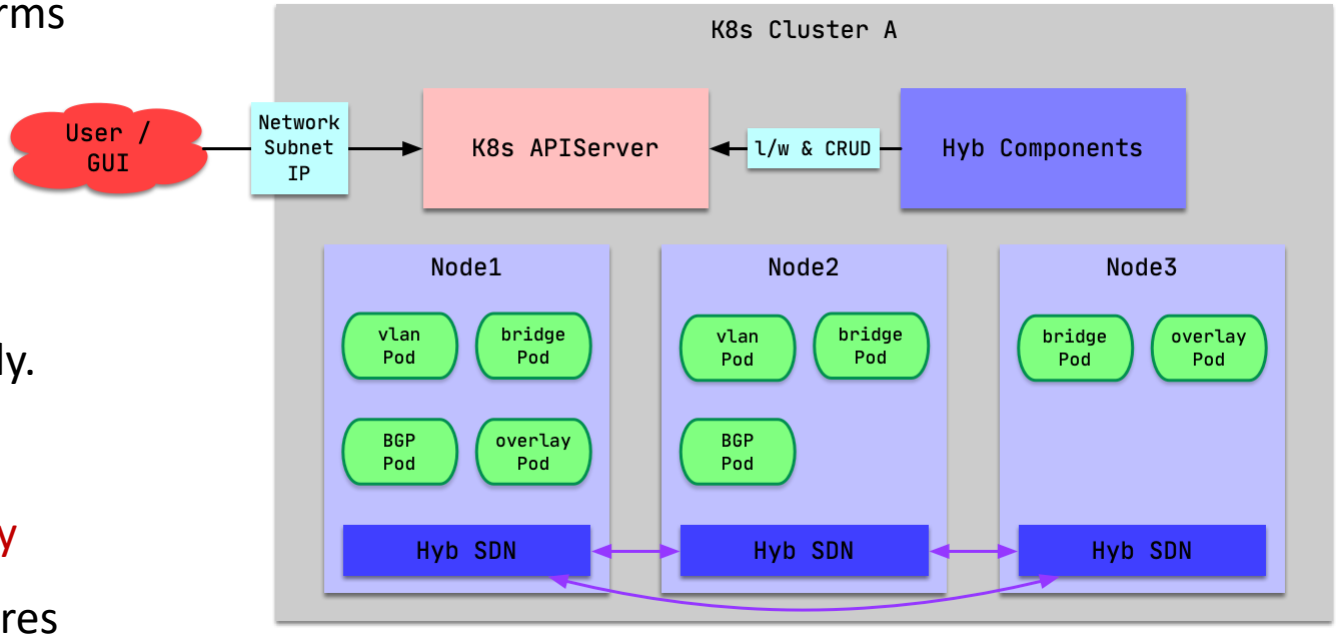
Implement a container network solution with mature underlying network technology, and ensure that both overlay and underlay network have the same flexible and scalable IP address management capability under the premise of co-existence.

Introduction to Hybridnet

Hybridnet is an open source container networking solution designed for hybrid clouds, integrated with Kubernetes and used officially by following well-known PaaS platforms

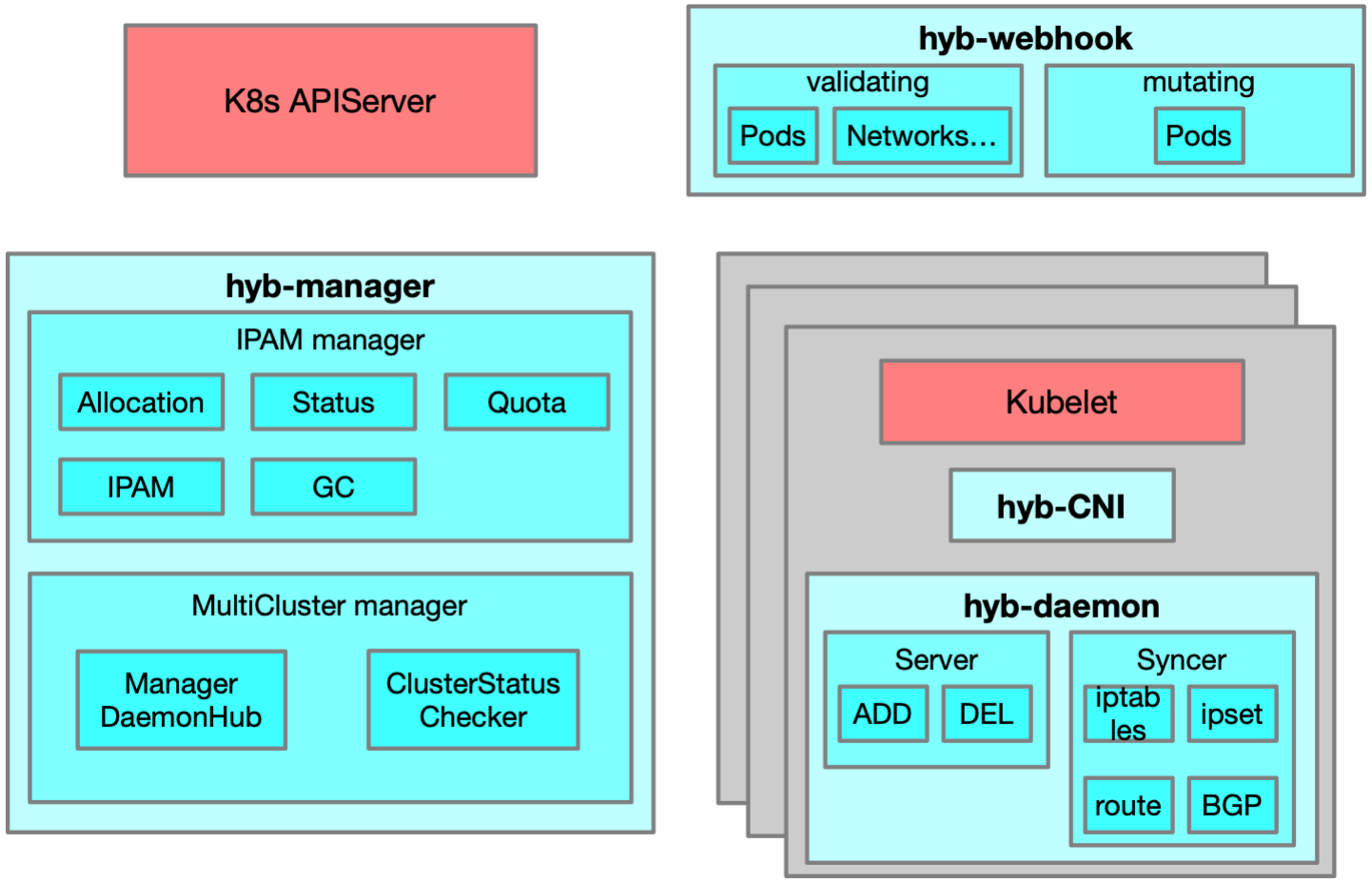
- ACK Distro of Alibaba Cloud
- AECP of Alibaba Cloud
- SOFAStack of Ant Financial Co.

