GOPS 2019 shenzhen

GOPS 全球运维大会 2019

-AIOps 风向标

深圳站

指导单位：

主办单位：

大会时间：2019年4月12日-13日

大会地址：深圳市南山区圣淘沙大酒店（翡翠店）

云计算时代携程的网络架构变迁

Ctrip Network Architecture Evolution In the Cloud Computing Era

赵亚楠 携程资深架构师

Yanan Zhao, Senior Architect @Ctrip

About Me

- Join Ctrip Cloud @2016
- Currently Lead Ctrip Cloud Network & Storage Team
- Focus
  - Networking
  - Distributed storage
- Blog: https://arthurchiao.github.io

目录

0. About Ctrip Cloud

- Cloud team
  - ~2013
  - OpenStack / Baremetal / Mesos / K8S
- CDOS: unified resource management
- Private cloud
  - VM, BM, Container
- Public cloud
  - Vendors: AWS / Tecent / UCloud / others
  - VM, Container

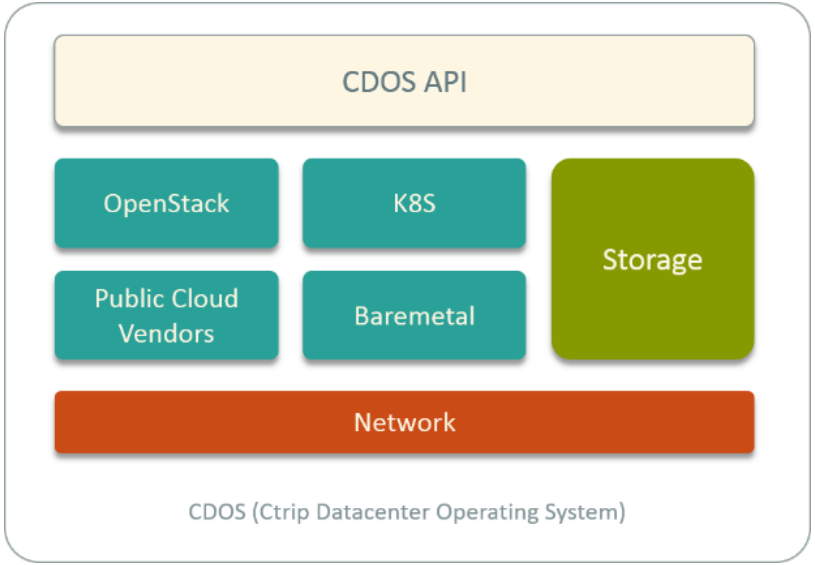


Fig 1. Ctrip Datacenter Operating System (CDOS)

0. About Ctrip Cloud

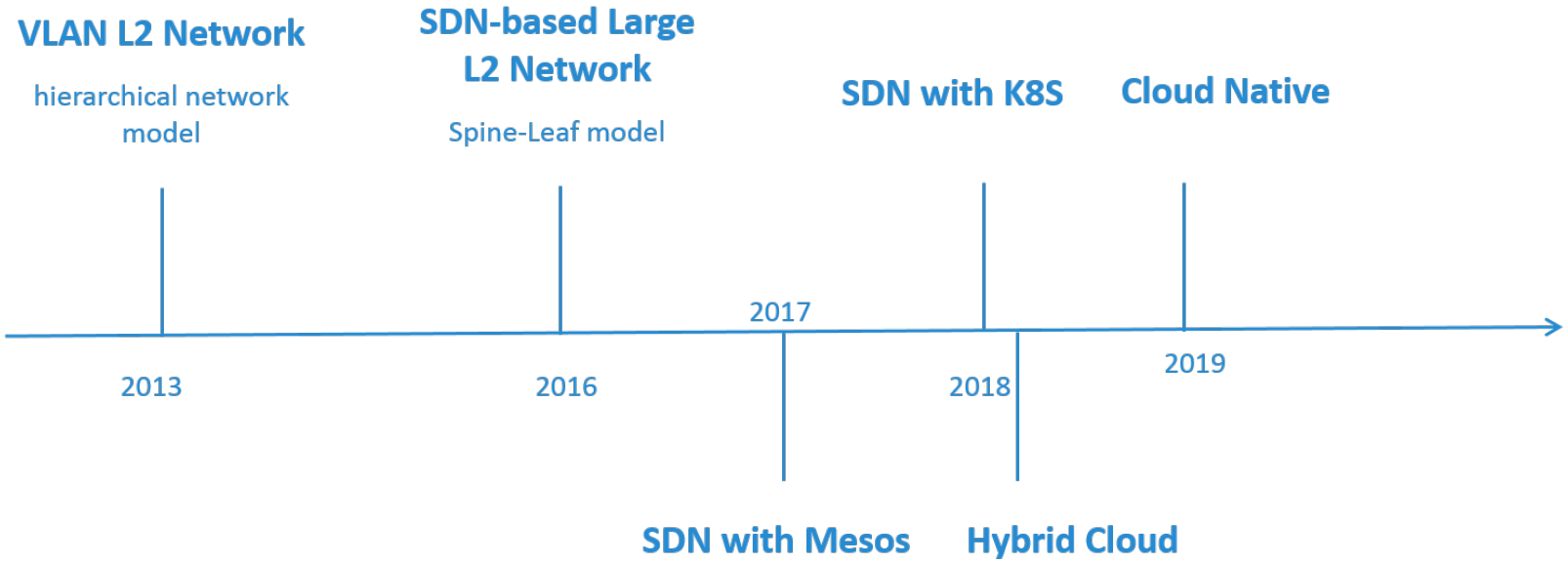


Fig 2. Ctrip network architecture evolution

1. VLAN-based L2 Network

- OpenStack-based private cloud
  - ~2013
  - VM and BM
- Network requirements
  - High Performance
    - Instance-to-instance latency, throughput, etc
  - L2 isolation
  - Routable instance IP
  - Security requirements less critical

Bare Metal

1. VLAN-based L2 Network

- Solution
  - Based on OpenStack “provider network” model [1]
- Advantages
  - Network GW on HW device
  - Instance IP routable
  - Higher performance
    - No overlay encapsulation/decapsulation
    - Routing by HW device

