OpenContrail

Functionality

- Neutron v2 compatible API.
- High-performance.
- Distributed Router (forwarding, ACLs, NAT, DHCP, Proxy ARP, metadata proxy, etc…).
- Neutron networks implemented as standard compliant L3VPN networks.
- Broadcast / multicast support.
- Network policy.
- Service insertion.

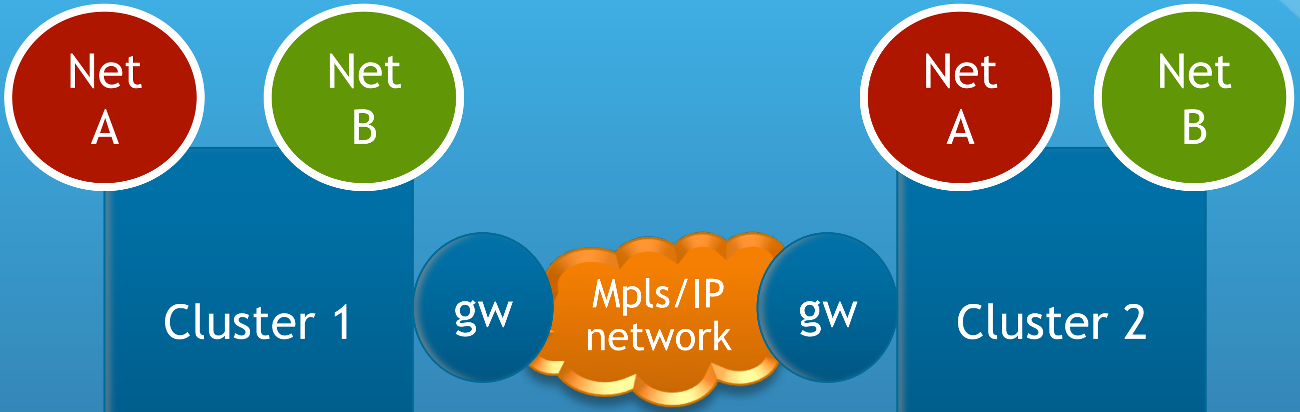
Distributed router

- Each virtual-machine interface is associated with a private routing table (VRF).
- This VRF contains the routes for all networks that the VM has reachability to.
- Distributed services: ARP, DHCP, ACLs.
- ACL enforcement in the ingress server.
- Traffic flows directly between ingress and egress server, reducing latency and fabric utilization.

Extending neutron networks

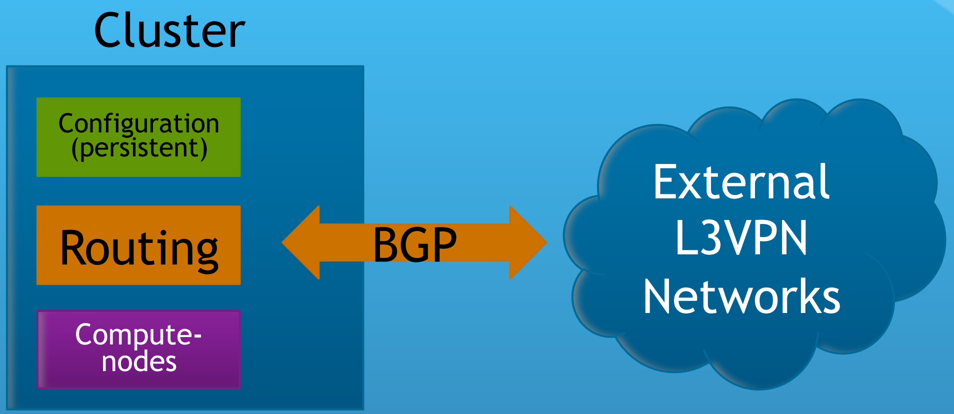
- With OpenContrail a neutron network can be extended beyond the cluster as an L3VPN:
  - To a VLAN;
  - Across a WAN to another L3VPN;
  - To another OpenStack cluster (potentially across a WAN);
- This is applicable to a range of scenarios:
  - Bare-metal services on a VLAN can be brought into the cluster as a neutron network.
  - Neutron networks can exist across different OpenStack clusters.
  - The neutron network context can be preserved across the WAN.