Hybridnet is actively used in near 100 sites currently.



Hybridnet allows users to create overlay & underlay networks in the cluster at the same time, and ensures direct connectivity between all underlay & overlay pods while maintaining high-performance communication.

Components



- Manager
  - IPAM
  - Multi-Cluster connectivity
- Webhook
  - validating
  - mutating
- Daemon
  - Server, called by CNI
  - Syncer, local resources syncing
- CNI
  - CNI binary

Design Principles

“hybrid-lay” Principles

- Support underlay & overlay type networks simultaneously at node level
- Complete connectivity between underlay & overlay type pods, and support K8s Service & NetworkPolicy natively
- Underlay & overlay share the same network models, equipped with the same advanced IPAM abilities (E.g., specifying subnet or IP address, IP retain.)

Underlay & Overlay Constraints

- High performance, reachability from outside and scheduling strategies based on node network topology is assured for pods in underlay network
- Overlay network is unique as a whole, and IPs could shift around arbitrarily within a cluster
- Overlay pods shall communicate with underlay pods without boundaries, but IP ranges of the two types networks cannot overlap

Core Models

Network

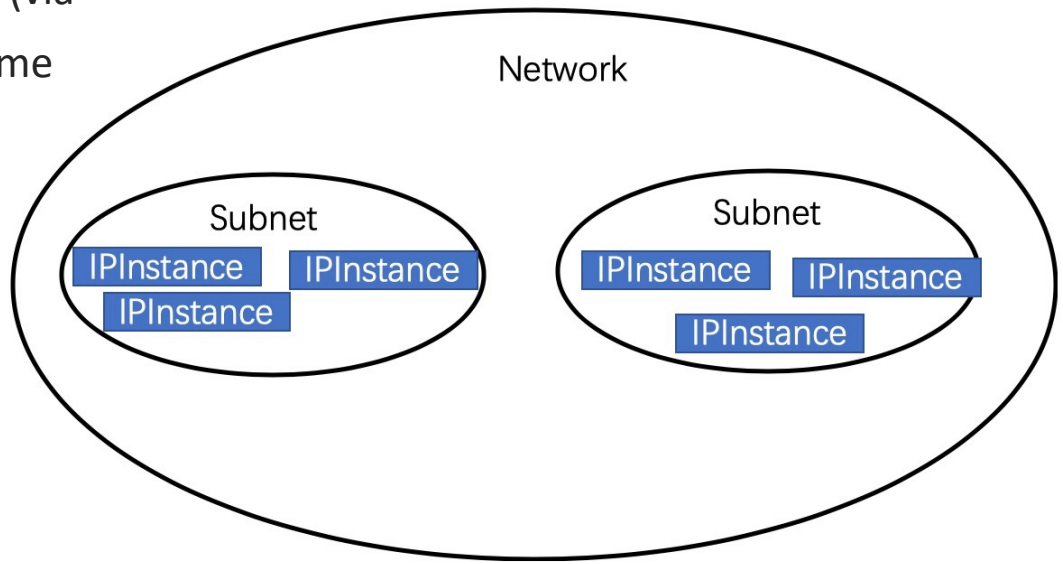
Every network is a scheduling domain of pods, which means that a pod using a specific network will be scheduled to nodes attached to that network (via node labels) . The scheduling domain represents a set of nodes with some same network properties (E.g., VLAN tag, gateway).

