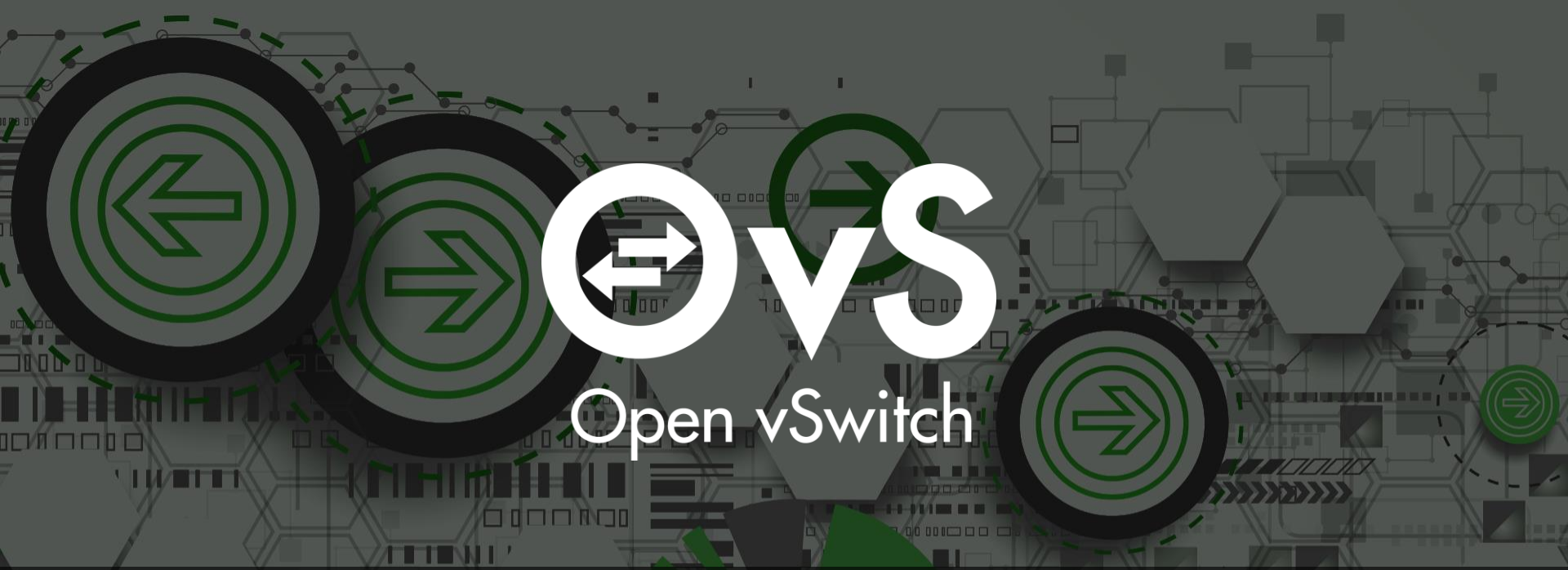


Nodus Network Policy and OVN Balancer

By:
Kuralamudhan Ramakrishnan(kuralamudhan.ramakrishnan@intel.com) | July 19th, 2021

