Policy-based Routing in OVN

Mary Manohar

Sragdhara D Chaudhuri

Nutanix

Outline

- What is Policy-based routing?
- Implementation in OVN
- Service-chaining
- Enhancements

What is Policy-based routing?

Traditional IP routing is destination-based.

Ability to deny, permit, reroute traffic to a different endpoint based on

- IP Source/Destination address
- IP Protocol type
- L4 Source/Destination ports
- Incoming interface (subnet) on the router

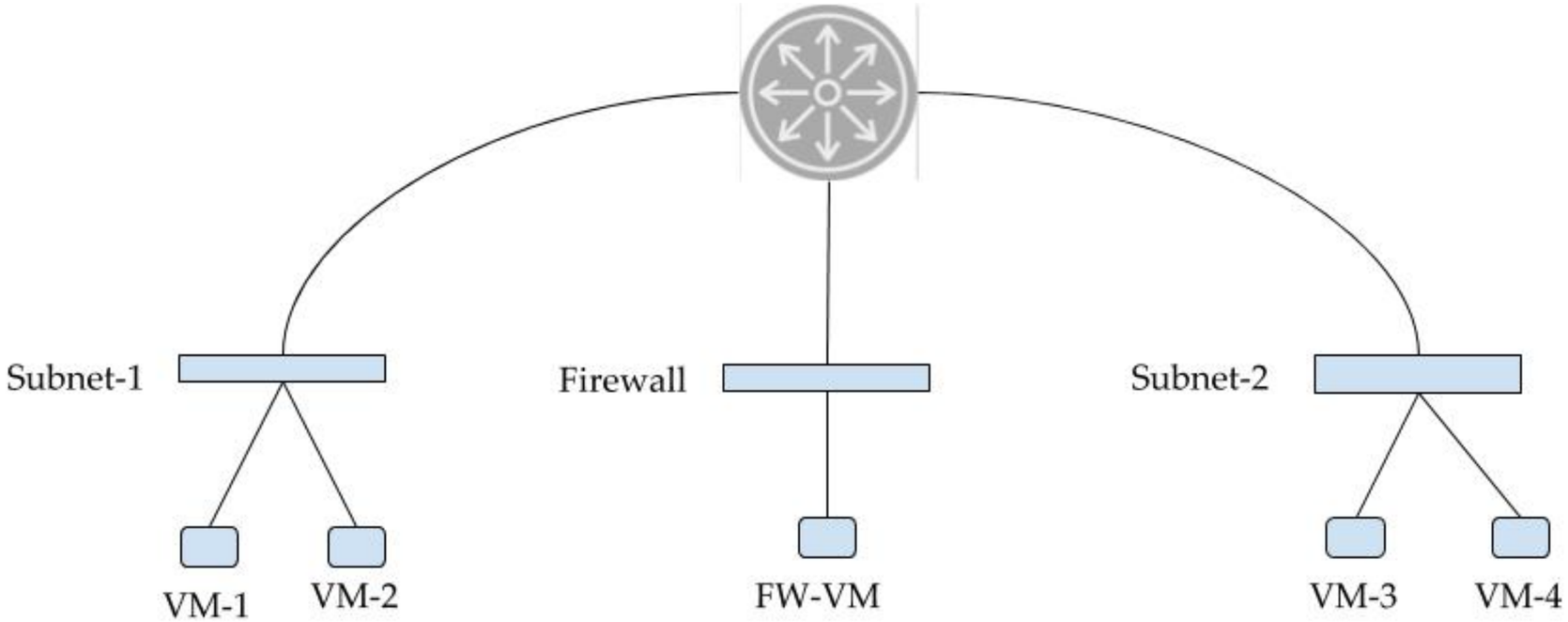
Every policy has a priority value associated with it.

Policy with the highest priority wins.

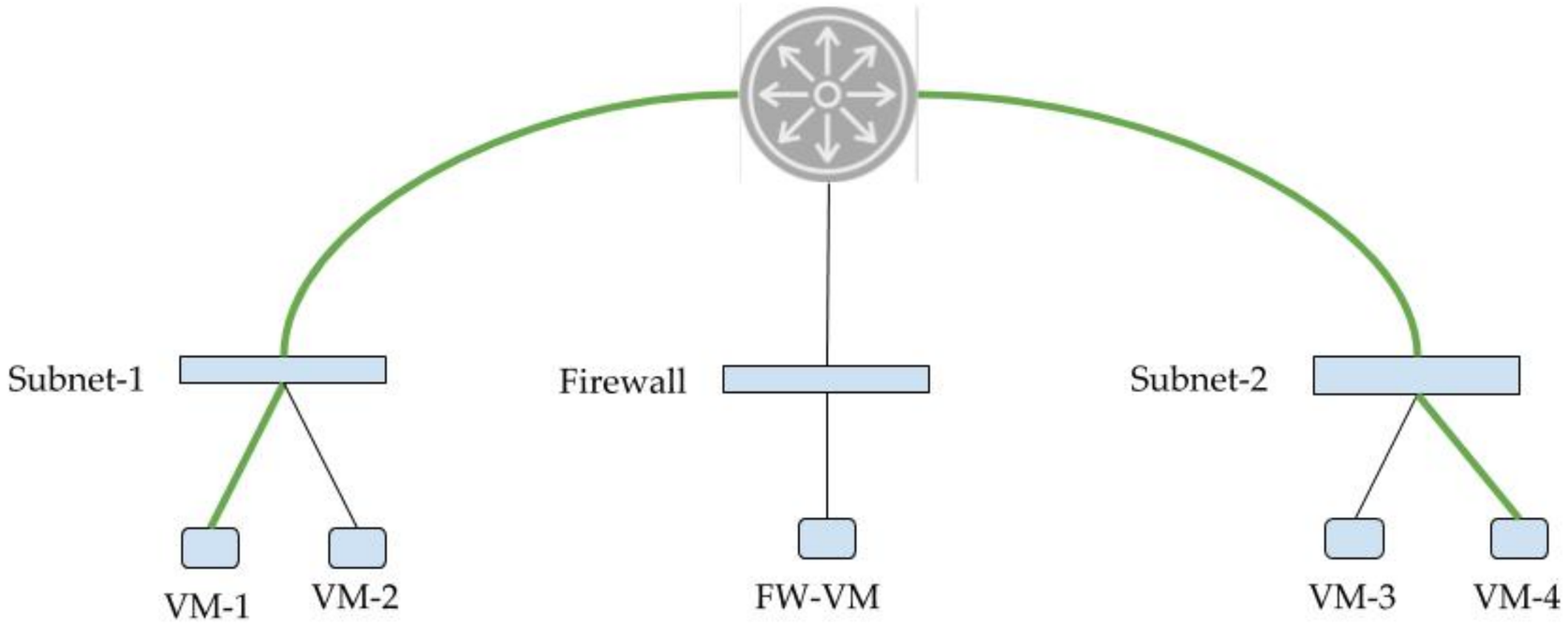
Why Policy-based Routing?