Subnet

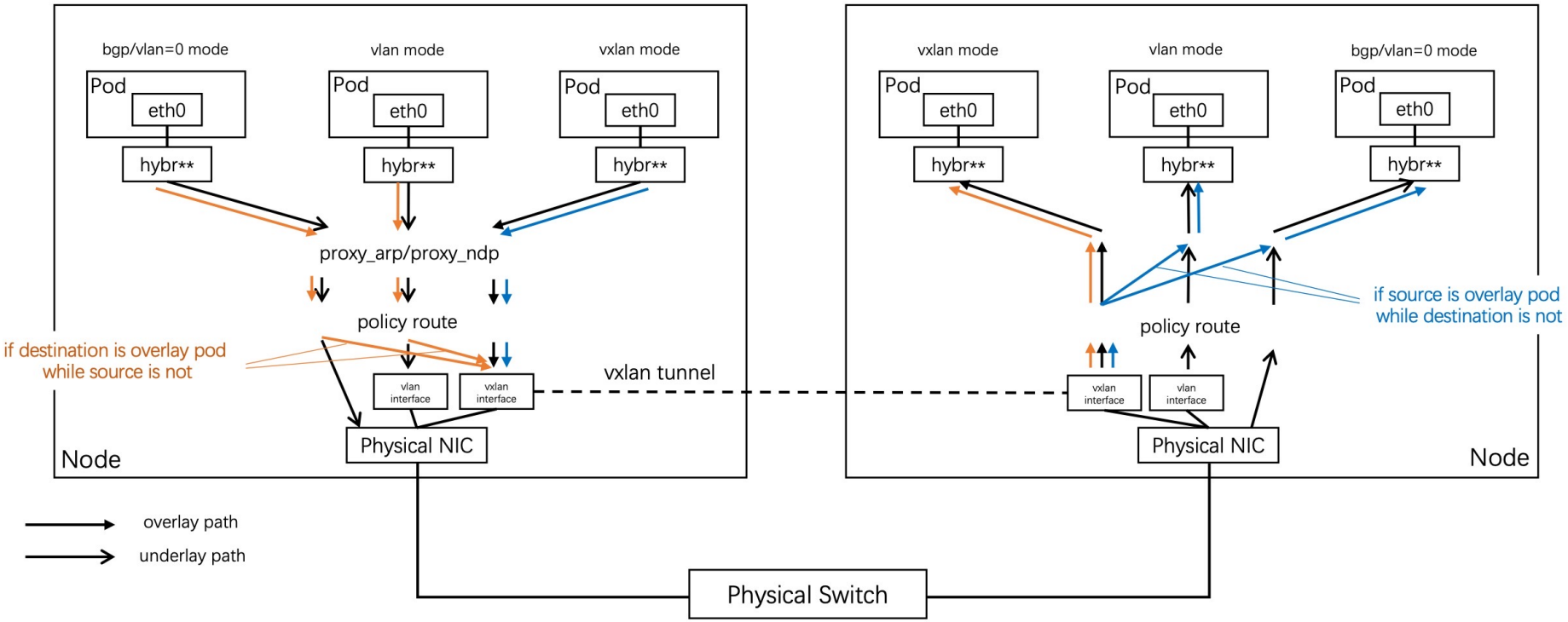
Every subnet belongs to a specific network, ant it contains an IP address range of container network.

IPInstance

Every ipinstance is corresponding to an IP address, and it belongs to a pod. IPInstance is a namespace scoped resource object, while Network & Subnet are cluster scoped resources.



How to implement “hybrid-lay”?

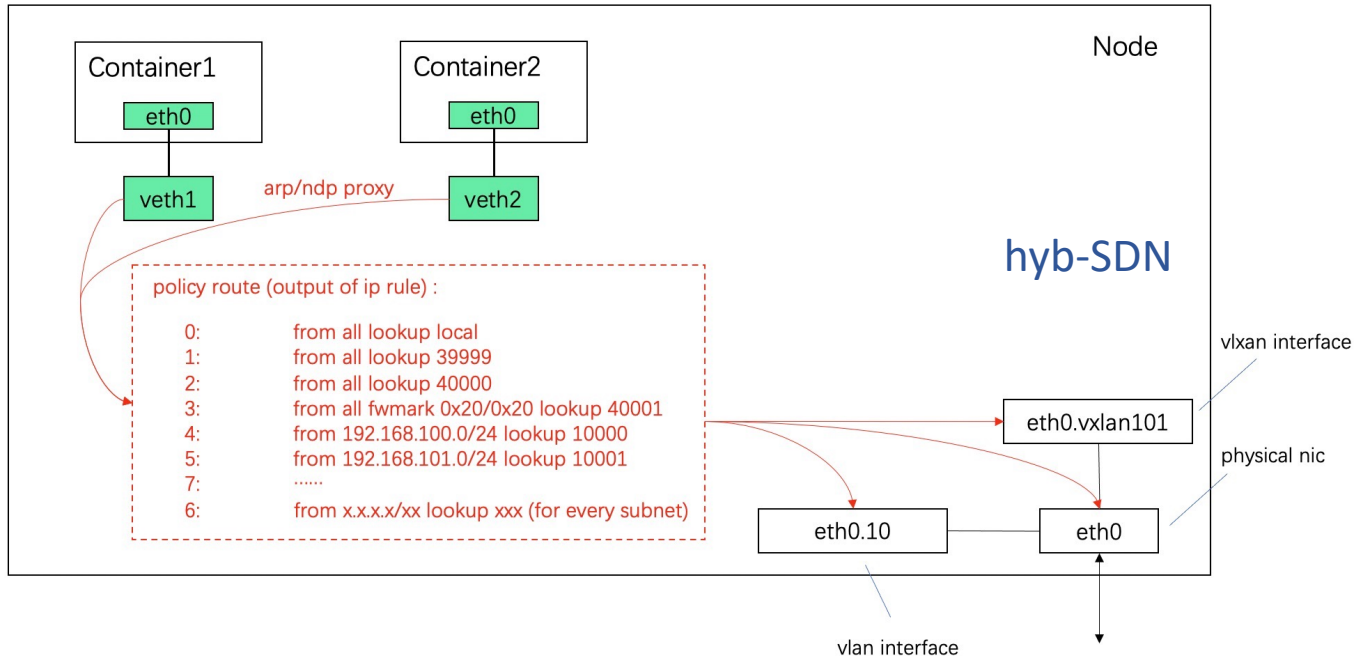


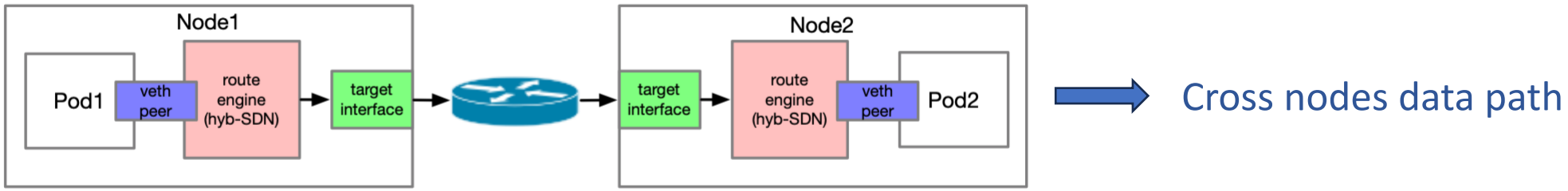
Hybridnet leverages **policy route** as the core implementation of data plane. Policy route is supported since Linux kernel 2.2, and get widely adopted, which proved its maturity and stability.