Acknowledgement:
Srinivasa Addepalli, Ritu Sood

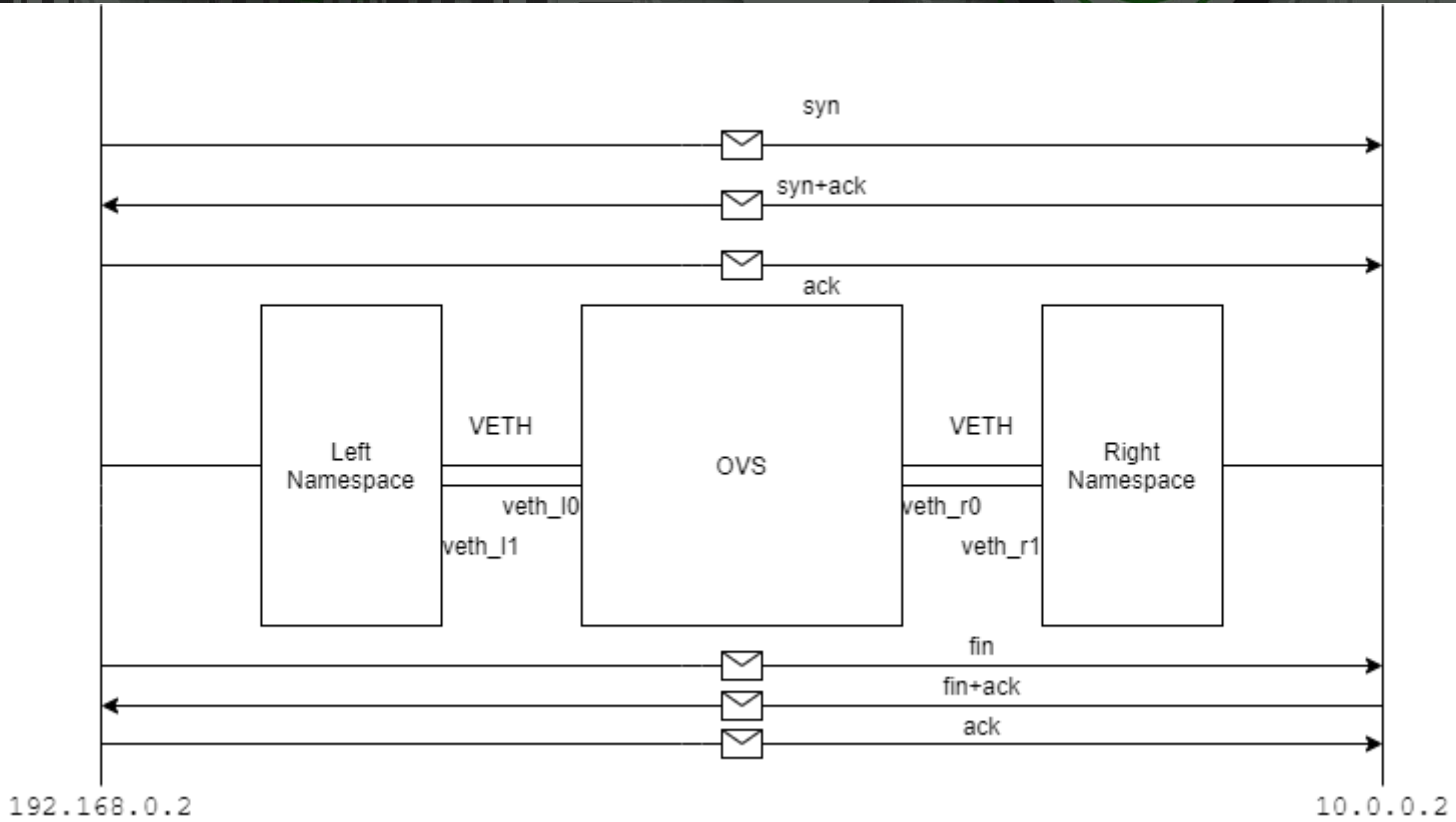
Agenda

- OVS Conntrack Intro
- OVN ACLs
- OVN Load Balancer
- OVN4NFV Network Policy design
- OVN4NFV OVN Balancer

OVS Conntrack Intro

- OVS uses the Connection Tracking system along with OpenFlow flows.
- OpenFlows match on the state of the TCP, UDP connections
- Important details to understand on the Conntrack Fields
    - ct_state:
        - new: new connection or uncommited connection
        - Trk: packet is tracked
        - Est: packet in the established connection

OVS Conntrack Intro



OVS Conntrack Intro

(flow #1)
```
 $ ovs-ofctl add-flow br0 \
   "table=0, priority=50, ct_state=-trk, tcp, in_port=veth_l0, actions=ct(table=0)"
```

(flow #2)
```
$ ovs-ofctl add-flow br0 \
   "table=0, priority=50, ct_state=+trk+new, tcp, in_port=veth_l0, actions=ct(commit),veth_r0"
```

(flow #3)
```
$ ovs-ofctl add-flow br0 \
   "table=0, priority=50, ct_state=-trk, tcp, in_port=veth_r0, actions=ct(table=0)"
```

(flow #4)
```
$ ovs-ofctl add-flow br0 \
   "table=0, priority=50, ct_state=+trk+est, tcp, in_port=veth_r0, actions=veth_l0"
```

(flow #5)
```
$ ovs-ofctl add-flow br0 \
   "table=0, priority=50, ct_state=+trk+est, tcp, in_port=veth_l0, actions=veth_r0"
```

