

Open Source AceCon

2021 智能云边开源峰会

AI x Cloud Native x Edge Computing

人工智能 × 云原生 × 边缘计算

Kube-OVN

一个非著名K8s CNI 网络插件

刘韬

灵雀云工程师, Kube-OVN
maintainer

目录

- ✓ | 为何要 Kube-OVN?
- ✓ | 什么是 Kube-OVN?
- ✓ | 怎么用 Kube-OVN

为何要有 Kube-OVN?

对 Kubernetes 网络:

- 如何兼容传统网络架构，容器网络需和已有基础实施互通互联
- 传统应用对固定 IP 的需求
- 容器多网络平面，多网卡的管理
- 多租户，VPC 类型容器网络需求
- 多集群网络互通，跨云容器网络成为难题



什么是 Kube-OVN?

面向企业场景的容器网络

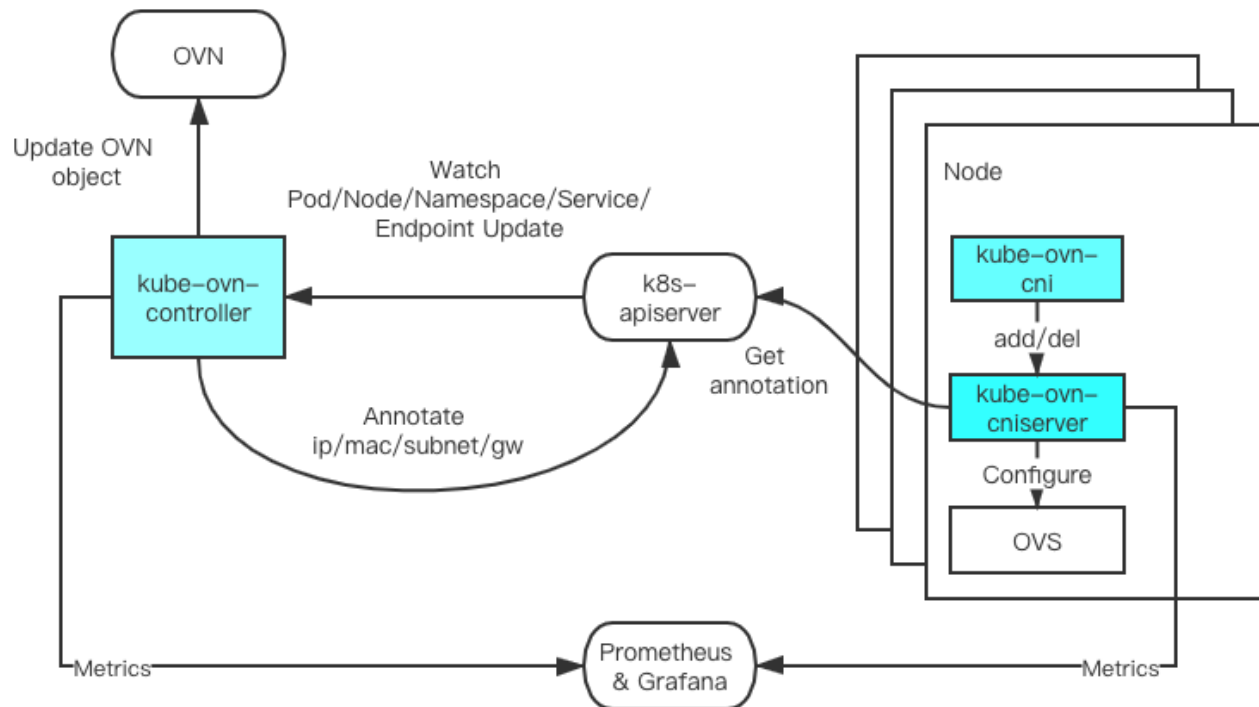


- 利用社区最为成熟的 OVS 作为网络底座
- 基于 Kubernetes 架构原生设计
- 结合灵雀云企业端多年实践打造功能
- 复用 OVS 社区的生态

什么是 Kube-OVN?