How to implement “hybrid-lay”?

Abstracts **hyb-SDN** as the node local networking component, which consists of policy route, ipset, iptables & netlink etc.

- All traffic generated from containers will go through the route table of the host
- Distinguish network type by IP range
- Choose network device by network type
- User-mode ARP/NDP proxy acceleration



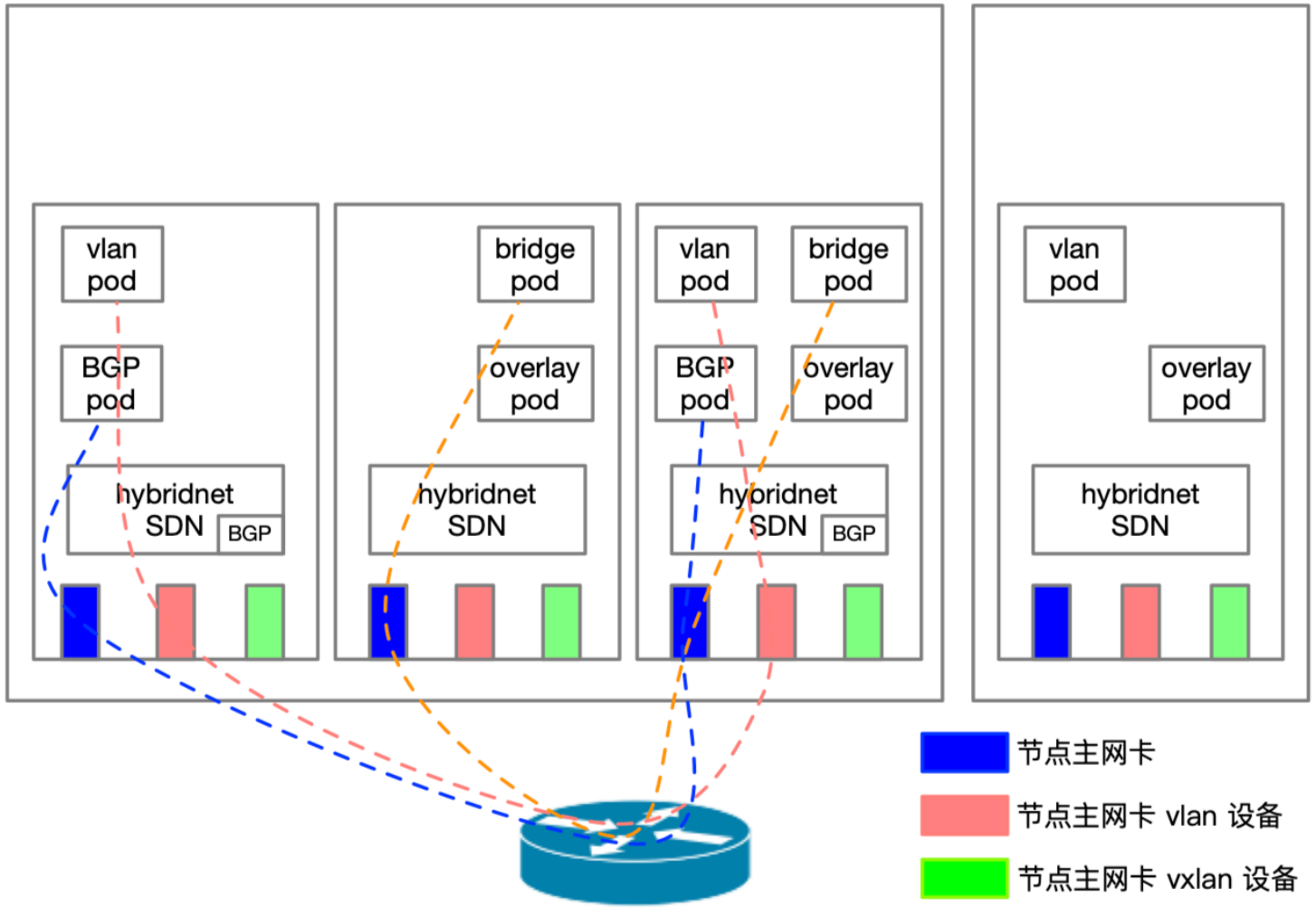


Cross nodes data path

Hybrid-lay Data Path Overview

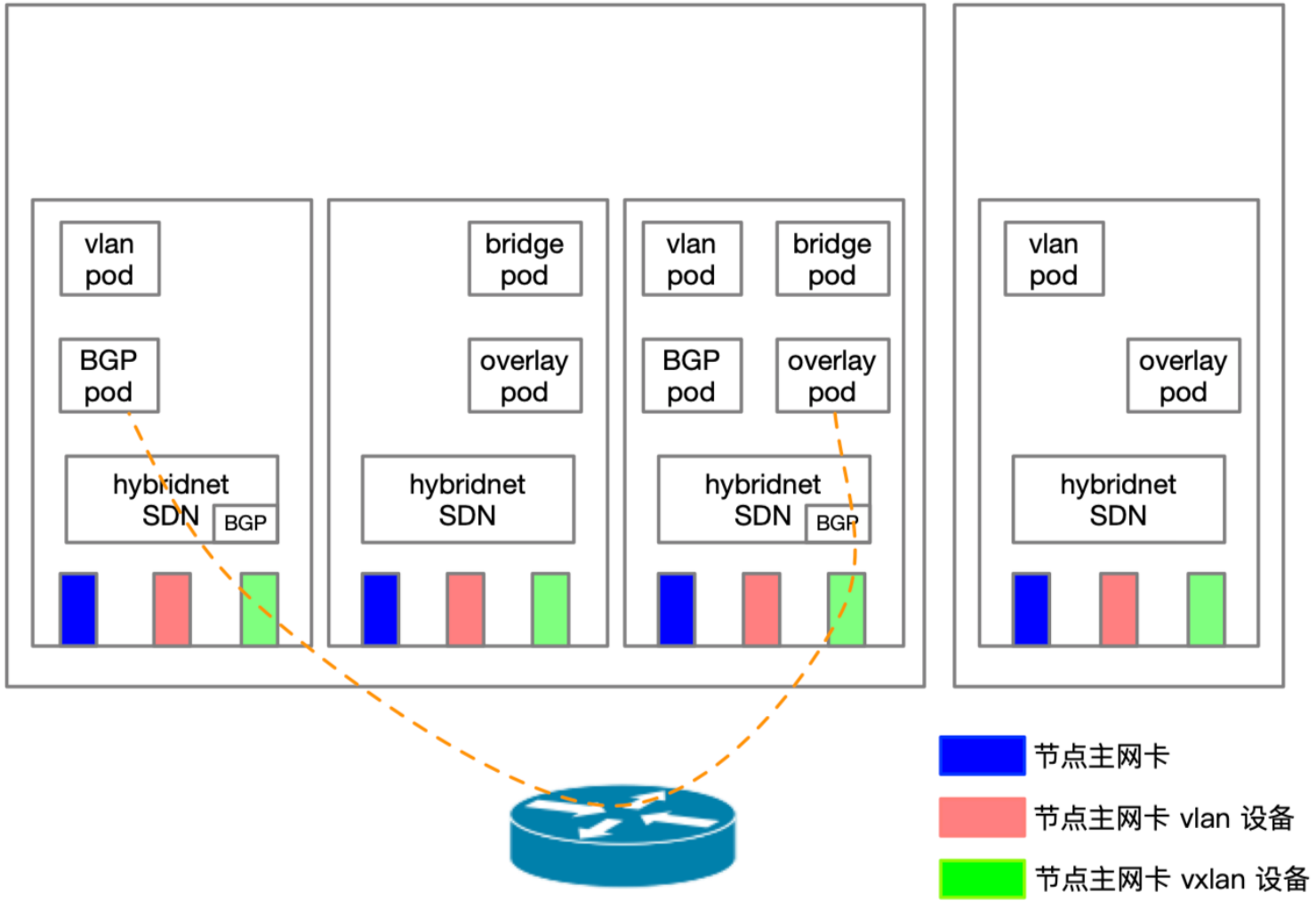