OVS Conntrack Intro

TCP Segment	ct_state(flow#)
Connection Setup	
192.168.0.2 → 10.0.0.2 [SYN] Seq=0	-trk(#1) then +trk+new(#2)
10.0.0.2 → 192.168.0.2 [SYN, ACK] Seq=0 Ack=1	-trk(#3) then +trk+est(#4)
192.168.0.2 → 10.0.0.2 [ACK] Seq=1 Ack=1	-trk(#1) then +trk+est(#5)
Data Transfer	
192.168.0.2 → 10.0.0.2 [ACK] Seq=1 Ack=1	-trk(#1) then +trk+est(#5)
10.0.0.2 → 192.168.0.2 [ACK] Seq=1 Ack=2	-trk(#3) then +trk+est(#4)
Connection Teardown	
192.168.0.2 → 10.0.0.2 [FIN, ACK] Seq=2 Ack=1	-trk(#1) then +trk+est(#5)
10.0.0.2 → 192.168.0.2 [FIN, ACK] Seq=1 Ack=3	-trk(#3) then +trk+est(#4)
192.168.0.2 → 10.0.0.2 [ACK] Seq=3 Ack=2	-trk(#1) then +trk+est(#5)

OVN ACL Intro

- OVN uses the OVS+Conntrack to implement ACLs
- ACLs configured, there are new entries in the logical flow table in the stages switch_in_pre_acl, switch_in_acl, switch_out_pre_acl, and switch_out_acl.
- Let create following rules with OVN ACLs
  - Allow incoming ICMP requests and associated return traffic.
  - Allow incoming SSH connections and associated return traffic.
  - Drop other incoming IP traffic.
- OVN ACLs
- For Example:- Pod with name “acl” has the port default_acl-66d888699-sfs26 created by ovn4nfv

```
ovn-nbctl acl-add ovn4nfvk8s-default-nw 1001 'outport == "default_acl-66d888699-sfs26" && ip && tcp && tcp.dst == 22' allow-related
ovn-nbctl acl-add ovn4nfvk8s-default-nw 1001 'outport == " default_acl-66d888699-sfs26 " && icmp' drop
```