组件架构

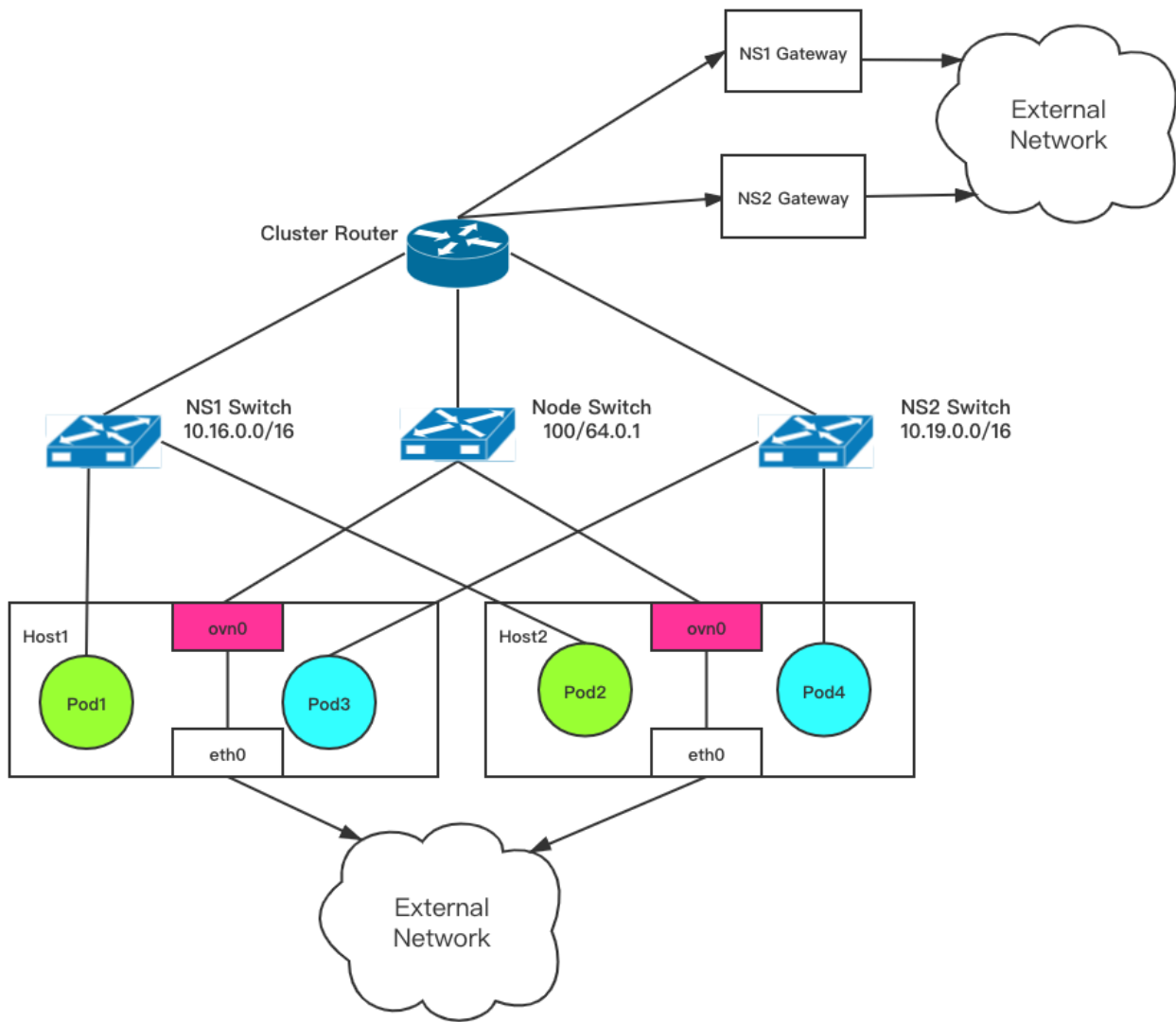
- 基于标准 K8s API, 通过Operator模式进行开发
- 所有功能和配置可通过 Annotation, CRD 完成
- 第三方可根据自己需求进行界面调整



什么是 Kube-OVN?