Access \ Accessed By	overlay pod	underlay pod	node	external
overlay pod	bi-directional tunnel	bi-directional tunnel	bi-directional tunnel	Underlay routes + SNAT
underlay pod	bi-directional tunnel	Underlay routes	Underlay routes	Underlay routes
node	bi-directional tunnel	Underlay routes	Underlay routes	Underlay routes
external	-	Underlay routes	Underlay routes	-

Hybrid-lay Data Path: Underlay



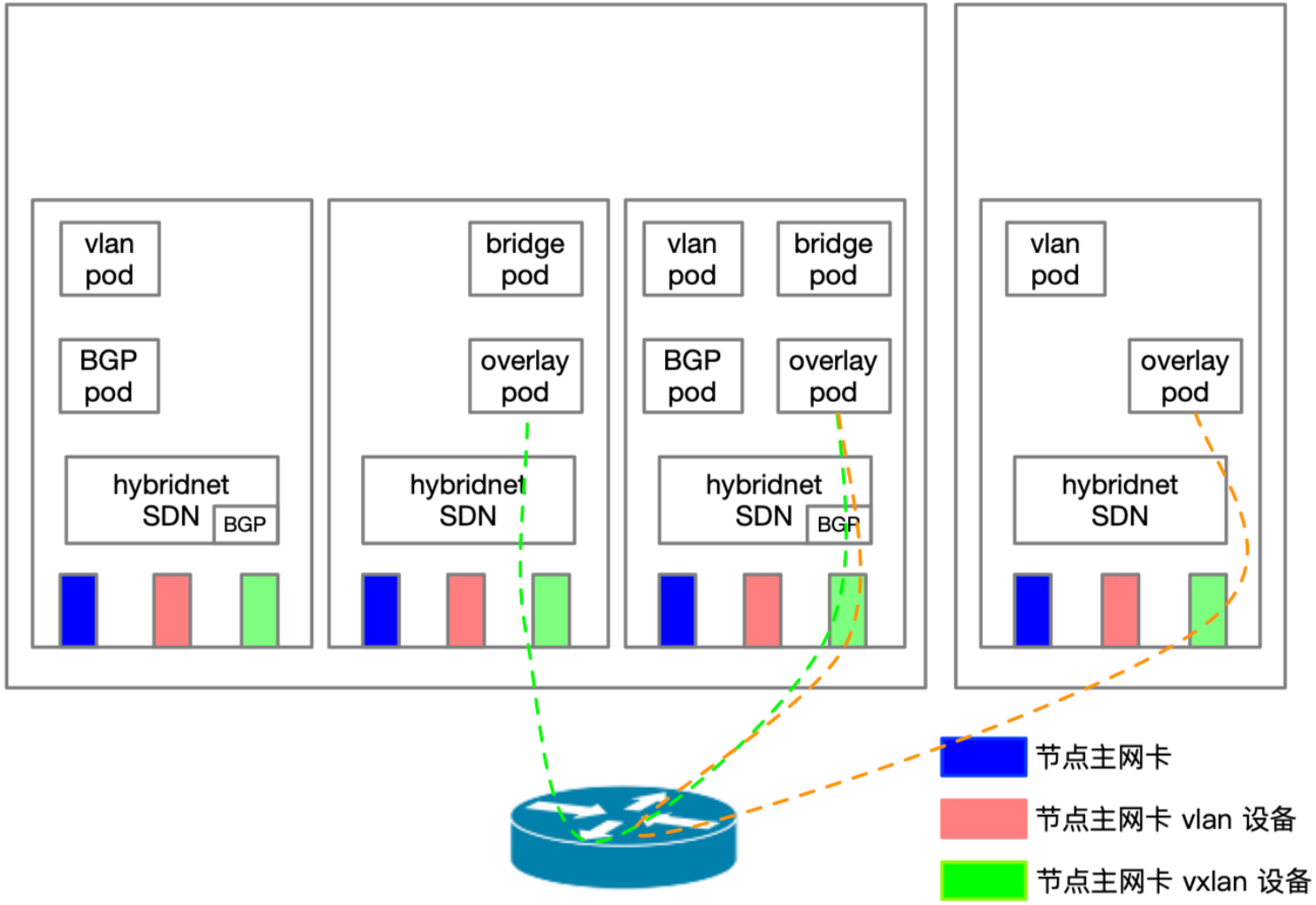
- Multiple underlay modes
  - VLAN
  - Bridge
  - BGP
- Underlying
  - BUM
  - L3 routing
  - VTEP