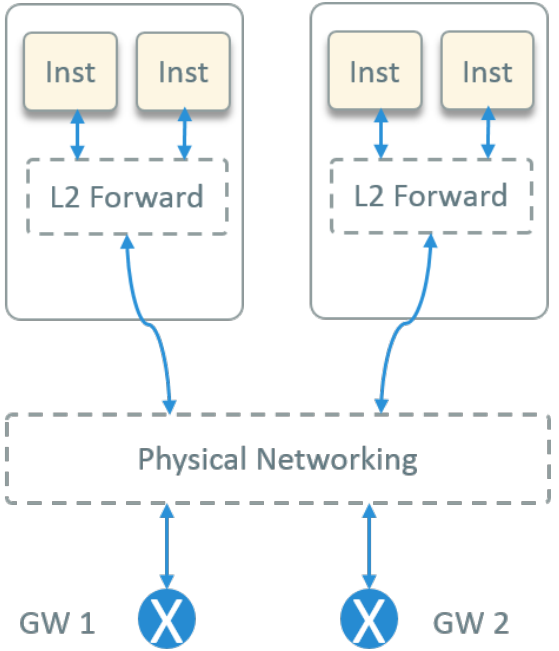


Fig 3. Provider network model in OpenStack

1. VLAN-based L2 Network

- Other aspects
  - L2 segmentation: VLAN
  - ML2: OVS
  - L2 Agent：Neutron OVS Agent
  - L3 Agent: NO
  - DHCP: NO
  - Floating IP: NO
  - Security Group：NO