Overlay 网络

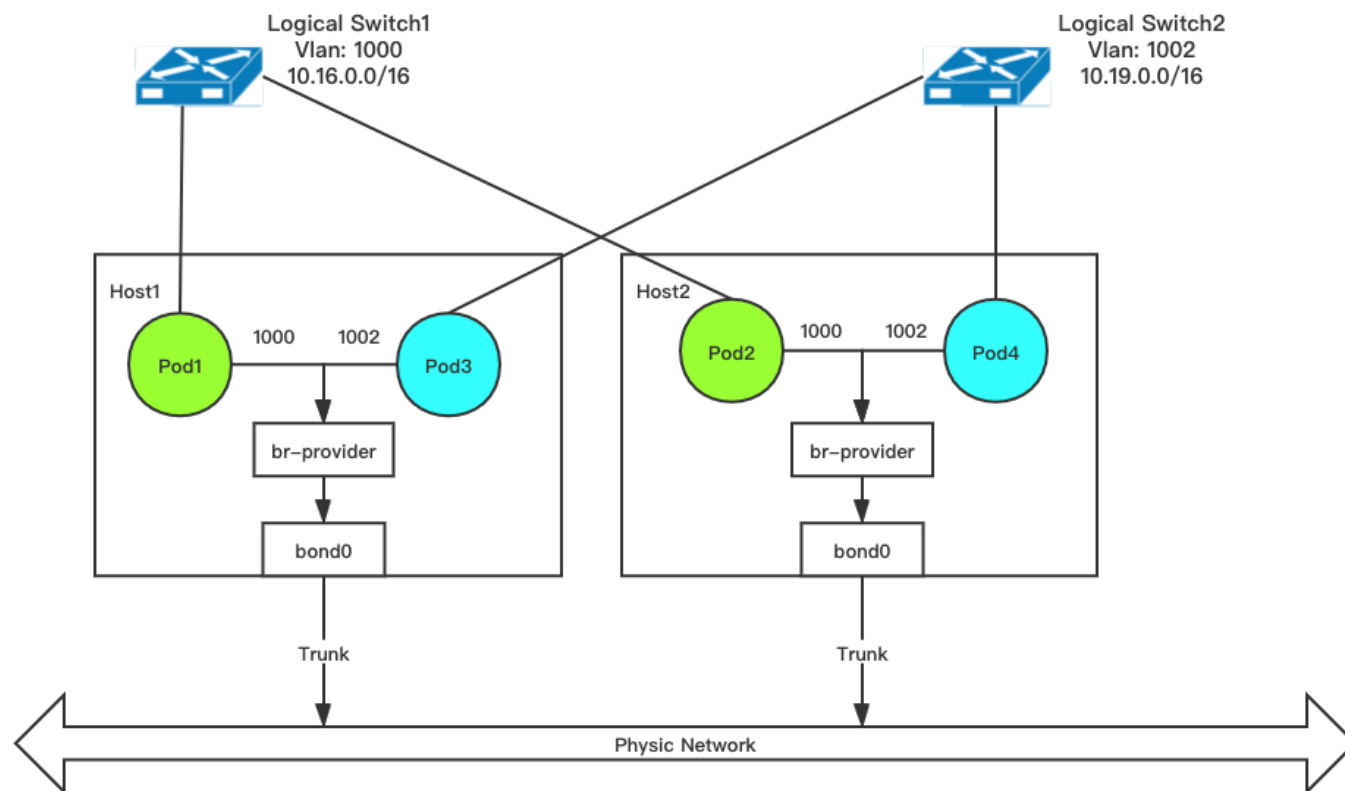
- Geneve Overlay 网络，可灵活添加网络功能(VPC, ACL, etc.)，并和现有物理网络保持独立
- 子网和主机无关，容器 IP 地址可以在整个集群漂移
- 子网绑定 Namespace，方便多租户地址空间管理，以及地址和项目，应用进行关联
- 分布式网关和集中式网关两种出网方式，每个命名空间可以有独立的网络控制



什么是 Kube-OVN?