Network interoperability



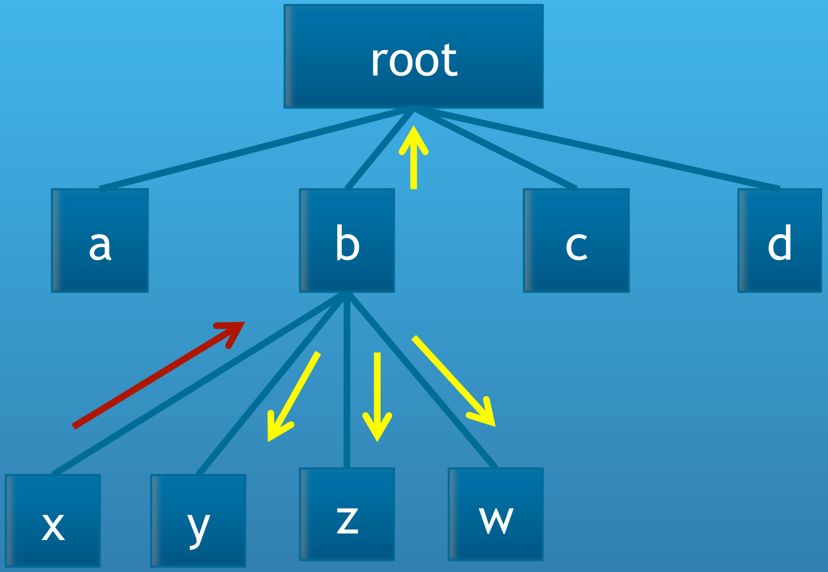
Network isolation is maintained outside the cluster via BGP/MPLS L3VPN.

L3VPN model



- Contrail control-plane uses a peering model to interoperate with other L3VPN capable devices (e.g. routers or other clusters);
- BGP peering allows for interconnection across different administrative authorities.
- Existing routers can interconnect an L3VPN network other network technologies (e.g. VLAN).

Scalable multicast



- Control plane calculates a distribution tree with the hypervisors that have VMs for a particular multicast group.
- Bi-directional tree: packet can be injected in the middle of the tree.

Network policy

- Cloud infrastructure provides common services to multiple tenants.
  - E.g. databases, logging, caches, management and monitoring.
- Service owners must be able to control the access to the service.
- Current neutron model (router) is inverted; doesn’t allow for multiple services to define their access policies.

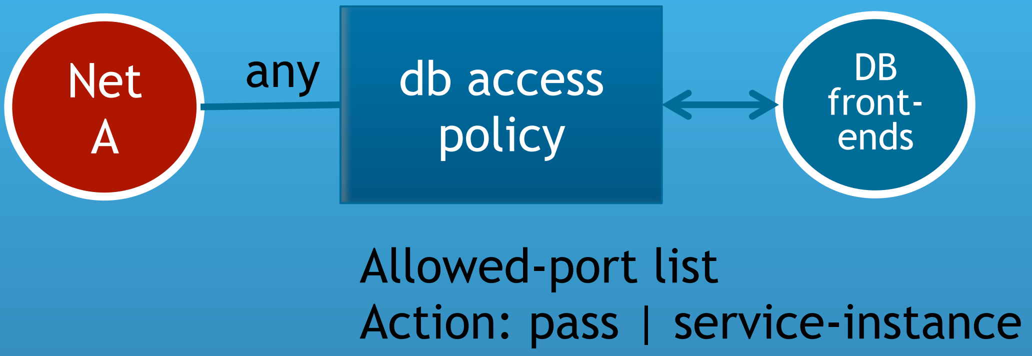
Service model

- Service manager defines a list of traffic access rules that apply to clients of the service; this is done as a network-policy. Policy can specify the client or accept wildcard.
- Clients can select multiple policies that apply: providing connectivity and access control rules.

Service insertion

- Policy rule can be accept/deny or:
- Apply a specific service: firewall, IPS, NAT, DDoS mitigation, etc.
- Service can be scaled horizontally.
- Contrail automatically adjusts the routing.
- Works for edge as well as transit networks (using dynamic routing).

Example



- DB service owner can insert a firewall on-the-fly whenever need arises.
- Network-policy injects routing information (modified if service instance is selected) and ACL rules applied by the ingress hypervisor switch.