OVN ACL = OVS + OVS Conntrack

```
cookie=0x559d7639, duration=124.833s, table=44, n_packets=3, n_bytes=217, priority=65535,ct_state=-new+est-rel+rpl-inv+trk,ct_label=0/0x1,metadata=0x3 actions=resubmit(,45)
 cookie=0x21de9271, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=65535,icmp6,metadata=0x3,nw_ttl=255,icmp_type=135,icmp_code=0 actions=resubmit(,45)
 cookie=0x21de9271, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=65535,icmp6,metadata=0x3,nw_ttl=255,icmp_type=136,icmp_code=0 actions=resubmit(,45)
 cookie=0x188268dd, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=65535,ct_state=+inv+trk,metadata=0x3 actions=drop
 cookie=0x188268dd, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=65535,ct_state=+est+rpl+trk,ct_label=0x1/0x1,metadata=0x3 actions=drop
 cookie=0x11922202, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=65535,ct_state=-new-est+rel-inv+trk,ct_label=0/0x1,metadata=0x3 actions=resubmit(,45)
 cookie=0x5f20c46e, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-est+trk,icmp6,reg15=0xe,metadata=0x3 actions=drop
 cookie=0x5f20c46e, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-est+trk,icmp,reg15=0xe,metadata=0x3 actions=drop
 cookie=0x5f20c46e, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+est+trk,ct_label=0x1/0x1,icmp6,reg15=0xe,metadata=0x3 actions=drop
 cookie=0xaae1b80d, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+est+trk,ct_label=0/0x1,icmp,reg15=0xe,metadata=0x3 actions=ct(commit,zone=NXM_NX_REG13[0..15],exec(load:0x1->NXM_NX_CT_LABEL[0]))
 cookie=0xaae1b80d, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+est+trk,ct_label=0/0x1,icmp6,reg15=0xe,metadata=0x3 actions=ct(commit,zone=NXM_NX_REG13[0..15],exec(load:0x1->NXM_NX_CT_LABEL[0]))
 cookie=0x5f20c46e, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+est+trk,ct_label=0x1/0x1,icmp,reg15=0xe,metadata=0x3 actions=drop
 cookie=0x1da8000f, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-new+est-rpl+trk,ct_label=0/0x1,tcp,reg15=0xe,metadata=0x3,tp_dst=22 actions=resubmit(,45)
 cookie=0x77182ac8, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-new+est-rpl+trk,ct_label=0x1/0x1,tcp6,reg15=0xe,metadata=0x3,tp_dst=22 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x77182ac8, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-new+est-rpl+trk,ct_label=0x1/0x1,tcp,reg15=0xe,metadata=0x3,tp_dst=22 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x1da8000f, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=-new+est-rpl+trk,ct_label=0/0x1,tcp6,reg15=0xe,metadata=0x3,tp_dst=22 actions=resubmit(,45)
 cookie=0x77182ac8, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+new-est+trk,tcp6,reg15=0xe,metadata=0x3,tp_dst=22 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x77182ac8, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=2001,ct_state=+new-est+trk,tcp,reg15=0xe,metadata=0x3,tp_dst=22 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x2a2728b6, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=1,ct_state=+est+trk,ct_label=0x1/0x1,ipv6,metadata=0x3 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x2a2728b6, duration=124.833s, table=44, n_packets=0, n_bytes=0, priority=1,ct_state=+est+trk,ct_label=0x1/0x1,ip,metadata=0x3 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x2a2728b6, duration=124.832s, table=44, n_packets=0, n_bytes=0, priority=1,ct_state=-est+trk,ipv6,metadata=0x3 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x2a2728b6, duration=124.832s, table=44, n_packets=1, n_bytes=85, priority=1,ct_state=-est+trk,ip,metadata=0x3 actions=load:0x1->NXM_NX_XXREG0[97],resubmit(,45)
 cookie=0x99d903ee, duration=3495826.449s, table=44, n_packets=302148, n_bytes=19381723, priority=0,metadata=0x3 actions=resubmit(,45)
 cookie=0xa03bc2f5, duration=3495826.449s, table=45, n_packets=302152, n_bytes=19382025, priority=0,metadata=0x3 actions=resubmit(,46)
 cookie=0x5c936925, duration=3495826.451s, table=46, n_packets=302152, n_bytes=19382025, priority=0,metadata=0x3 actions=resubmit(,47)
 cookie=0xdf17263b, duration=3495826.451s, table=47, n_packets=0, n_bytes=0, priority=100,ipv6,reg0=0x4/0x4,metadata=0x3 actions=ct(table=48,zone=NXM_NX_REG13[0..15],nat)
 cookie=0xdf17263b, duration=3495826.451s, table=47, n_packets=0, n_bytes=0, priority=100,ip,reg0=0x4/0x4,metadata=0x3 actions=ct(table=48,zone=NXM_NX_REG13[0..15],nat)
 cookie=0x8de71d6, duration=3495826.451s, table=47, n_packets=1, n_bytes=85, priority=100,ip,reg0=0x2/0x2,metadata=0x3 actions=ct(commit,zone=NXM_NX_REG13[0..15],exec(load:0->NXM_NX_CT_LABEL[0])),resubmit(,48)
 cookie=0x8de71d6, duration=3495826.449s, table=47, n_packets=0, n_bytes=0, priority=100,ipv6,reg0=0x2/0x2,metadata=0x3 actions=ct(commit,zone=NXM_NX_REG13[0..15],exec(load:0->NXM_NX_CT_LABEL[0])),resubmit(,48)
 cookie=0x151674a0, duration=3495826.451s, table=47, n_packets=302151, n_bytes=19381940, priority=0,metadata=0x3 actions=resubmit(,48)
 cookie=0xe925063c, duration=3495826.451s, table=48, n_packets=302152, n_bytes=19382025, priority=0,metadata=0x3 actions=resubmit(,49)
 cookie=0x15acb3c8, duration=3495826.449s, table=49, n_packets=2, n_bytes=84, priority=100,metadata=0x3,dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,64)
 cookie=0xa8ea333, duration=3495826.449s, table=49, n_packets=302150, n_bytes=19381941, priority=50,reg15=0x1,metadata=0x3 actions=resubmit(,64)
 cookie=0xd061f215, duration=2533.311s, table=49, n_packets=0, n_bytes=0, priority=50,reg15=0xe,metadata=0x3 actions=resubmit(,64)
 cookie=0x0, duration=3495826.451s, table=64, n_packets=102385, n_bytes=4300170, priority=100,reg10=0x1/0x1,reg15=0x1,metadata=0x3 actions=push:NXM_OF_IN_PORT[],load:0->NXM_OF_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x0, duration=2533.311s, table=64, n_packets=0, n_bytes=0, priority=100,reg10=0x1/0x1,reg15=0xe,metadata=0x3 actions=push:NXM_OF_IN_PORT[],load:0->NXM_OF_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x0, duration=3495892.639s, table=64, n_packets=199767, n_bytes=15081855, priority=0 actions=resubmit(,65)
 cookie=0x0, duration=3495826.452s, table=65, n_packets=302152, n_bytes=19382025, priority=100,reg15=0x1,metadata=0x3 actions=output:"ovn4nfv0-4f26ae"
 cookie=0x0, duration=2533.311s, table=65, n_packets=0, n_bytes=0, priority=100,reg15=0xe,metadata=0x3 actions=output:ced14bdd7844581
```