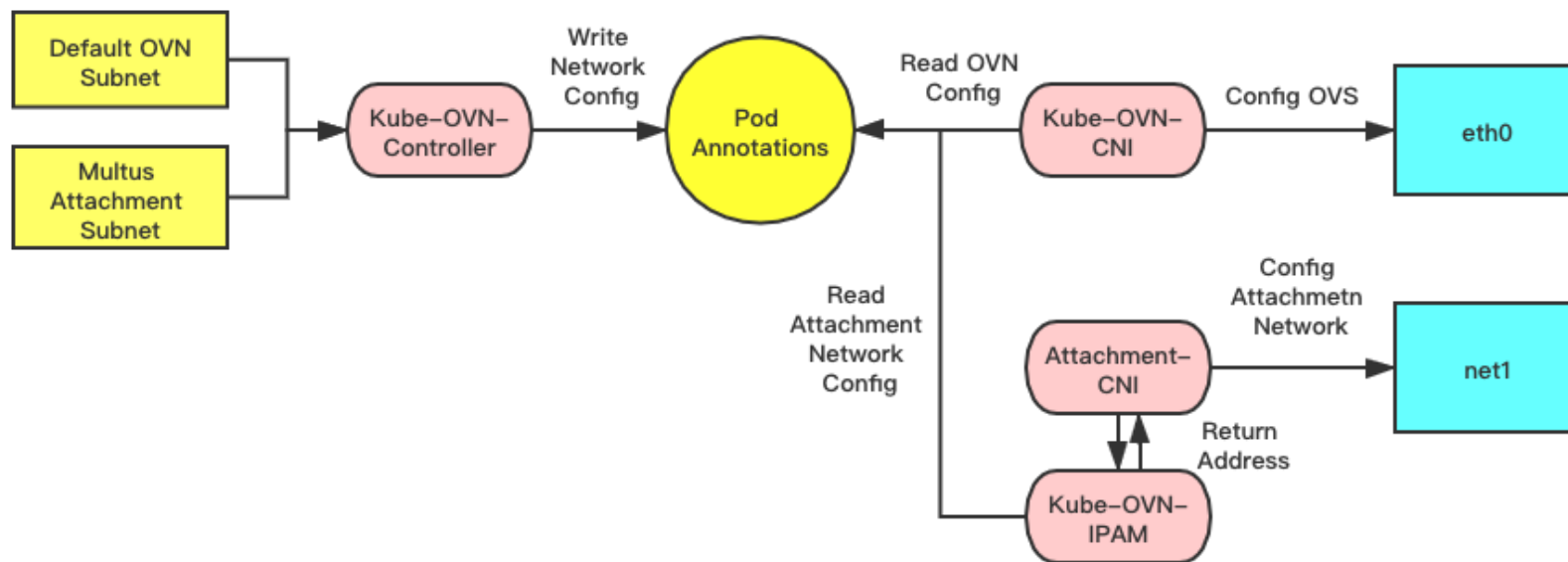
Underlay 网络

- Underlay 网络，和物理网络直接打通，提供直连底层结构的高性能网络.
- 支持 Pod 运行在不同 Vlan 网络中，可通过 Vlan 实现隔离
- 提供固定 IP/Mac 的能力
- 提供 Underlay 与 Overlay 共存能力



什么是 Kube-OVN?