- Service-insertion on the router.
  - Override routing decision to reroute certain types of traffic to services like firewall or VPN.
- Permit/deny rules on router.

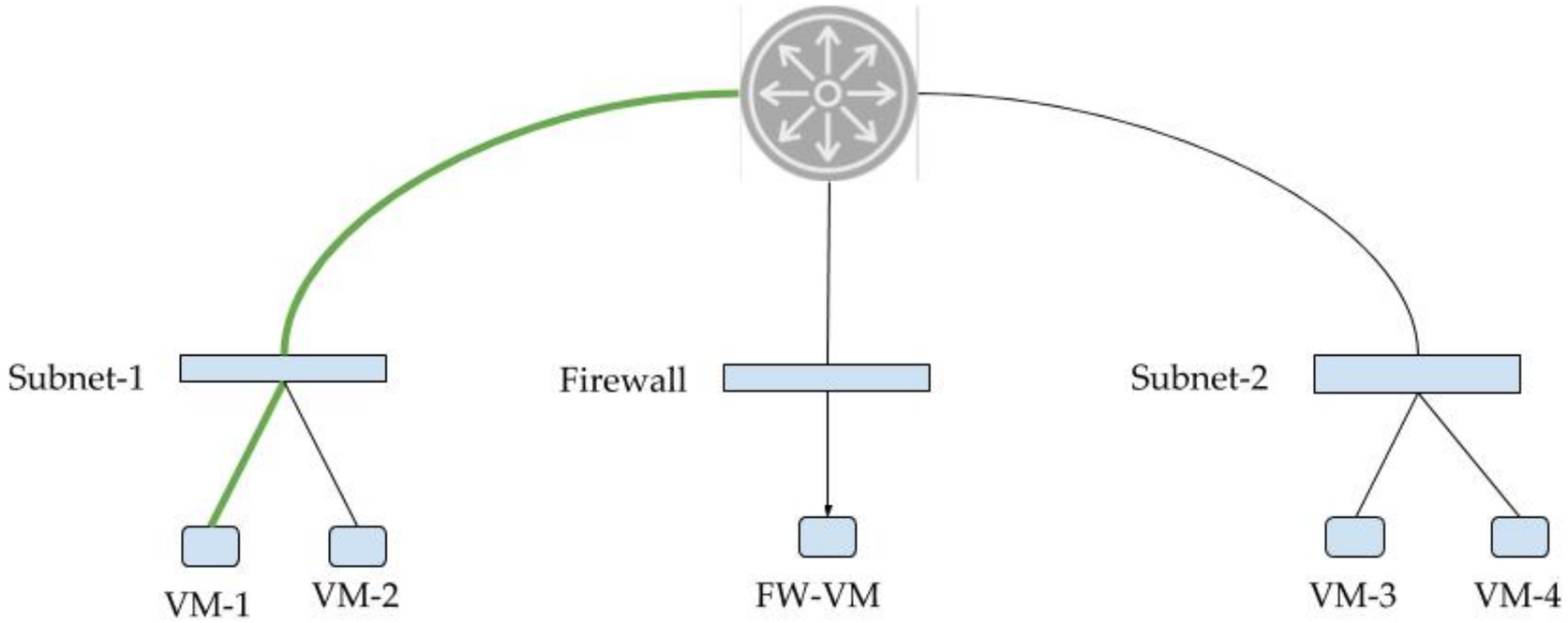
Example network