Hybrid-lay Data Path: Underlay + Overlay



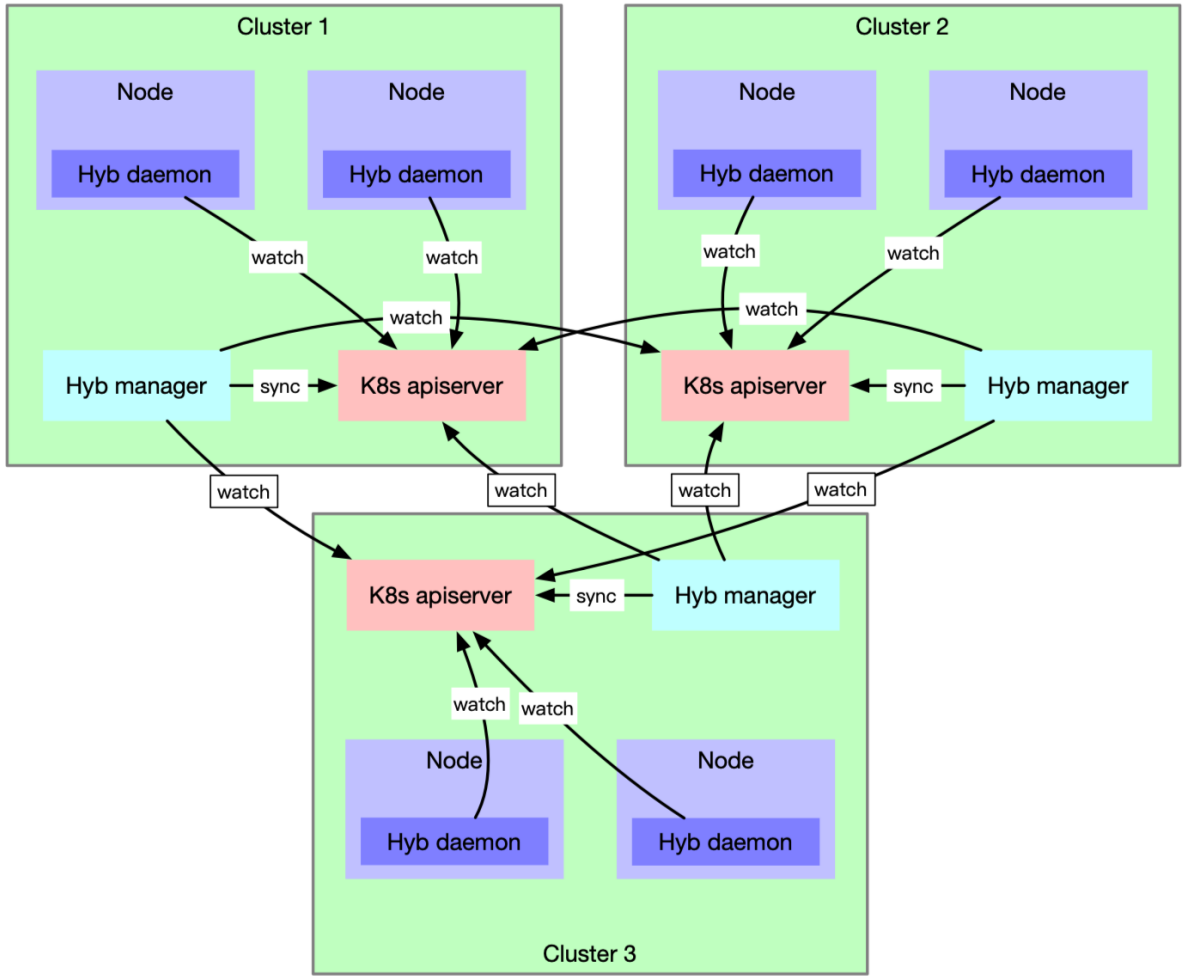
- Bi-directional tunnel

Hybrid-lay Data Path: Overlay



- Support multi-cluster connectivity
- Single Cluster overlay
  - VxLAN
  - Two Layer Addressing
    - user mode arp/ndp proxy
    - BUM