容器多网卡管理

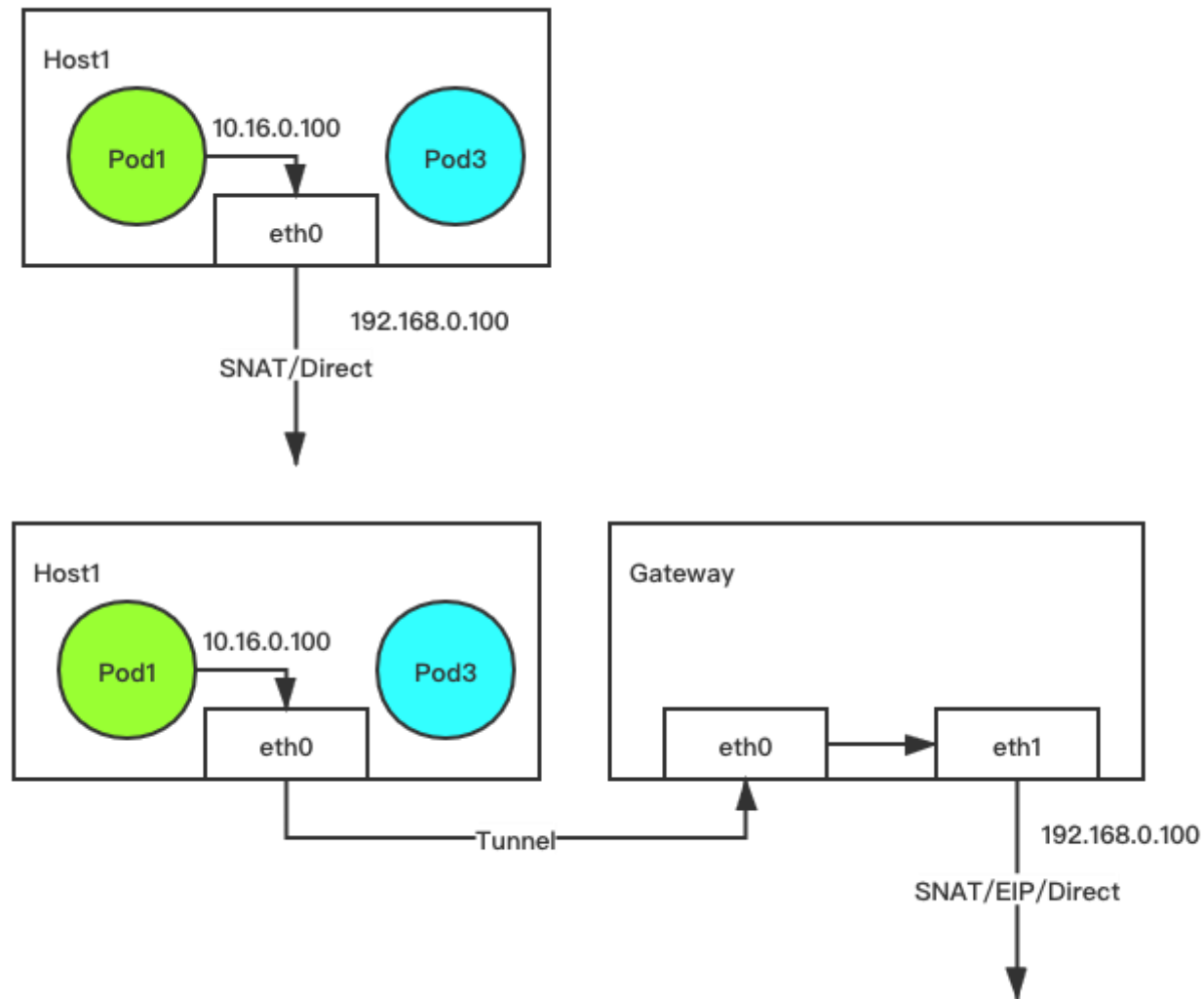


- 特殊业务需要多网络：网关，VNF
- 统一管理主网络和附属网络
- 为附属网络提供子网，固定 IP 等功能

什么是 Kube-OVN?

容器网络和外部网络打通

- Namespace 级别出网控制
 - 分布式网关
 - 集中式网关
 - NAT 控制
- Pod 级别出网控制
 - SNAT
 - EIP
- Underlay 方案
 - Pod 接入底层 Vlan，利用物理能力打通
- 打通其他网络
 - 与 Openstack 共享底层 ovn 网络



怎么用 Kube-OVN?

简单的使用方法

- 工作负载固定地址方法:
- Pod 可通过 annotation 指定固定 IP/MAC
- Workload 可通过 ippool annotation 指定一组 IP, 达到固定 IP 效果
- StatefulSet 特殊优化, Pod 名和 IP 一一对应
- IP 地址可在整个集群内漂移

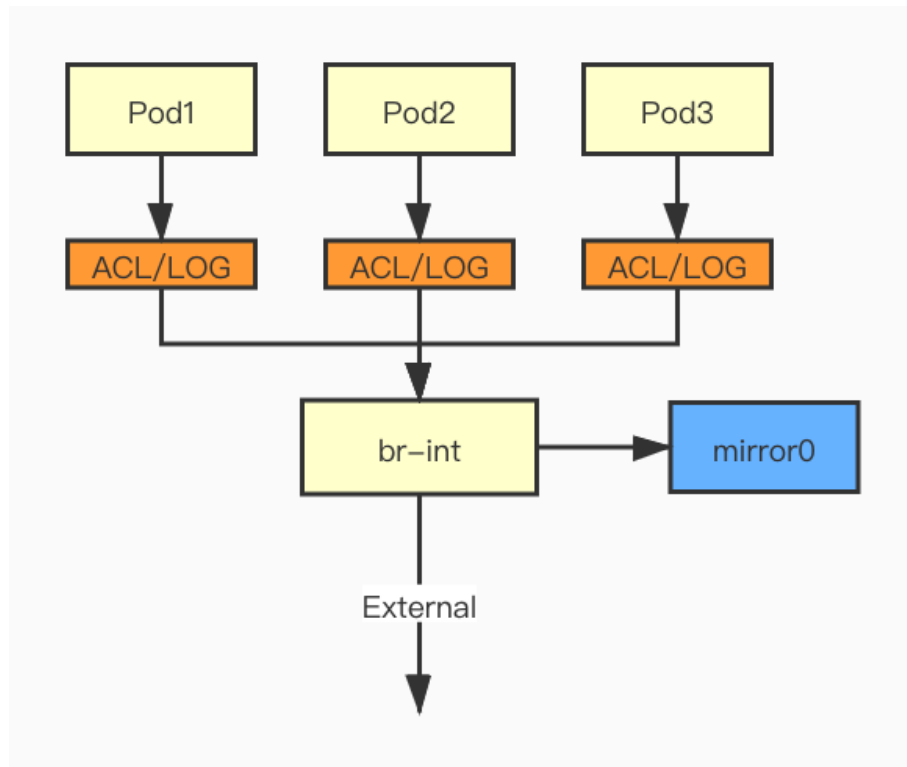
```
apiVersion: v1
kind: Pod
metadata:
  name: static-ip
  namespace: ls1
  annotations:
    ovn.kubernetes.io/ip_address: 10.16.0.15
    ovn.kubernetes.io/mac_address: 00:00:00:53:6B:B6
spec:
  containers:
    - name: static-ip
      image: nginx:alpine
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: ls1
  name: starter-backend
  labels:
    app: starter-backend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: starter-backend
  template:
    metadata:
      labels:
        app: starter-backend
      annotations:
        ovn.kubernetes.io/ip_pool: 10.16.0.15,10.16.0.16,10.16.0.17
```

怎么用 Kube-OVN?