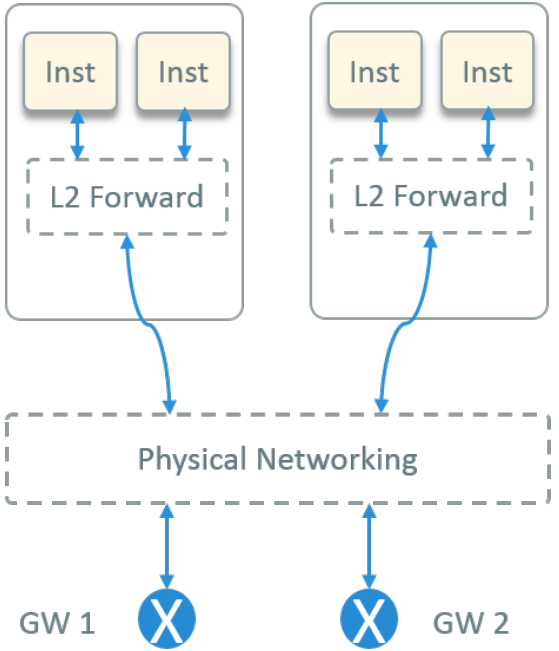


Fig 3. Provider network model in OpenStack

1. VLAN-based L2 Network

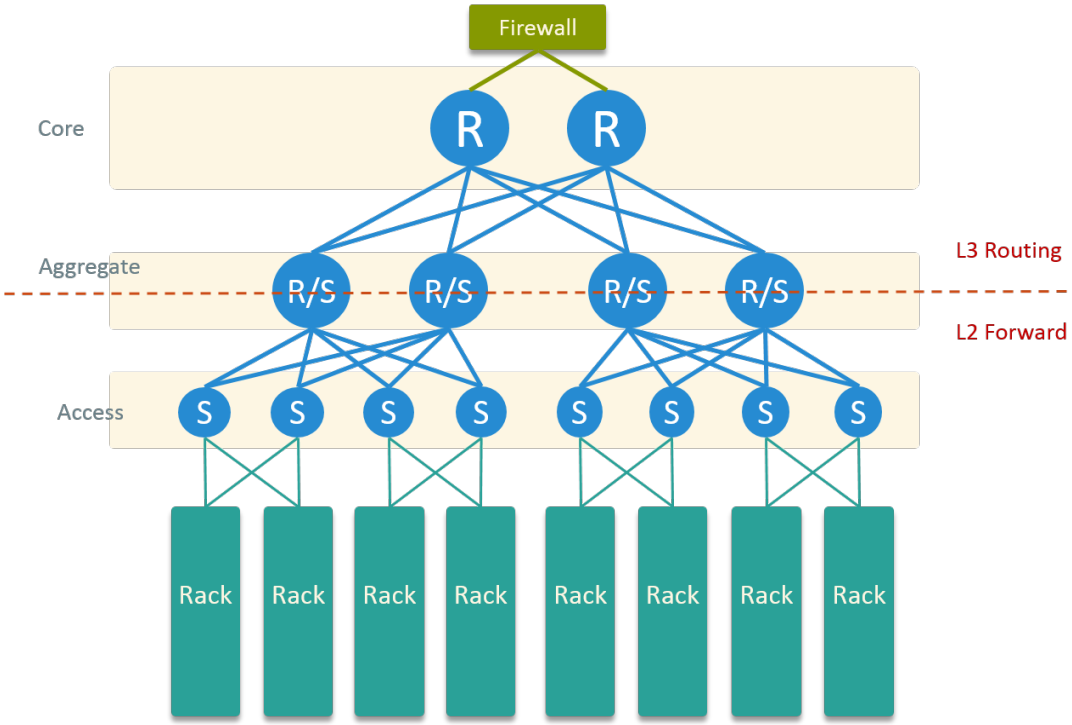


Fig 4. Physical network topology in data center

1. VLAN-based L2 Network

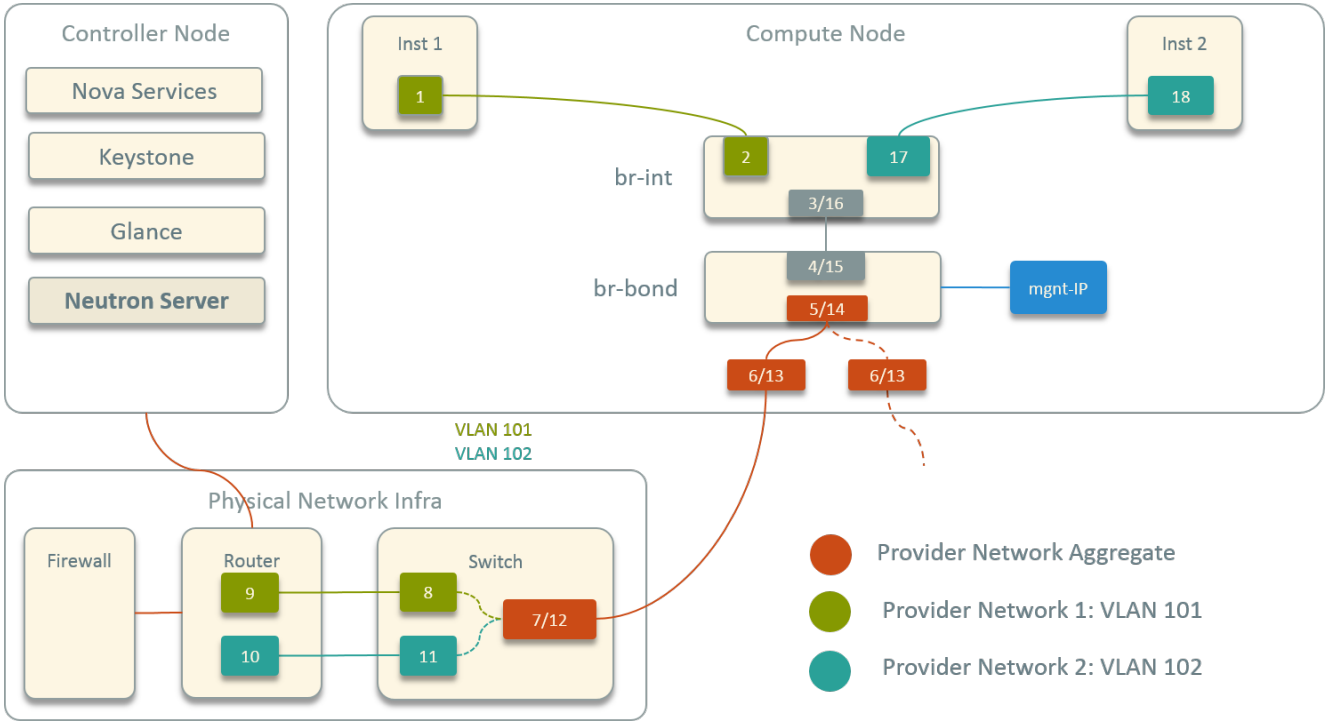


Fig 5. Designed virtual network topology within a compute node

1. VLAN-based L2 Network

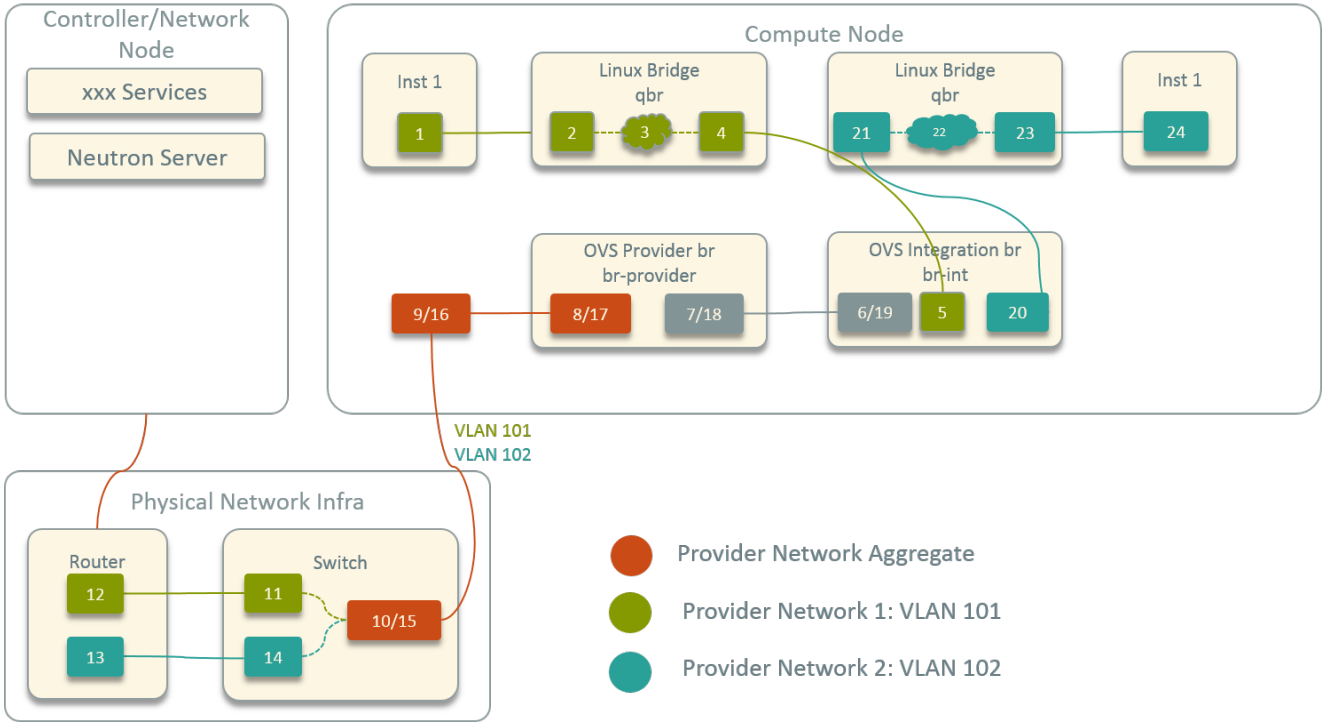


Fig 6. Virtual network topology within a compute host in legacy OpenStack

1. VLAN-based L2 Network

- Advantages
  - Fewer OpenStack components
    - No L3 agent、DHCP agent、neutron metadata agent、network node
    - Ease of Dev & Ops
  - Fewer hops in traffic path, lower latency
    - Instance-to-instance: 24 -> 18 hops
  - GW on HW device, higher performance compared with SW solution
  - Instance IP routable, benefit tracking & monitoring systems
- Disadvantages
  - Security: no security group (compensated by HW firewall)
  - Automation: network/subnet provision relies on HW configuration

2. SDN-based Large L2 Network

- New challenges
  - ~2016
  - Hierarchical network topology: hard to scale
  - Core router: the potential bottleneck, large failure radius
  - Host throughput ceiling: 2 x 1Gbps physical NIC
  - Flooding in large VLAN segments
  - VLAN hard limit: 4096
  - Multi-tenancy & VPC needs
  - Automatic network provision needs