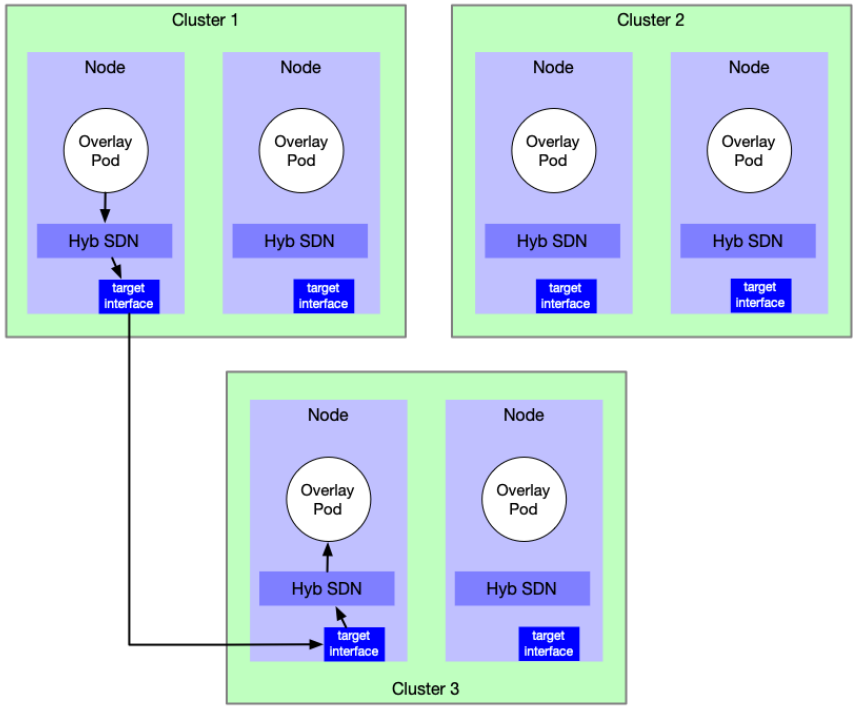
Multi-Cluster Connectivity



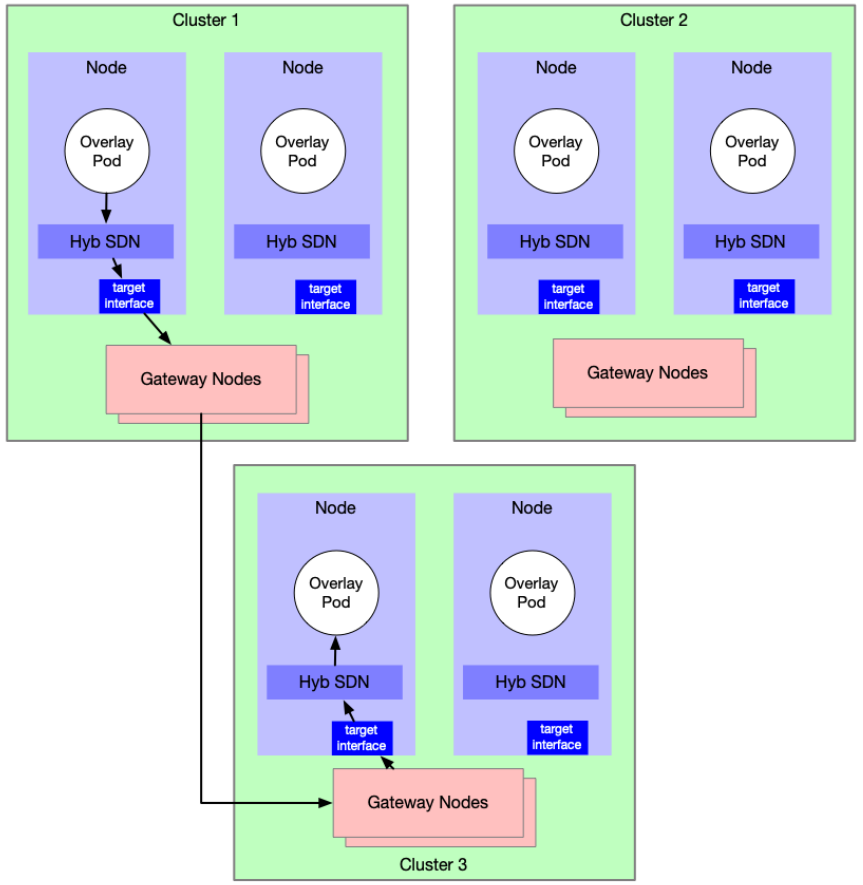
- Cluster mesh
  - P2P mode (HA)
  - No external gateway or storage dependency

Multi-Cluster Connectivity

- Multi-Cluster Overlay Data Path
  - Non-gateway
  - No bottleneck point of traffic & accident
  - Less hops, higher efficiency



VS



Advanced IPAM – IP Retain

- IP Retain
  - Support stateful workload kinds (including custom ones)
  - Add scheduling constraint to workloads automatically
  - Support release reserved IP