容器网络安全

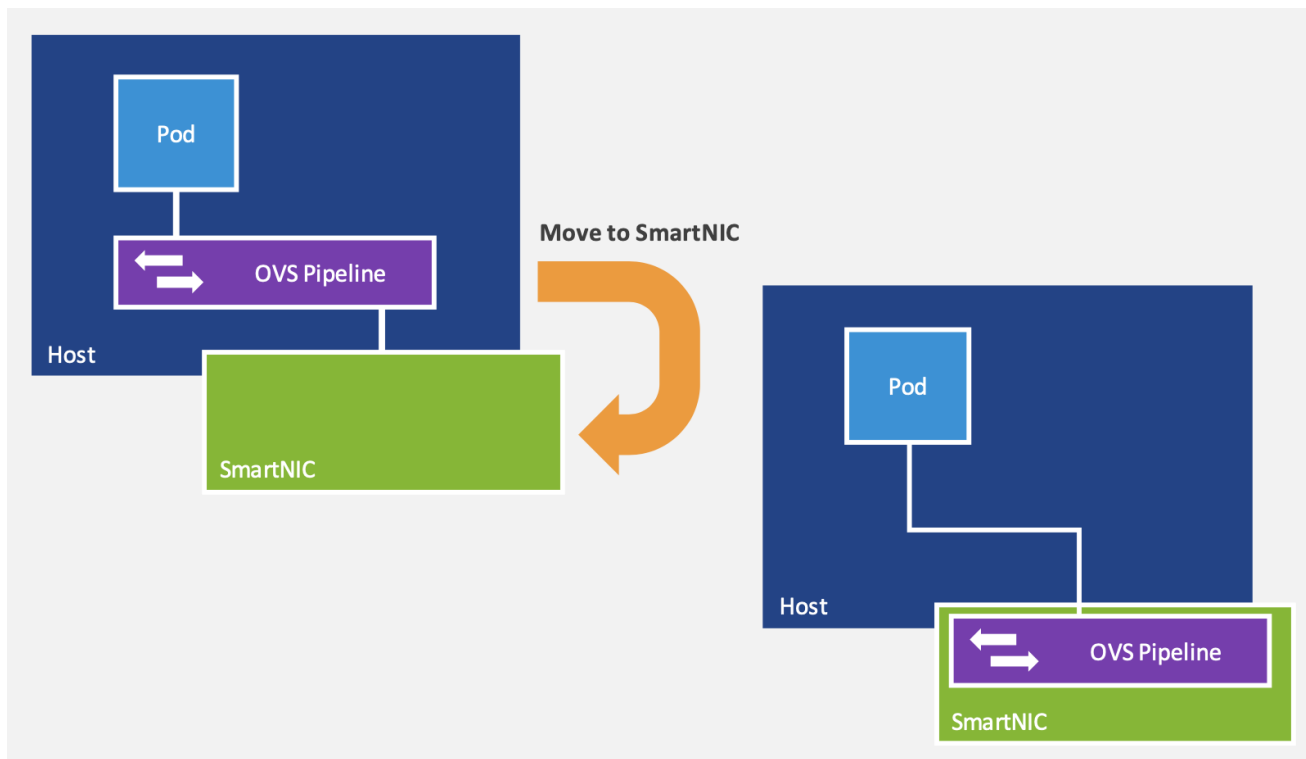
- 标准 NetworkPolicy 支持
- 子网 ACL 控制
- ACL 日志记录规则拦截数据
- PortSecurity 支持
- 容器流量全量镜像，便于安全审计，流量分析
- 动态 QoS 限制双向带宽



怎么用 Kube-OVN?

容器网络性能

OVN调优，单集群支撑 200节点 1w+ Pod
流表实现 Service 避免 Iptables 性能损耗
硬件加速方案，卸载所有流表匹配至智能网卡
跨节点通信，单核吞吐量一倍提升



怎么用 Kube-OVN?

完善的监控体系

OVN/OVS 的完整监控

集群网络质量实时监控

- Node/Pod/Service/DNS 连通性
- Node/Pod/Service/DNS 网络

Tracing/Tcpdump 工具支持, 性能测试工具支持.

容器网络流量完整镜像

Prometheus/Grafana 集成

<https://github.com/alauda/kube-ovn/blob/master/docs/ovn-ovs-monitor.md>

怎么用 Kube-OVN?