Network Policy -> OVN ACLS

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
        except:
        - 172.17.1.0/24
    - namespaceSelector:
        matchLabels:
          project: myproject
    - podSelector:
        matchLabels:
          role: frontend
    ports:
    - protocol: TCP
      port: 6379
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
    ports:
    - protocol: TCP
      port: 5978
```

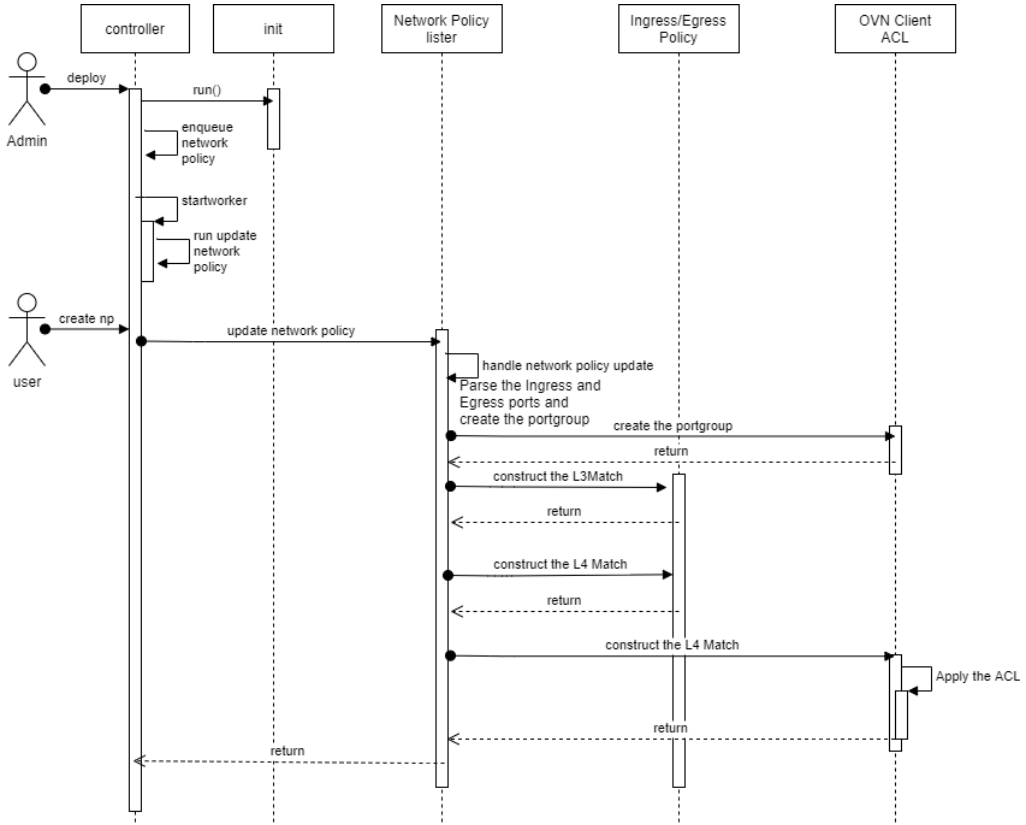
Ingress by OVN ACL