2. SDN-based Large L2 Network

- Solution
  - HW + SW
  - OpenStack + SDN
- Spine-Leaf topology [2]
  - Shorter traversing path
  - Full-mesh connectivity
    - Ease of expansion
    - More resilient to HW failures
- 2 x 10/25Gbps/NIC for each host

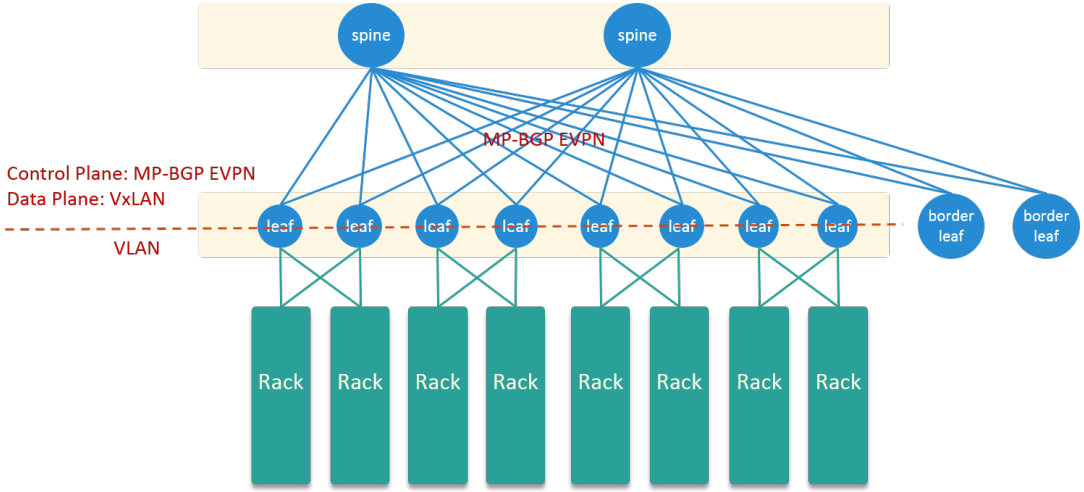


Fig 7. Network topology in new data center: Spine-Leaf

2. SDN-based Large L2 Network

- Custom SDN solution
- Separated control & data plane [2]
  - Data plane: VxLAN
  - Control plane: MP-BGP-EVPN
- Distributed GW
- Multi-Tenancy

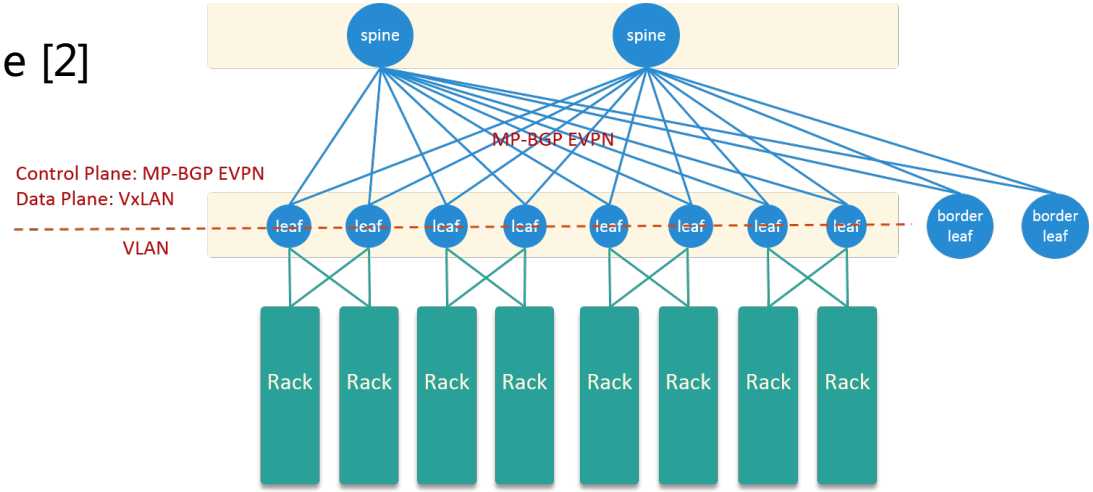


Fig 7. Network topology in new data center: Spine-Leaf

2. SDN-based Large L2 Network

- CNC: Ctrip Network Controller
  - Central SDN controller
  - Mange all Spine and Leaf nodes
  - Dynamic configurations to Spine/Leaf
  - Integration with Neutron server
- Neutron server
  - Add CNC ML2 & L3 plugins
  - New finite state machine (FSM) for port status
  - New APIs interact with CNC
  - DB schema changes