Destination-based routing

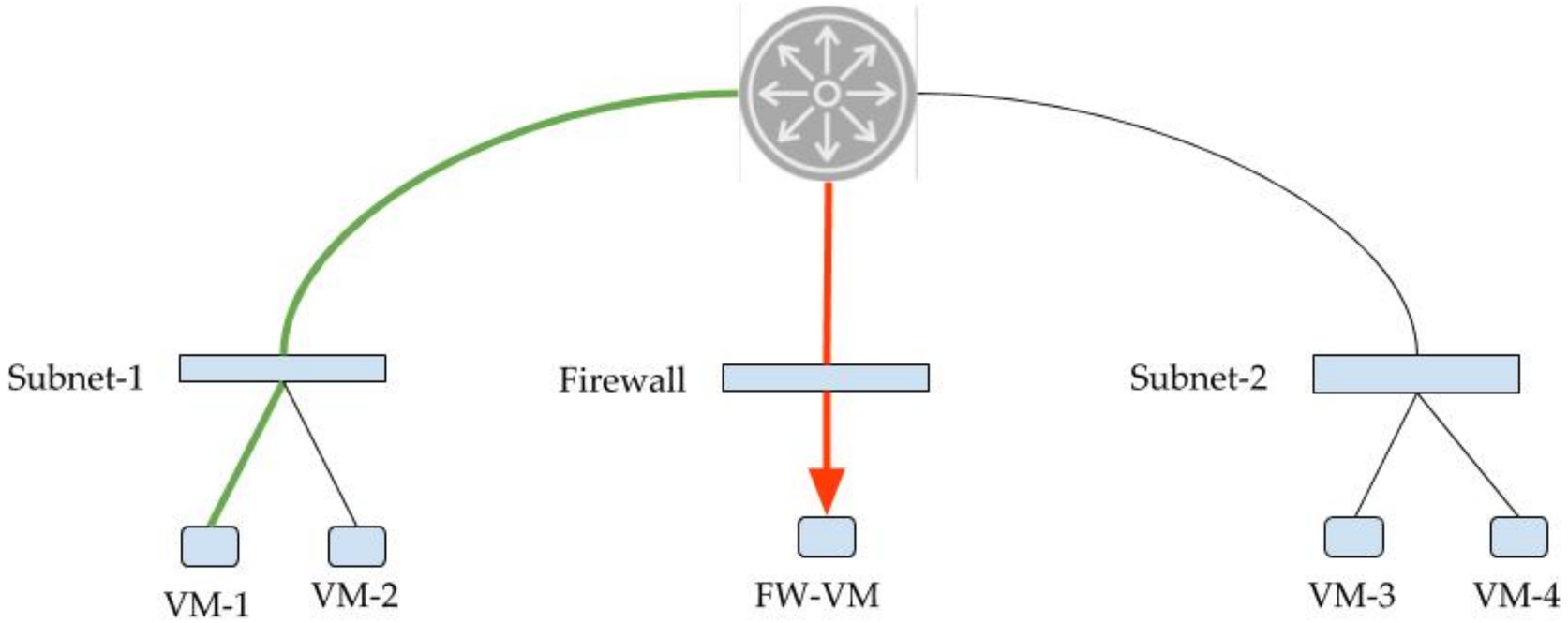


Policy-based Routing - L3 service insertion



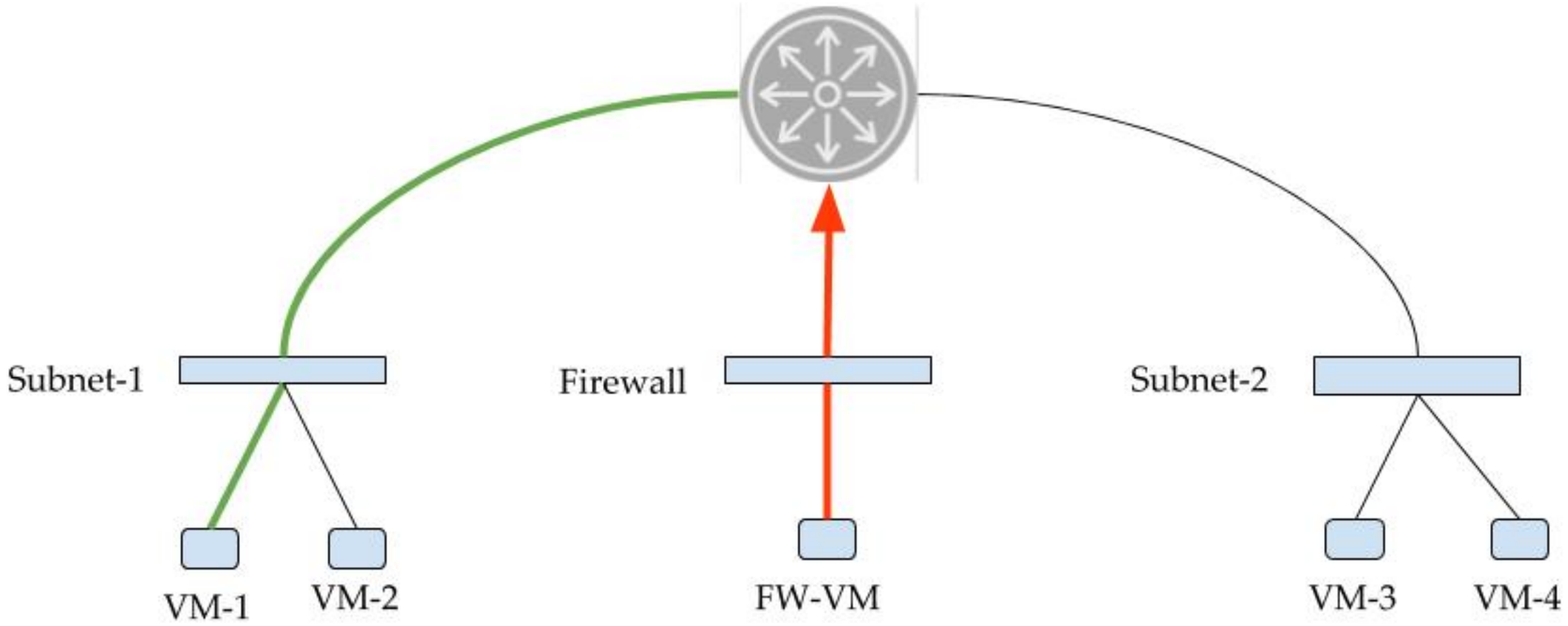
priority: 10 **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** FW-VM