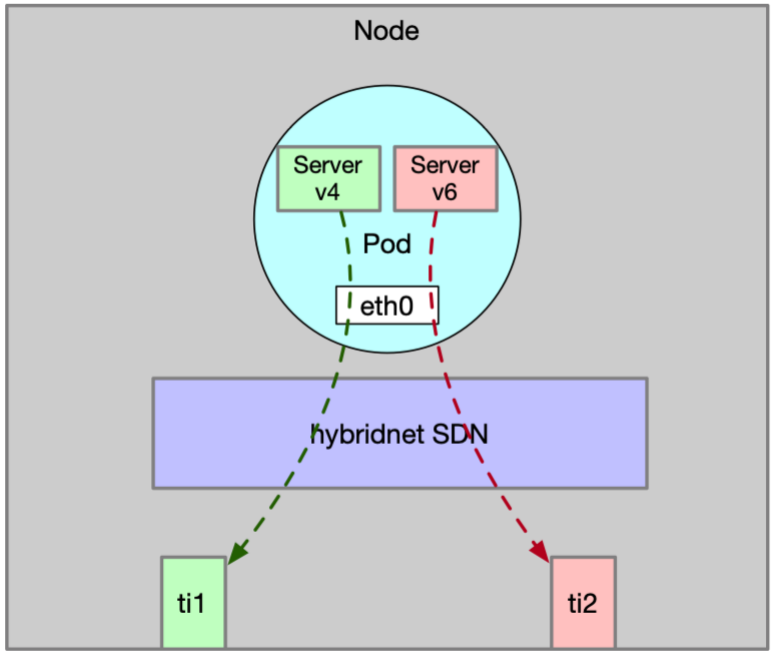
```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: curl-ss
spec:
  selector:
    matchLabels:
      app: curl-ss
  replicas: 3
  serviceName: "curl"
  template:
    metadata:
      annotations:
        networking.alibaba.com/ip-pool: "192.168.56.101,192.168.56.102,192.168.56.254".
        networking.alibaba.com/specified-network: network1
      labels:
        app: curl-ss
```

- Specifying IP
  - Support specifying Network, Subnet & IP
  - Success rate can be assured by subnet *reservedIPs field*

```yaml
apiVersion: networking.alibaba.com/v1
kind: Subnet
metadata:
  name: subnet1
spec:
  network: network1                          # Required. The Network which this Subnet belongs to.

  netID: 0                                   # Optional. Depends on the Network's configuration.
                                             # If the Network's netID is empty, Subnet's netID must not be e
                                             # If the Network's netID is not empty, Subnet's netID must be
                                             # either empty or the same to the Network's netID.
                                             # For an Overlay Network, this field must be empty.

  range:
    version: "4"                             # Required. Can be "4" or "6", for ipv4 or ipv6.

    cidr: "192.168.56.0/24"                  # Required.

    gateway: "192.168.56.1"                  # Optional.
                                             # For Underlay VLAN Network, it refers to ASW gateway ip.
                                             # Gateway address will never be allocated to pods.

    start: "192.168.56.100"                  # Optional. The first usable ip of cidr.

    end: "192.168.56.200"                    # Optional. The last usable ip of cidr.

    reservedIPs: "192.168.56.101","192.168.56.102"    # Optional. The reserved ips for later assignment.
```

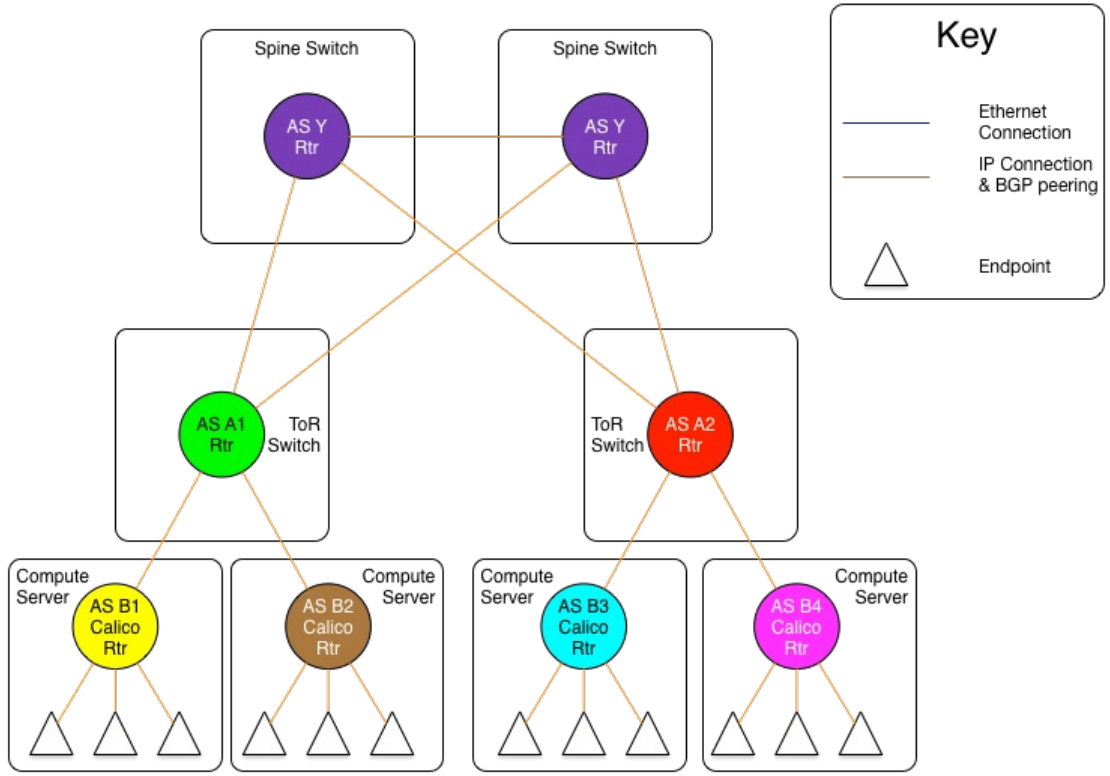
Advanced IPAM – Dual Stack

- DualStack, can be enabled via feature gate
- Supported IP family modes
  - IPv4Only
  - IPv6Only
  - DualStack
- Single interface with Multiple addresses



Multiple Modes of Underlay Network

- Multiple Modes
  - VLAN, bridge & BGP can coexist
- VLAN & Bridge
  - L3 forwarding within host
- BGP
  - Support Downward Default model




KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
Thanks!