```
ovn-nbctl acl-add to-lport 1001 'outport == portGroupName
priority match= "ip4.src == 172.17.0.0/16 && ip4.src !=
172.17.1.0/24"' allow-related
```

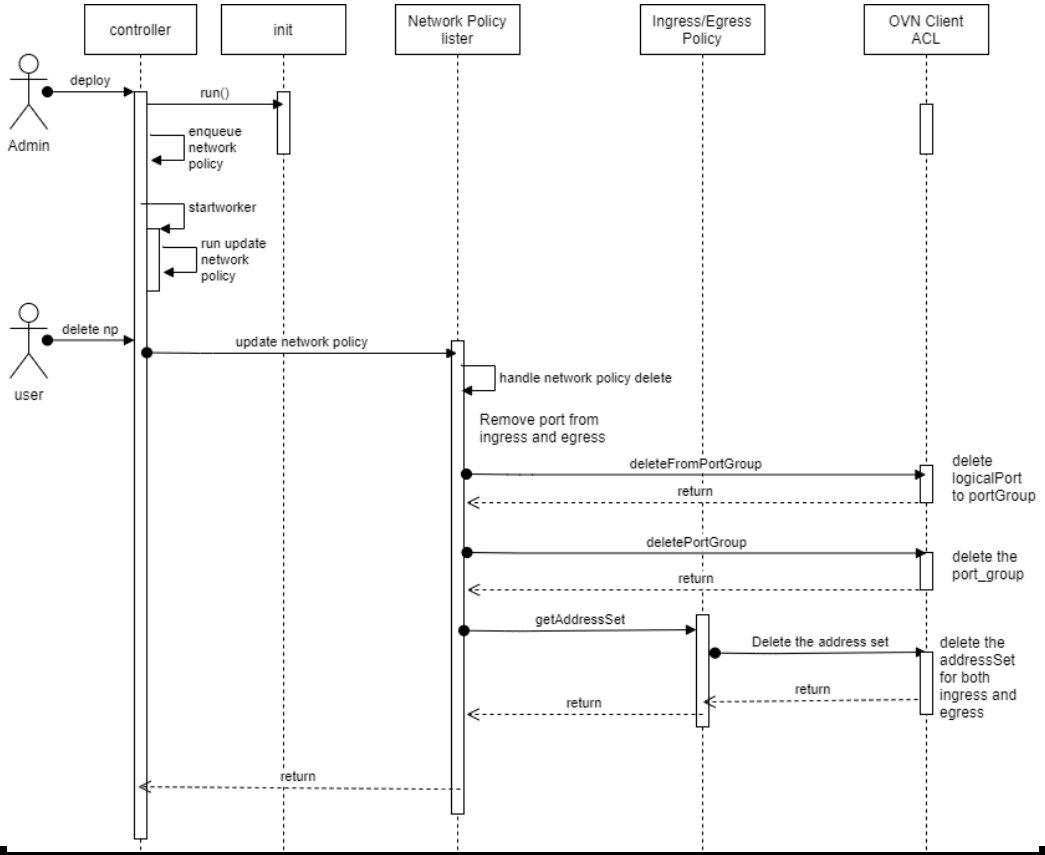
Egress by OVN ACL