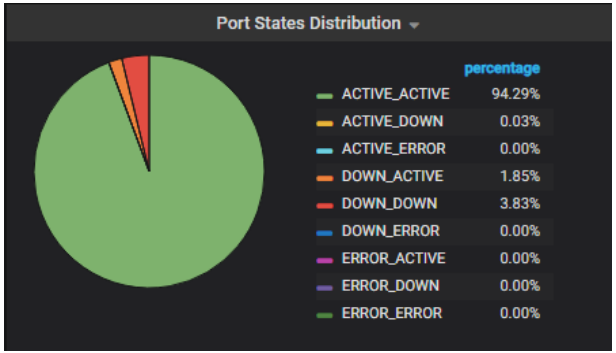


Fig 8. Monitoring panel for the neutron ports' states

2. SDN-based Large L2 Network

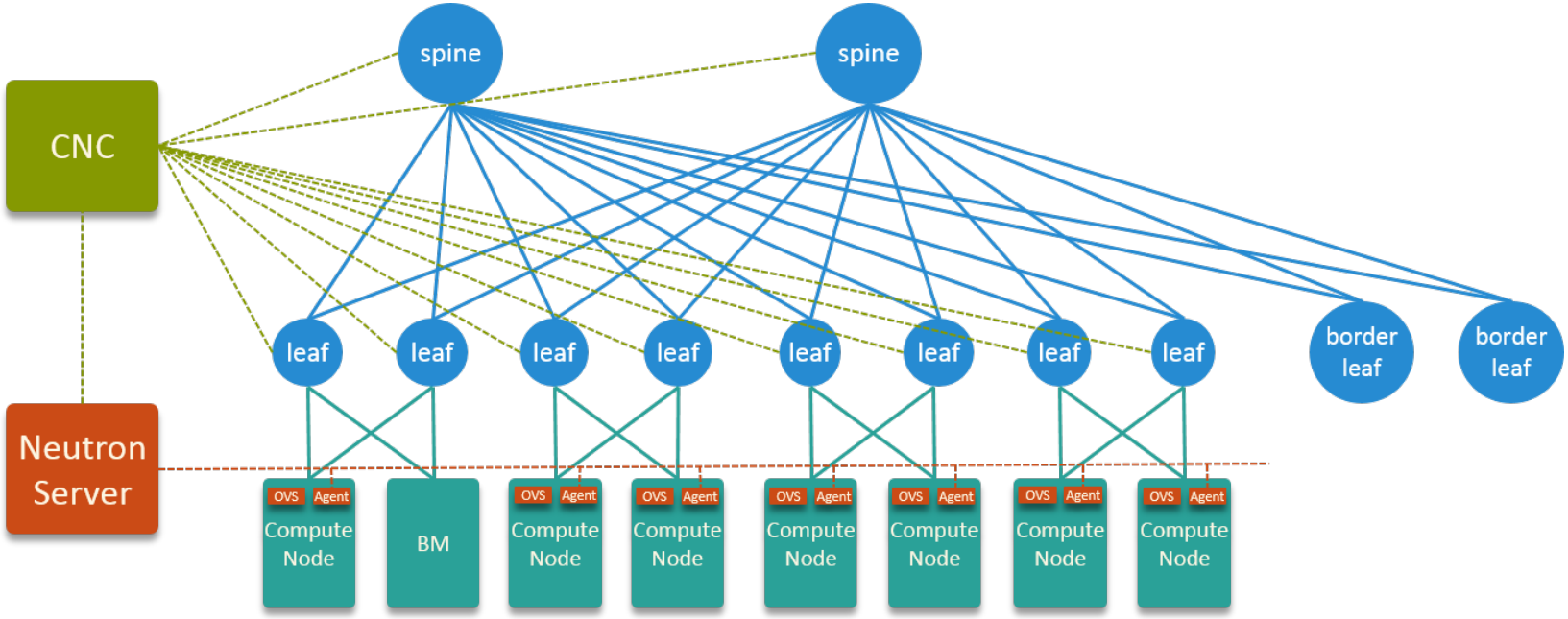


Fig 9. HW + SW topology of the designed SDN solution

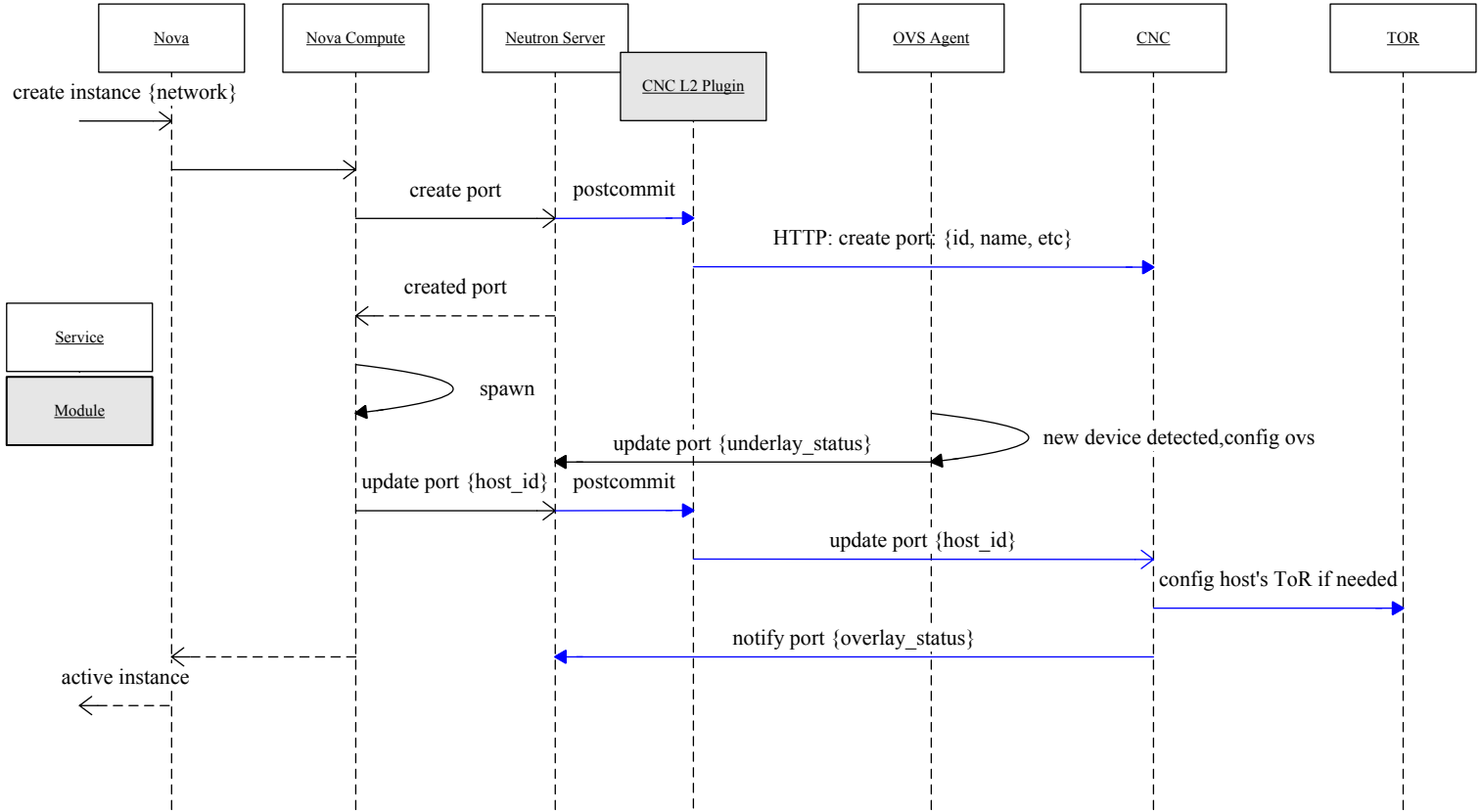


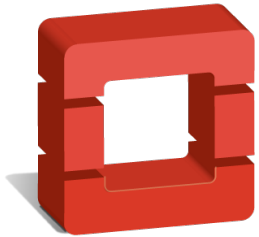
Fig 10. Network setup steps during instance spawning