Packet redirected to FW-VM



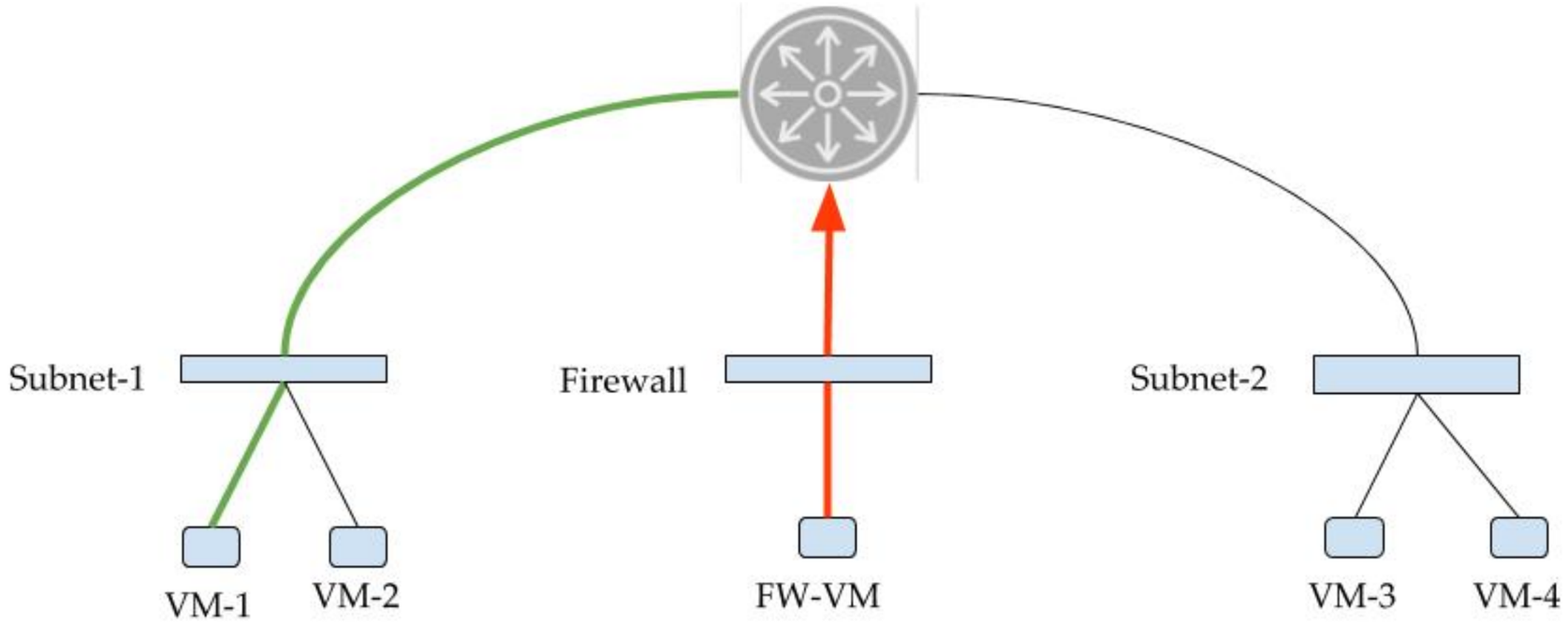
priority: 10 **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** FW-VM

Packet coming back from firewall



priority: 10 **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** FW-VM

Packet coming back from firewall



priority: 20 **inport:** Firewall-interface **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **permit —> Higher priority**

priority: 10 **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** FW-VM

Implementation in OVN

Destination-based routing in OVN router pipeline

IT0:	L2 Admission Control
IT1:	IP Input
IT2:	Defrag
IT3:	UNSNAT
IT4:	DNAT
IT5:	IP Routing
IT6:	ARP/ND Resolution
IT7:	Gateway Redirect
IT8:	ARP Request

ET0:	UNDNAT
ET1:	SNAT
ET2:	Egress Loopback
ET3:	Delivery

Policy-based routing implementation

Standard OVN Router Pipeline

IT0:	L2 Admission Control
IT1:	IP Input
IT2:	Defrag
IT3:	UNSNAT
IT4:	DNAT
IT5:	IP Routing
IT6:	ARP/ND Resolution
IT7:	Gateway Redirect
IT8:	ARP Request

ET0:	UNDNAT
ET1:	SNAT
ET2:	Egress Loopback
ET3:	Delivery

Modified OVN Router Pipeline

IT0:	L2 Admission Control
IT1:	IP Input
IT2:	Defrag
IT3:	UNSNAT
IT4:	DNAT
IT5:	IP Routing
IT6:	Policy-Based Routing
IT7:	ARP/ND Resolution
IT8:	Gateway Redirect
IT9:	ARP Request

ET0:	UNDNAT
ET1:	SNAT
ET2:	Egress Loopback
ET3:	Delivery

Policy-based overrides destination-based routing

IT0:	L2 Admission Control
IT1:	IP Input
IT2:	Defrag
IT3:	UNSNAT
IT4:	DNAT
IT5:	IP Routing
IT6:	Policy-Based Routing
IT7:	ARP/ND Resolution
IT8:	Gateway Redirect
IT9:	ARP Request

ET0:	UNDNAT
ET1:	SNAT
ET2:	Egress Loopback
ET3:	Delivery

Implications: Packets coming to Floating IPs

IT0:	L2 Admission Control
IT1:	IP Input
IT2:	Defrag
IT3:	UNSNAT
IT4:	DNAT
IT5:	IP Routing
IT6:	Policy-Based Routing
IT7:	ARP/ND Resolution
IT8:	Gateway Redirect
IT9:	ARP Request

← SNAT IPs are changed to private IPs here

← Floating IPs are changed to private IPs here

← Floating IPs / SNAT IPs are not visible here

ET0:	UNDNAT
ET1:	SNAT
ET2:	Egress Loopback
ET3:	Delivery

New pipeline stage

IP-Routing table carries destination-based routes.