完善的监控体系

Trace

```
[root@liumengxin-ovn1-192 ~]# kubectl ko trace default/web-0 114.114.114.114 udp 53
Defaulting container name to ovn-central.
Use 'kubectl describe pod/ovn-central-748d766476-rnzkk -n kube-system' to see all of the
+ kubectl exec ovn-central-748d766476-rnzkk -n kube-system -- ovn-trace --ct=new ovn-def
== 114.114.114.114 && udp.src == 10000 && udp.dst == 53'
Defaulting container name to ovn-central.
Use 'kubectl describe pod/ovn-central-748d766476-rnzkk -n kube-system' to see all of the
# udp,reg14=0x2d,vlan_tci=0x0000,d1_src=00:00:00:38:06:e7,d1_dst=00:00:00:8d:3a:fc,nw_sr

ingress(dp="ovn-default", inport="web-0.default")
-----
0. ls_in_port_sec_l2 (ovn-northd.c:4769): inport == "web-0.default", priority 50, uuid
next;
20. ls_in_l2_lkup (ovn-northd.c:7542): eth.dst == 00:00:00:8d:3a:fc, priority 50, uuid d
output = "ovn-default-ovn-cluster";
output;

egress(dp="ovn-default", inport="web-0.default", output="ovn-default-ovn-cluster")
-----
1. ls_out_pre_lb (ovn-northd.c:5118): ip, priority 100, uuid f1507423
reg0[0] = 1;
next;
3. ls_out_pre_stateful (ovn-northd.c:5135): reg0[0] == 1, priority 100, uuid 054a7b02
ct_next;

ct_next(ct_state=new|trk)
-----
10. ls_out_port_sec_l2 (ovn-northd.c:4835): output == "ovn-default-ovn-cluster", priori
output;
/* output to "ovn-default-ovn-cluster", type "patch" */

ingress(dp="ovn-cluster", inport="ovn-cluster-ovn-default")
-----
```

Tcpdump

```
[root@liumengxin-ovn1-192 ~]# kubectl ko tcpdump kube-system/kube-ovn-pinger-dpqt2
+ kubectl exec -it kube-ovn-cni-6zkfc -n kube-system -- tcpdump -nn -i 6454a509c784_h
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on 6454a509c784_h, link-type EN10MB (Ethernet), capture size 262144 bytes
06:05:10.374863 IP 10.16.0.2 > 192.168.16.44: ICMP echo request, id 14683, seq 2, length 24
06:05:10.375198 IP 192.168.16.44 > 10.16.0.2: ICMP echo reply, id 14683, seq 2, length 24
06:05:10.475656 IP 10.16.0.2 > 192.168.16.45: ICMP echo request, id 28537, seq 0, length 24
06:05:10.475991 IP 192.168.16.45 > 10.16.0.2: ICMP echo reply, id 28537, seq 0, length 24
06:05:10.575899 IP 10.16.0.2 > 192.168.16.45: ICMP echo request, id 28537, seq 1, length 24
06:05:10.576217 IP 192.168.16.45 > 10.16.0.2: ICMP echo reply, id 28537, seq 1, length 24
06:05:10.675879 IP 10.16.0.2 > 192.168.16.45: ICMP echo request, id 28537, seq 2, length 24
06:05:10.676252 IP 192.168.16.45 > 10.16.0.2: ICMP echo reply, id 28537, seq 2, length 24
06:05:10.776784 IP 10.16.0.2 > 192.168.16.46: ICMP echo request, id 3032, seq 0, length 24
06:05:10.776856 IP 192.168.16.46 > 10.16.0.2: ICMP echo reply, id 3032, seq 0, length 24
06:05:10.877009 IP 10.16.0.2 > 192.168.16.46: ICMP echo request, id 3032, seq 1, length 24
06:05:10.877063 IP 192.168.16.46 > 10.16.0.2: ICMP echo reply, id 3032, seq 1, length 24
06:05:10.977129 IP 10.16.0.2 > 192.168.16.46: ICMP echo request, id 3032, seq 2, length 24
06:05:10.977221 IP 192.168.16.46 > 10.16.0.2: ICMP echo reply, id 3032, seq 2, length 24
06:05:11.077668 IP 10.16.0.2.41252 > 10.96.0.1.443: Flags [P.], seq 2749951050:2749951089, ac
06:05:11.077924 IP 10.96.0.1.443 > 10.16.0.2.41252: Flags [.], ack 39, win 59, options [nop,r
06:05:11.081087 IP 10.96.0.1.443 > 10.16.0.2.41252: Flags [P.], seq 1:58, ack 39, win 59, opt
06:05:11.081108 IP 10.96.0.1.443 > 10.16.0.2.41252: Flags [.], seq 58:1406, ack 39, win 59, d
06:05:11.081115 IP 10.96.0.1.443 > 10.16.0.2.41252: Flags [.], seq 1406:2751, ack 39, win 59
```

怎么用 Kube-OVN?