2. SDN-based Large L2 Network

- Summary
  - HW
    - Shorter traversing path in physical network
    - Distributed gateway
    - More resilient to HW failures, ease of expansion
  - SW
    - Central SDN controller, integrate with Neutron via plugins
    - Dynamic configuration to HW devices
    - Support both VM and BM provision
  - Multi-tenancy & VPC support

3 K8S & Hybrid Network

- Container platform
  - ~2017
  - Migrate some apps from VM/BM to container
- Container platform characteristics
  - Large scale instances, 10K ~ 100K containers per cluster
  - Higher deploy/destroy frequencies
  - Shorter spawn/destroy time: ~10s (VM: ~100s)
  - Container failure/drifting is the norm rather than exception



3 K8S & Hybrid Network

- Network Requirements
  - High performance, concurrent network APIs
  - Compatibility with existing systems
    - Container drifting with the same IP
  - Host agent/binary
    - Fast add/delete network for containers
- Solution: extend SDN to support Mesos/K8S
  - Reuse existing infrastructures
    - Neutron, CNC, OVS, Neutron-OVS-Agent
  - Develop a CNI plugin for neutron

3 K8S & Hybrid Network

- Neutron changes
  - New APIs
    - Allocate port by network labels
  - Performance Optimization
    - Bulk port API
    - Database access optimizations
    - Async API for high concurrency
    - Critical path refactor
  - Backport new features from upstream
    - Graceful OVS agent restart

3 K8S & Hybrid Network

- K8S CNI plugin for neutron
  - Counterpart of the libvirt network driver in VM provision
  - Create veth pair, attach to OVS and container netns
  - Configure MAC, IP, GW, etc
  - Update port to neutron server
- Existing network services/components upgrade
  - OVS: 2.3.1 LTS -> 2.5.6 LTS
    - ovs-vswitchd 100% CPU bug [3]
    - OVS port mirror bug [4]

3 K8S & Hybrid Network

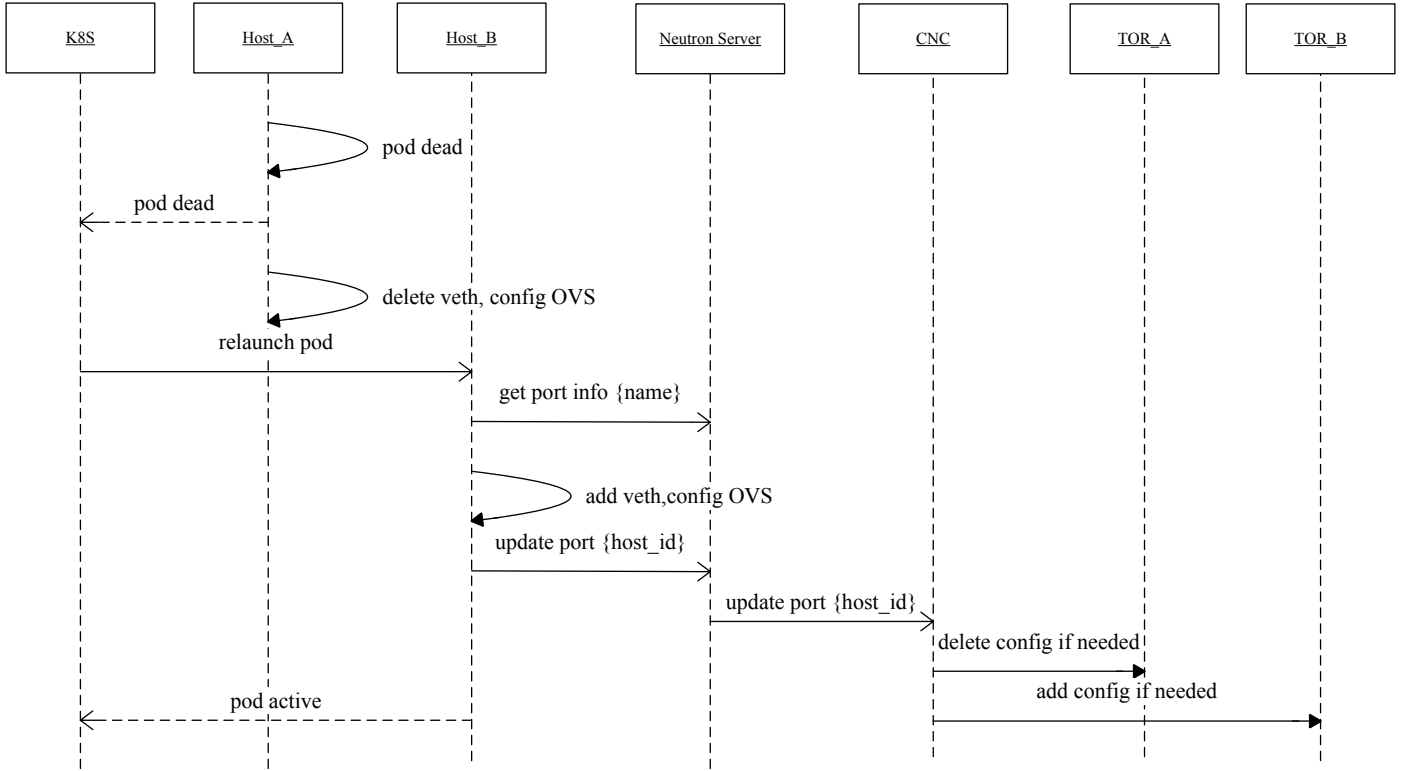


Fig 11. Pod drifting with the same IP within a K8S cluster

3 K8S & Hybrid Network

- Summary
  - Quickly integrate container platform into existing infra
  - Single global IPAM manages VM/BM/container network
- Current deployment scale
  - 4 availability zones (AZ)
  - Up to 500+ physical nodes (VM/BM/Container hosts) per AZ
  - Up to 500+ instances per host
  - Up to 20K+ instances per AZ