New table overrides the routing decision based on policies.

In the example:

- Traffic **to** the firewall was forwarded based on policies.
- Return traffic **from** firewall was forwarded based on IP-Routing table.

Ovn-nbctl commands

Add a policy

A policy is uniquely identified by <priority, match-string>

ovn-nbctl lr-policy-add **ROUTER** **PRIORITY** **MATCH** **ACTION** **[NEXTHOP]**

Action: Permit/drop/reroute

Example:

ovn-nbctl lr-policy-add lr1 10 "ip4.src == 1.1.1.0/24" drop

ovn-nbctl lr-policy-add lr1 10 "ip4.src == 2.2.2.0/24” drop

Ovn-nbctl commands

Delete a policy:

ovn-nbctl lr-policy-del **ROUTER [PRIORITY [MATCH]]**

Priority and match string are ***optional*** parameters.

<ROUTER, PRIORITY, MATCH>: the exact policy is deleted.

<ROUTER, PRIORITY>: All policies with given priority are deleted.

<ROUTER>: ALL policies under ROUTER are deleted.

Ovn-nbctl commands

List policies:

ovn-nbctl lr-policy-list **ROUTER**

```
611     ip4.dst==12.2.1.0/24 && ip4.src==11.2.1.0/24 && inport=="lrp-1"     allow
610     ip4.dst==12.2.1.0/24 && ip4.src==11.2.1.0/24                        reroute         13.2.1.12
600     ip4.dst==0.0.0.0/0 && ip4.src==0.0.0.0/0                            drop
```

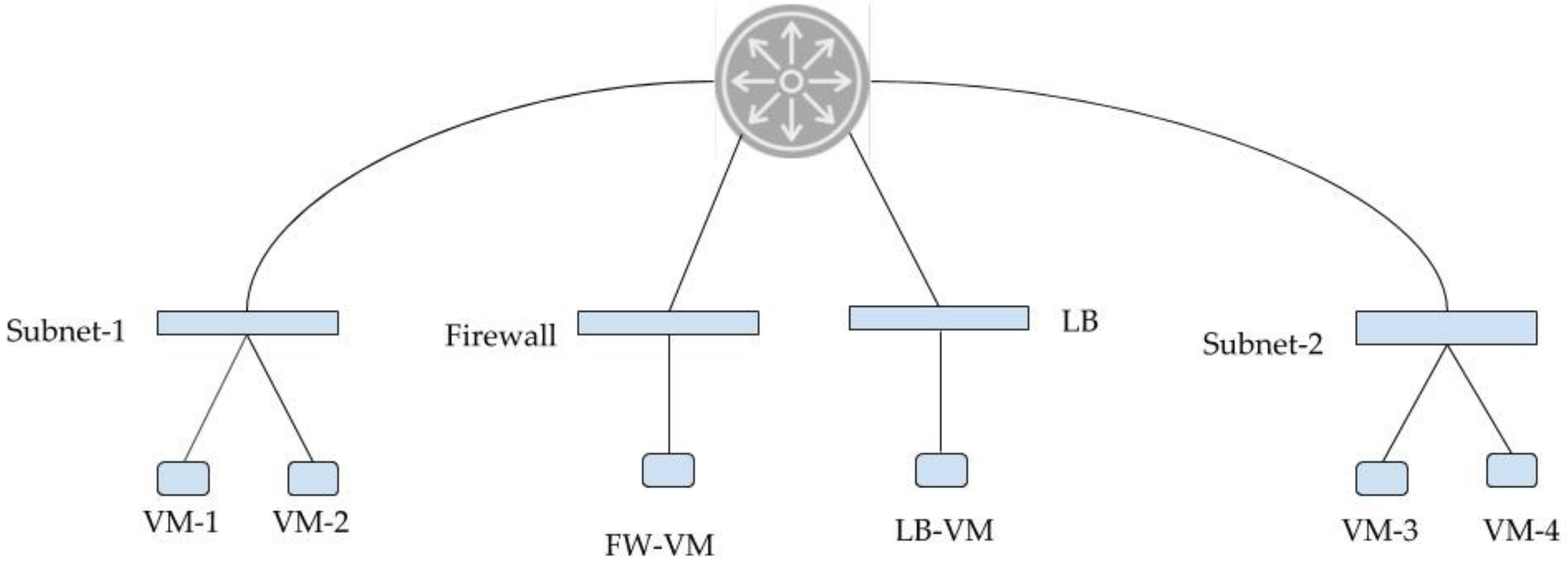
TTL Handling

The new pipeline stage will not decrement the TTL.

TTL is decremented in the IP-Routing stage.