```
ovn-nbctl acl-add to-lport 1001 'outport == portGroupName
priority match= "ip4.dst == 10.0.0.0/24"' allow-related
```

OVN ACLs to implement Network policy – Update NPs



OVN ACLs to implement Network policy – delete NPs

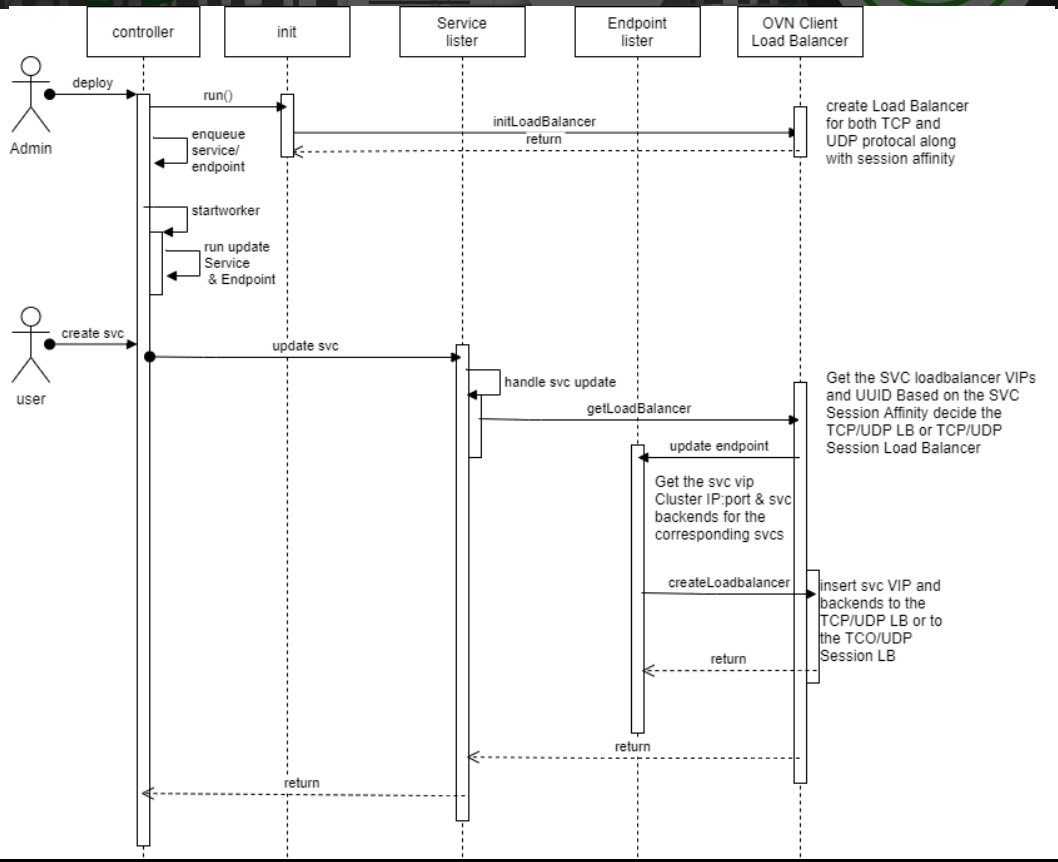


OVN Load Balancer

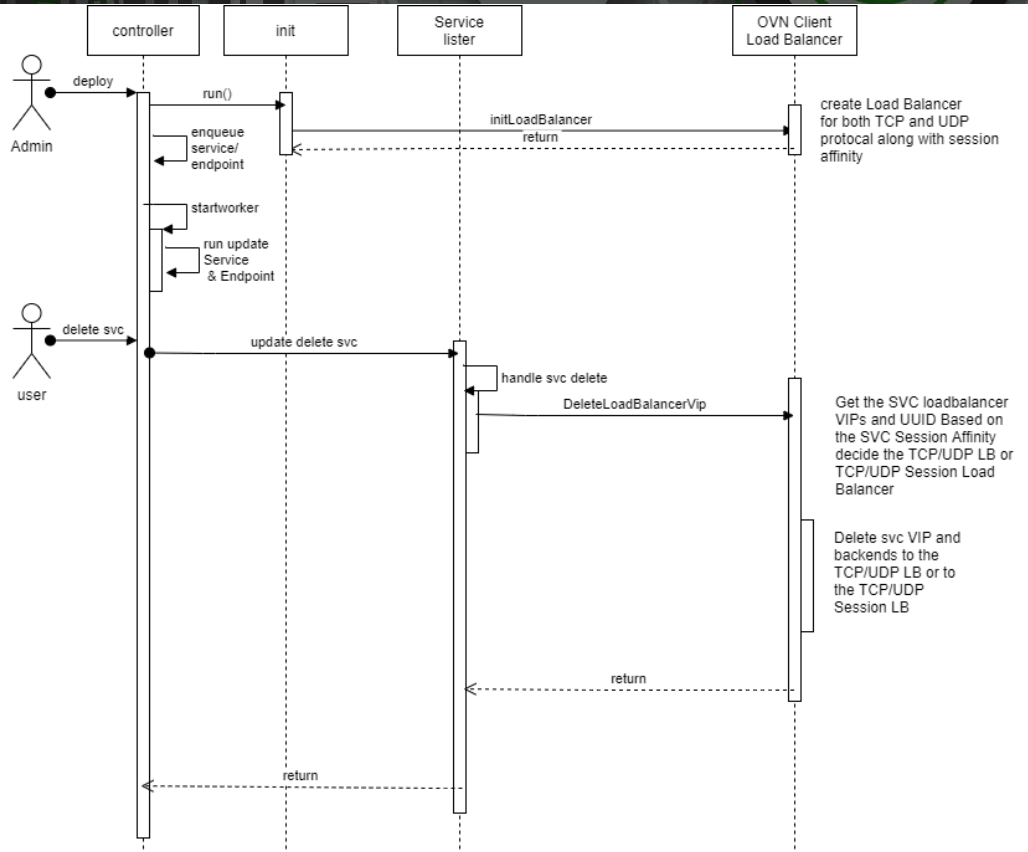
- OVN Load Balancer provides a hash-based load balancing mechanism, which can be used on logical switches or logical routers:
- Used on logical router
  - Can only be used on gateway router
  - Centralized (rather than distributed)
- Used in logical switch
  - Distributed
  - OVN Load Balancer can be used in client logical switch
- Sample here:

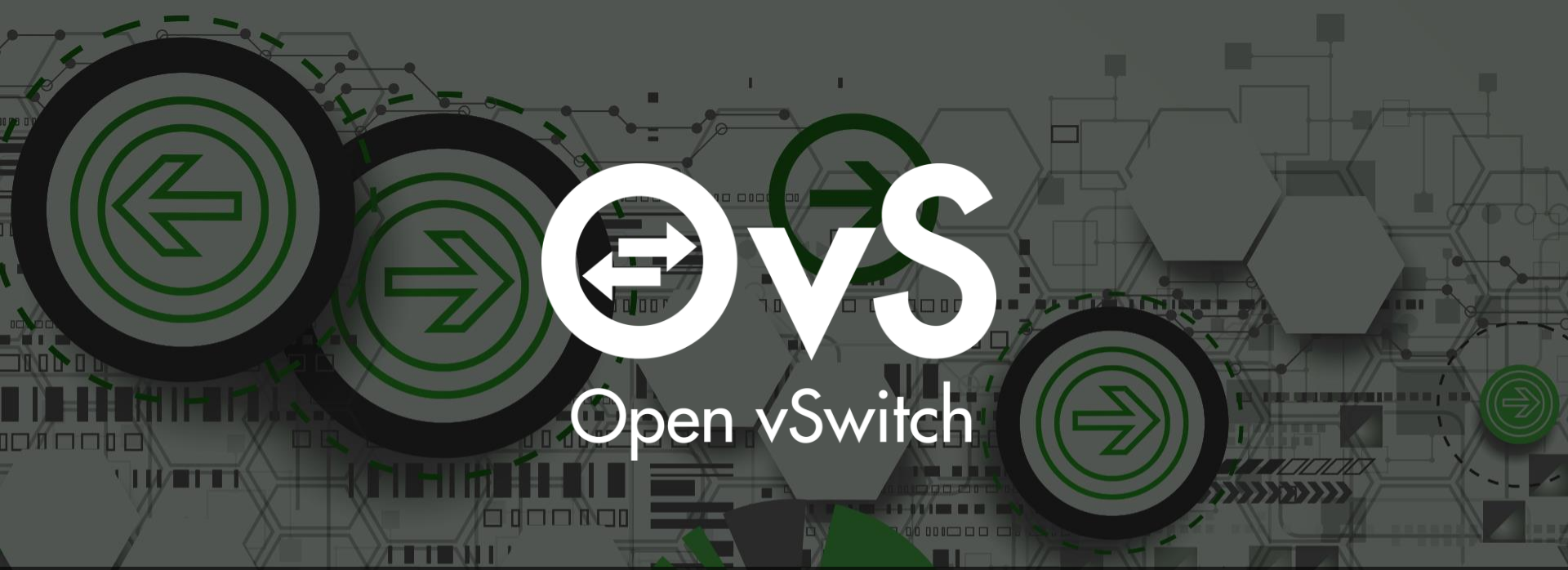
```
# uuid=`ovn-nbctl create load_balancer vips:10.254.10.10="192.168.100.10,192.168.100.11"`
# ovn-nbctl set logical_switch ls2 load_balancer=$uuid
# ovn-nbctl get logical_switch ls2 load_balancer
# ovn-nbctl ls-lb-list ls2
UUID                                    LB          PROTO       VIP             IPs
a19bece1-52bf-4555-89f4-257534c0b9d9                tcp/udp     10.254.10.10    192.168.100.10,192.168.100.11
```

OVN Load Balancer to implement SVC/Endpoint – Update SVC



OVN Load Balancer to implement SVC/Endpoint – delete SVC





Nodus – SFC

By:
Kuralamudhan Ramakrishnan(kuralamudhan.ramakrishnan@intel.com) | July 19th, 2021

Acknowledgement:
Srinivasa Addepalli, Ritu Sood

Nodus deployment Model