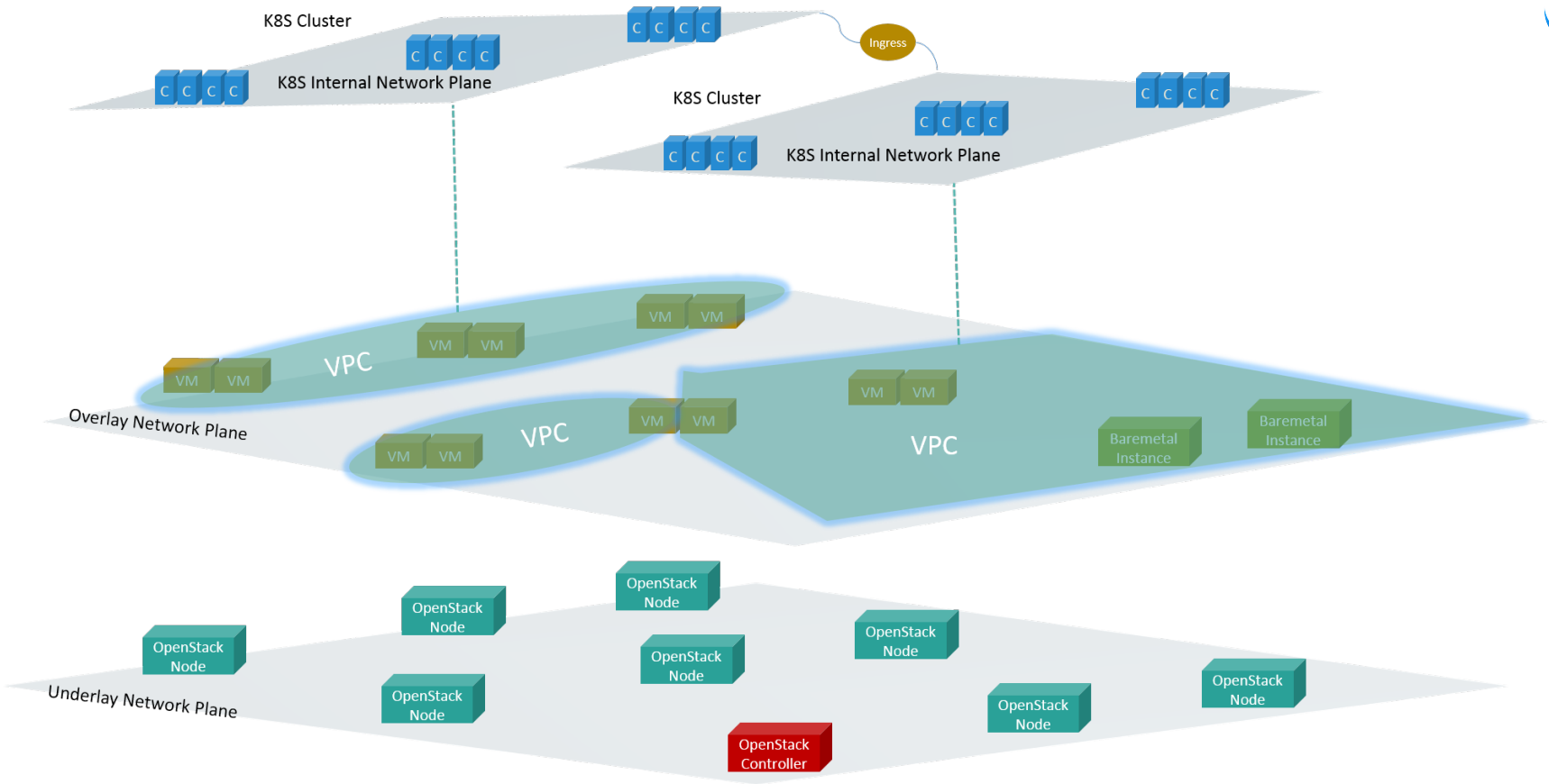


Fig 12. Layered view of the future network architecture

3 K8S & Hybrid Network

- Global deployment needs
  - ~2018
  - Private overseas DC: long design & building period
  - Public cloud vendors: quickly integrate into current (private cloud) infra
- Solution
  - VM/BM instances from public cloud vendors
  - Custom deployed and maintained K8S clusters
  - CDOS API: abstract vendor-specific details
  - Networking solution

3 K8S & Hybrid Network

- Network solution for K8S cluster (AWS)
  - Global IPAM
  - CNI
    - plug/unplug ENI to EC2 [5, 6]
    - Support attach/detach floating IP
  - ENI
    - As Pod network interface
    - One ENI dedicated to one Pod
    - Drifting with Pod
  - IP drifts with Pod

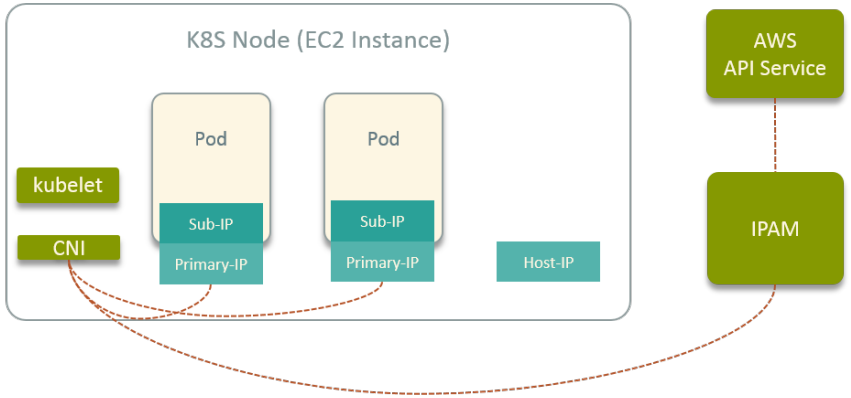


Fig 13. K8S network solution on public cloud vendor (AWS)

3 K8S & Hybrid Network

- Global networking
  - VPC
    - Private cloud
    - Public cloud vendors
    - Non-overlapped CIDR from private cloud VPC
  - Interconnect private & public VPC with Direct Connect
  - IP routable between private & public cloud if needed

Fig 14. VPCs distributed over the globe

4. Cloud Native Solutions

- New challenges faced
  - IPAM
    - Central IPAM may be the new bottleneck
    - Neutron is not designed for performance
    - Cloud native: prefer local IPAM (IPAM per host)
  - Large failure radius: IP drifting among entire AZ
  - Dense deployment of containers will hit HW limit of leaf nodes
  - Increasingly strong host firewall (L4-L7) needs
- Candidates
  - Calico/Cilium/Others