Service Chaining

Service-chaining

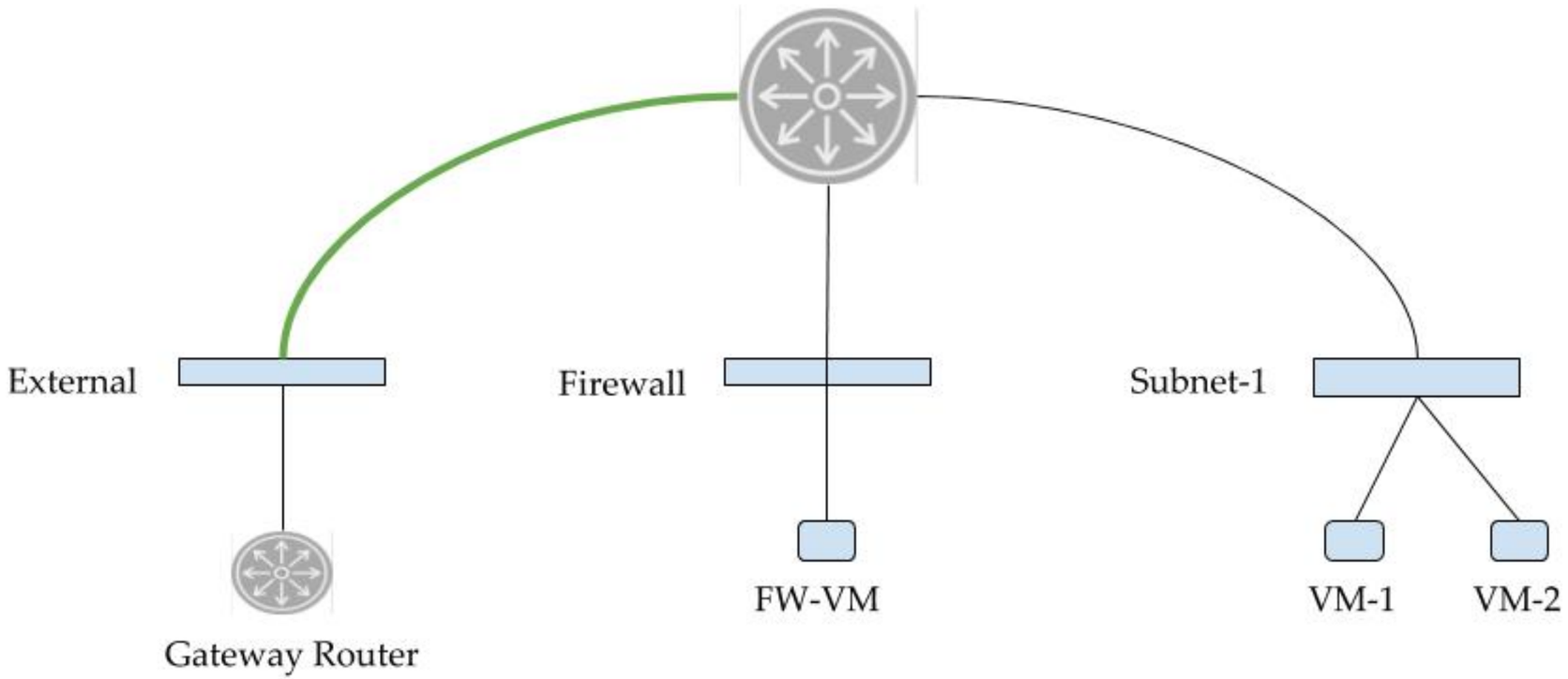


priority: 30 **inport:** LB-interface **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **permit**

priority: 20 **inport:** Firewall-interface **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** LB-VM