社区情况

- Apache 2.0 开源，商业版和社区版功能完全一致
- 入选 CNCF Landscape
- 天翼云，金山云，华为，锐捷，Intel 等公司参与社区贡献
- 30+ 社区用户



AsiaInfo 亚信

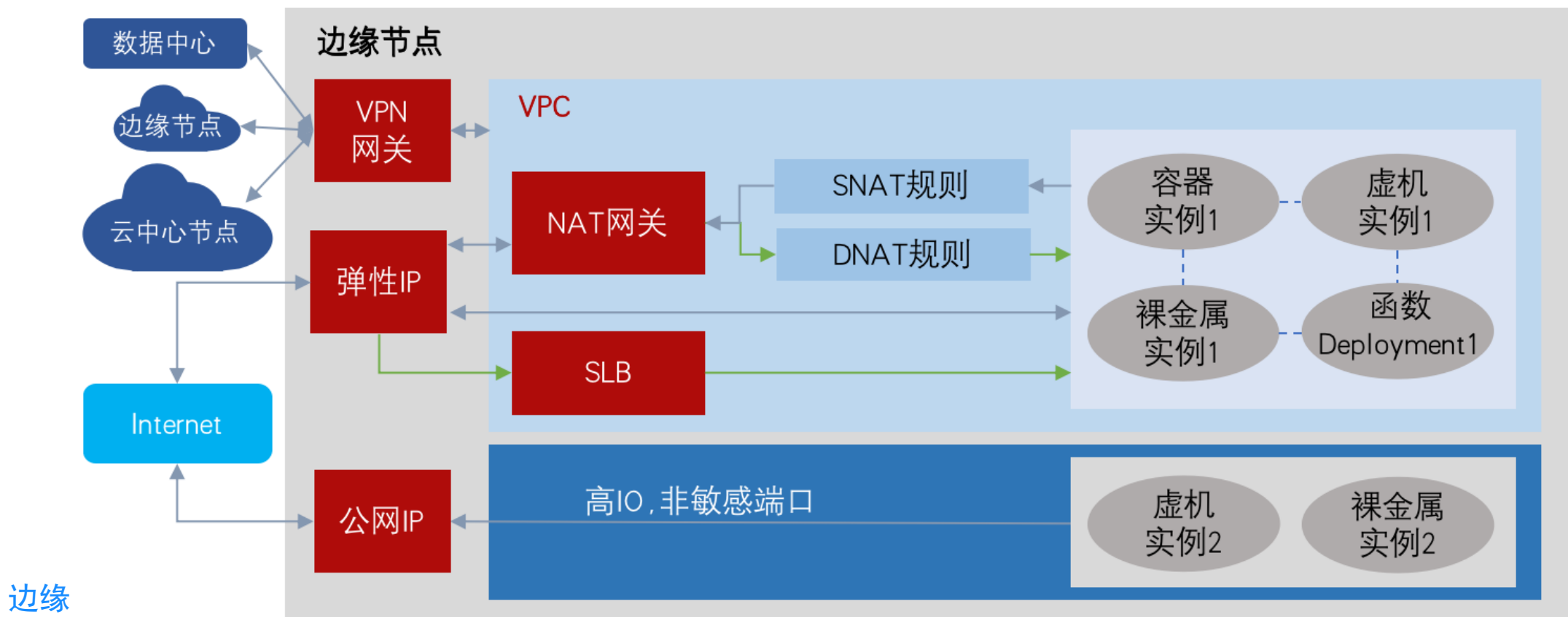


怎么用 Kube-OVN?

电信天翼云边缘计算案例

多租户共享 Kubernetes 集群，每个租户一个子网

容器多网卡对外绑定 EIP，并进行带宽控制，同时实现NAT和LB功能



怎么用 Kube-OVN?

企业版 Kube-OVN

运维支持:

- 由灵雀云安排工程师远程或到现场进行实施、安装或升级等，解决客户问题；
- 服务完成后协助客户验证并确保业务的正常运行，同时出具相关的优化报告及建议；
- 发现 Kube-OVN 有严重安全漏洞时，第一时间通知客户；

技术支持:

- 灵雀云专家对问题进行排查、判断，为客户做出解答；
- 遵循客户要求排查故障，并出具故障报告；
- 对于 Kube-OVN 自身导致的问题，为客户提供优化方案；
- 依据客户要求提供Kube-OVN的培训；

欢迎加入 Kube-OBN 社区!

社区官网: <https://www.kube-ovn.io>

公众号: Kube-OVN



微信交流群: 加V备注 “Kube-OVN”



Open Source AceCon

2021 智能云边开源峰会

AI x Cloud Native x Edge Computing

人工智能 × 云原生 × 边缘计算

Thank You