4. Cloud Native Solutions

- Brand-new solution [7]
- Kernel 4.8+
- eBPF/BPF: extended Berkeley Packet Filter
- BPF-based connectivity & security
- L4-L7 network policy
- Components
  - CLI
  - Plugin for orchestrator integration
  - Policy repository
  - Host agent

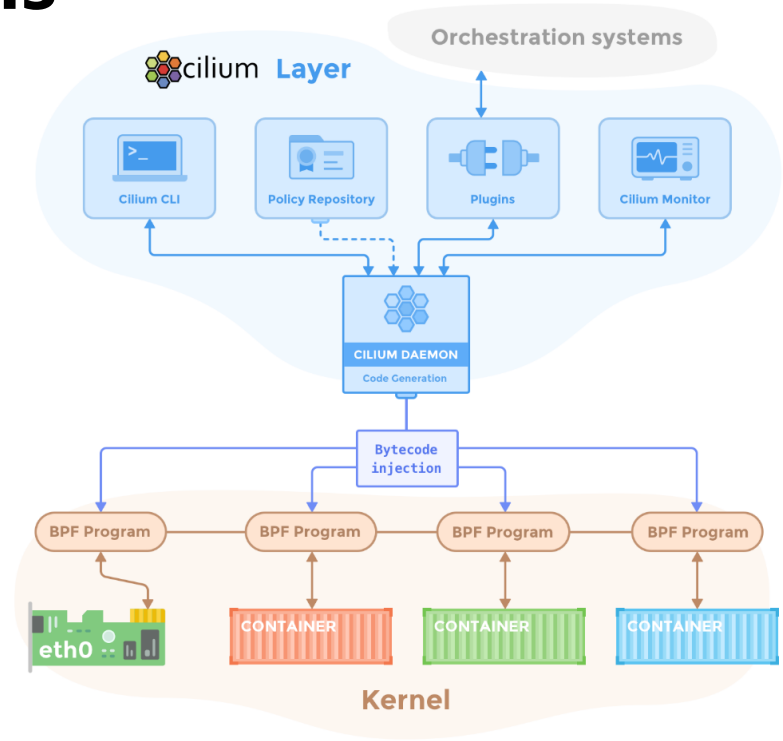


Fig 15. Cilium

4. Cloud Native Solutions

- Host Networking
  - Per-host CIDR
  - Gateway on host device
  - Inst-to-inst: BPF + Kernel Stack L2 forward
  - Inst-to-host: BPF + L3 Routing
- Cilium Agent
  - Listen endpoint changes
- CNI plugin
  - Create & configure veth pair
  - Generate BPF code

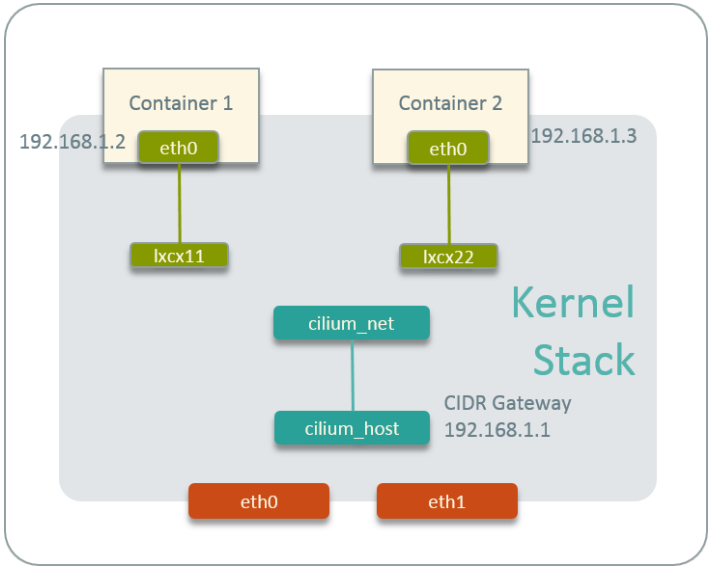


Fig 16. Cilium host networking

4. Cloud Native Solutions

- Multi-host networking
  - VxLAN
    - Overlay
    - Software VTEP in host
  - BGP
    - Private cloud
    - Public cloud BGP API

4. Cloud Native Solutions

- Advantages
  - K8S-native L4-L7 security policy support
  - High performance network policy enforcement
    - Theoretical complexity: BPF O(1) vs iptables O(n)
  - High performance forwarding plane (veth pair, IPVLAN)
  - Dual stack support (IPv4/IPv6)
  - Support run over flannel (Cilium only handles network policy)
  - Active community
    - Development driven by a company
    - Core developers from kernel community

4. Cloud Native Solutions

- Disadvantages
  - Latest kernel (4.8+ at least, 4.14+ better)
  - Not enough user stories & best practices yet
  - Higher dev & ops cost
    - Kernel stack (data structure, packet traversing path, etc)
    - BPF knowledge
  - Not enough trouble shooting tools (e.g. tracing, debug)
- Have a try and find the fun!

Q & A

References

1. OpenStack Doc: Networking Concepts, https://docs.openstack.org/neutron/rocky/admin/intro-os-networking.html
2. Cisco Data Center Spine-and-Leaf Architecture: Design Overview, https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white-paper-c11-737022.pdf
3. ovs-vswitchd: Fix high cpu utilization when acquire idle lock fails, https://mail.openvswitch.org/pipermail/ovs-dev/2014-October/290600.html
4. openvswitch port mirroring only mirrors egress traffic, https://bugs.launchpad.net/cloud-archive/+bug/1639273
5. Lyft CNI plugin, https://github.com/lyft/cni-ipvlan-vpc-k8s
6. Netflix: run container at scale, https://www.slideshare.net/aspyker/container-world-2018
7. Cilium Project, https://cilium.io/
8. Cilium Cheat Sheet, https://arthurchiao.github.io/blog/cilium-cheat-sheet/
9. Cilium Code Walk Through: CNI Create Network, https://arthurchiao.github.io/blog/cilium-code-walk-through-create-network/

Thanks

高效运维社区

开放运维联盟

荣誉出品

想第一时间看到高效运维社区的新动态吗？

高维在线