priority: 10 **src-ip:** Subnet-1 **dst-ip:** Subnet-2 **reroute-to:** FW-VM

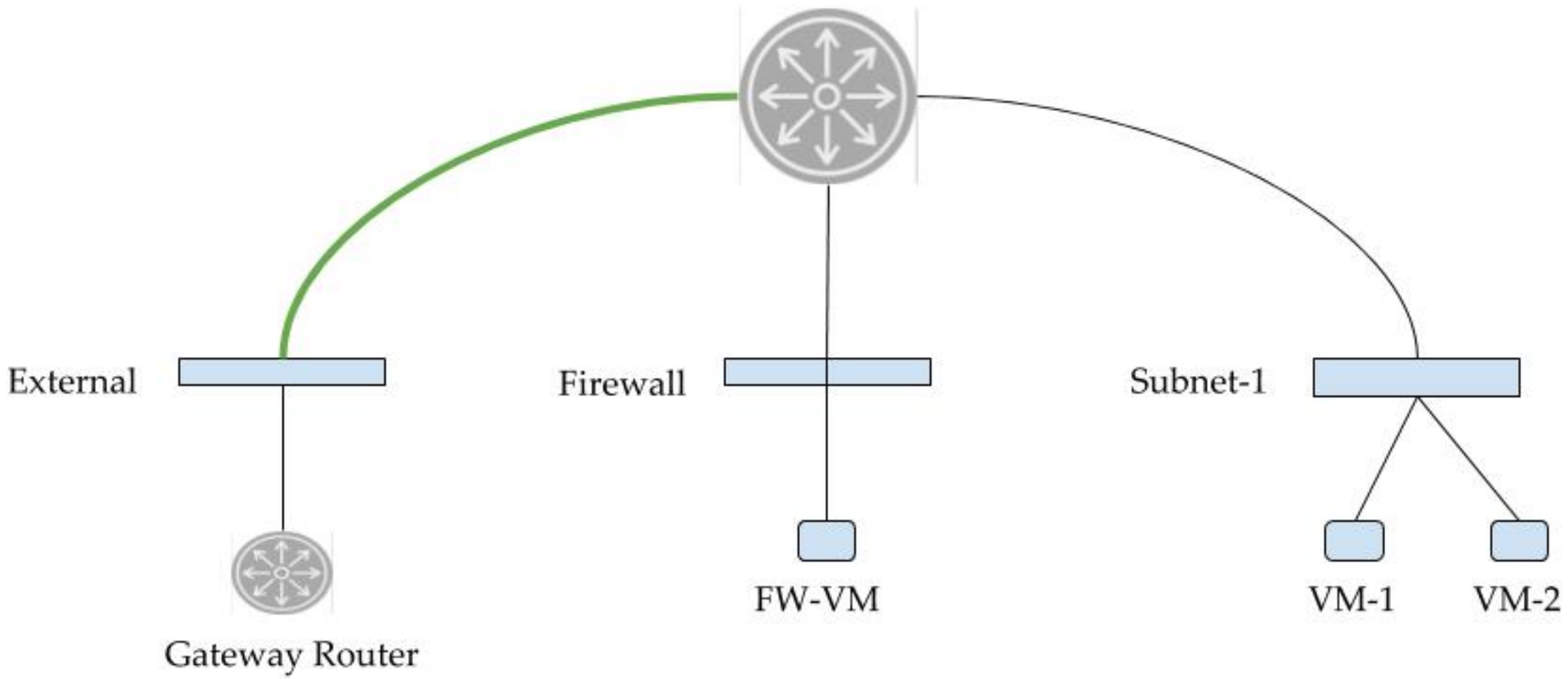
L3 service insertion - special case



src-ip: ?? **dst-ip:** Subnet-1 **reroute-to:** FW-VM

Source-IP: All IPs that don’t belong to any subnets attached to logical-router.

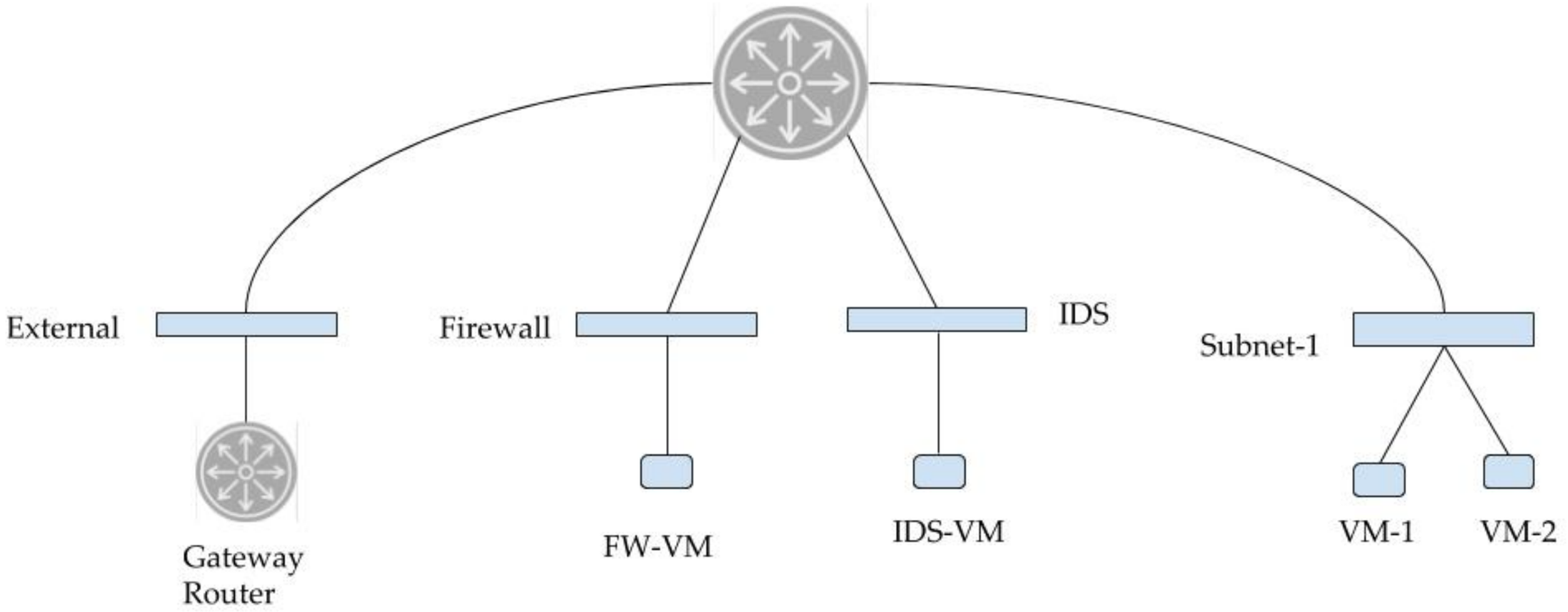
L3 service insertion - special case



priority: 20 **inport:** Firewall-interface **dst-ip:** Subnet-1 **permit**

priority: 10 **inport:** External-interface **dst-ip:** Subnet-1 **reroute-to:** FW-VM

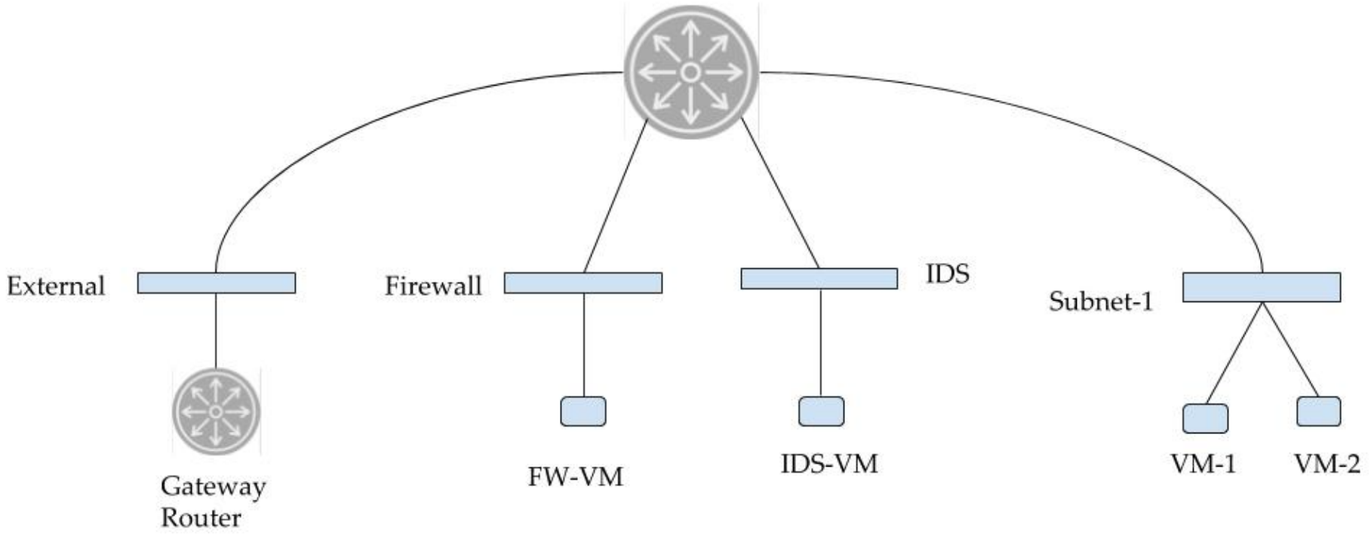
Service chaining - special case



All traffic to Subnet-1 goes through Firewall.

Traffic coming from External goes through Firewall and IDS.

Service chaining - special case



All traffic to Subnet-1 goes through Firewall.

Traffic coming from External goes to Firewall and IDS.

priority: 20 **inport:** Firewall-interface src-ip: ?? **dst-ip:** Subnet-1 **reroute-to:** IDS-VM <— How to identify traffic coming from ‘External’?

priority: 10 **src-ip:** 0.0.0.0/0 **dst-ip:** Subnet-1 **reroute-to:** FW-VM

Need for address-sets support

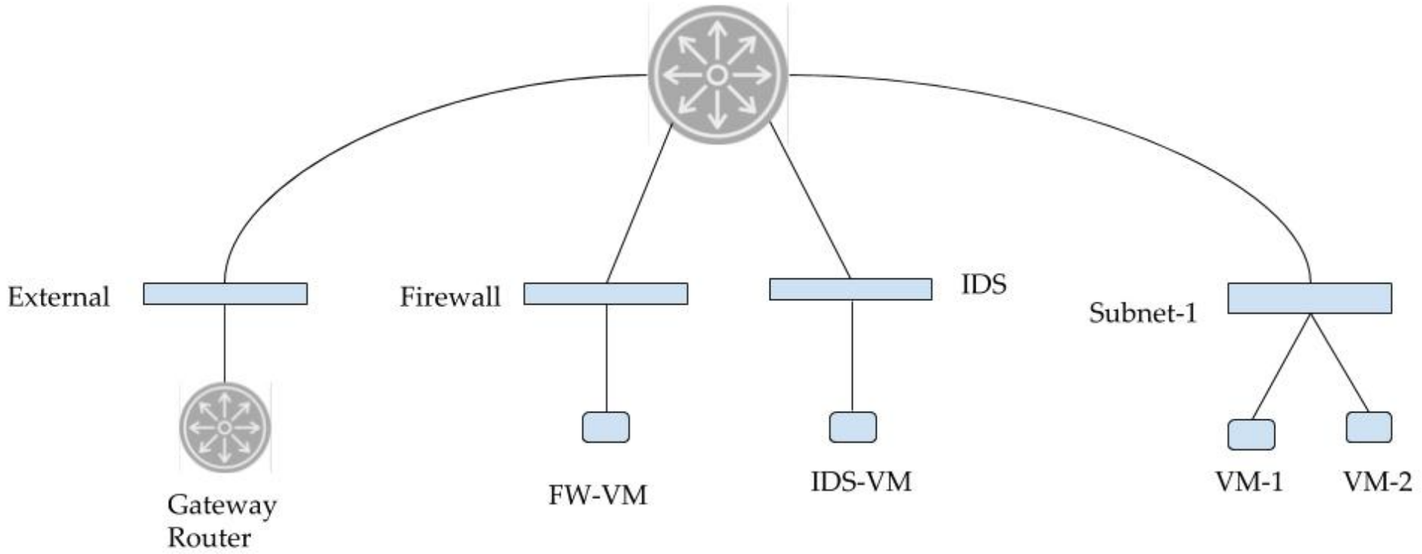
‘External’ - All IPs that don’t belong to any subnets attached to logical-router.

Need a way of marking subnets attached to logical router.

ovn-nbctl create Address_Set name=set1 addresses={internal-ips} —> Already supported.

ovn-nbctl create Address_Set name=set1 addresses={internal-prefixes} —> Could be an enhancement

Final set of policies



priority: 40 **inport:** IDS-interface permit

priority: 30 **inport:** Firewall-interface src-ip: $set1 **dst-ip:** Subnet-1 permit **<—- Local traffic to Subnet-1**

priority: 20 **inport:** Firewall-interface **src-ip:** 0.0.0.0/0 **dst-ip:** Subnet-1 **reroute-to: IDS-VM <— External traffic**

priority: 10 **src-ip:** 0.0.0.0/0 **dst-ip:** Subnet-1 **reroute-to:** FW-VM

Enhancements

Enhancements

- Logging
- Stateful policies
- ECMP for next-hop - Example: Pool of firewall VMs
- Address-sets: in match string for ip4.src and ip4.dst
- In addition to setting the next-hop, set fields in the packet like DSCP: For Gateway router to classify traffic and treat with different QoS.

Thank you