OVN Project Update

Russell Bryant (@russellbryant)
Kyle Mestery (@mestery)
Justin Pettit (@Justin_D_Pettit)
Ben Pfaff (@Ben_Pfaff)
The Case for Network Virtualization

• Network provisioning needs to be self-service.
• Virtual network needs to be abstracted from physical.
• Virtual network needs same features as physical.
Why Start OVN?

• Virtual networking for OVS, done from scratch using experience built up in the OVS project
• Natural evolution of the project--OVS had only provided the components to build a distributed switch
• Cloud management systems agnostic
Why Should OpenStack Care?

• Neutron’s primary job is to provide a cloud networking API abstraction
• OVN is a scope increase of OVS to implement many of the things Neutron needs
• If OVN succeeds in its mission, it reduces development burden on Neutron for OVS integration significantly
What is OVN?

• Open source L2/L3 network virtualization for Open vSwitch (OVS):
  ✓ Logical switches
  ✓ L2/L3/L4 ACLs (Security Groups)
  ✓ Logical routers
  ✓ Multiple tunnel overlays (Geneve, STT, and VXLAN)
  ✓ TOR-based and software-based logical-physical gateways

• Works on same platforms as OVS:
  ✓ Linux (KVM and Xen)
  ✓ Containers (Docker)
  ✓ DPDK
    – Hyper-V

• Integration with:
  ✓ OpenStack Neutron
    – Other CMSes

The “Toaster Oven” Release
The Particulars

• Developed by the same community as Open vSwitch
• Vendor-neutral
• Design and implementation all occur in public
• Developed under the Apache license
Goals

• Production-quality
• Straightforward design
• Scale to 1000s of hypervisors (each with many VMs/containers)
• Improved performance and stability over existing OpenStack OVS plugin
How is OVN Different?
OVN Architecture

OVN Architecture Diagram:

- **OpenStack/CMS Plugin**
- **Northbound DB**
- **Southbound DB**
- **ovn-northd**

**HV-1**:
- **ovn-controller**
  - **ovsdb-server**
  - **ovs-vswitchd**

**HV-n**:
- **ovn-controller**
  - **ovsdb-server**
  - **ovs-vswitchd**
Architecture

- Configuration coordinated through databases
- Logical flows, don’t worry about physical topology
- Local controller converts logical flow state into physical flow state
- Desired state clearly separated from run-time state
- Based on the architecture we wanted based on seeing a number of others using OVS
Neutron Plugin

- Speaks OVSDB to configure OVN via its Northbound database
- Goal: only run neutron API server, no agents
- No RabbitMQ, except for notifications (for Ceilometer, or a custom listener)
Security Groups (The Existing Way)

- Required extra Linux bridge and veth pair per VM
- Uses iptables
Security Groups (OVN ACLs)

- Uses kernel conntrack module directly from OVS
- Design benefits
  - No complicated pipeline
  - Faster* -- Fewer hops and veth ports

* http://blog.russellbryant.net/2015/10/22/openstack-security-groups-using-ovn-acls/
L3 (The Existing Way)

- Agent based
- Used the Linux IP stack and iptables
  - Forwarding
  - NAT
- Overlapping IP address support using namespaces
Current L3 Diagram
OVN L3 Design

• Native support for IPv4 and IPv6
• Distributed
• ARP/ND suppression
• Flow caching improves performance
  • Without OVN: multiple per-packet routing layers
  • With OVN: cache sets dest mac, decrements TTL
• No use of Neutron L3 agent
Gateways

• Based on “vtep” OVSDB schema included with OVS
  • Hardware: Arista, Brocade, Cumulus, Dell, HP, Juniper, Lenovo
  • OVN integration demo in the Brocade booth
  • Software: Implement “vtep” schema in software, via DPDK
    • Will become a reference for building OVS DPDK applications
• Later: move beyond the capabilities of the “vtep” schema to support fail-over, scale-out, and more stateful services
Physical Workload Integration

- OVN Cluster
- HV1
- PH1
- PH2
- VM1
- VM2
- VM3

API (OVSDB)
Tunnels (VXLAN)

VTEP (Top of Rack Switch)

Logical Network
Rolling Upgrades

- OVSDB schema is versioned
- Changes to schema will be carefully managed to be backwards compatible
- Allows rolling upgrades
  - Update databases first
  - Roll through upgrades to ovn-controller
- Same strategy OVS itself has been using
Status
Neutron Integration Status

• Neutron plugin supports
  – L2 networks
  – Provider Networks
  – Security Groups
  – binding:profile for containers in VMs without another overlay
  – binding:profile for connecting vtep gateways to Neutron networks

• Uses Neutron L3 agent by default today, OVN L3 integration in progress

• Will use OVN DHCP instead of DHCP agent when ready

• New Neutron APIs of interest
  – VLAN-aware VMs (more generally, parent/child ports)
  – networking-l2gw, networking-sfc
Upcoming Work in OVN

- NAT for OVS (Patches available)
- Native DHCP support (Patches available)
- Service Function Chaining (In progress)
- Basic load balancing (required for Kubernetes)
Resources

• Architecture described in detail in ovn-architecture (5)
• Available in the “master” branch of the main OVS repo:
  – https://github.com/openvswitch/ovs
  – http://openvswitch.org/support/dist-docs/
• Neutron plugin in its own repo:
  – https://git.openstack.org/openstack/networking-ovn.git
• Neutron integration docs, including devstack instructions:
  – http://docs.openstack.org/developer/networking-ovn/
How you can help

• Try it! Test it! Write Code!
• Report bugs and try it at scale
• Core OVN is being developed on ovs-dev mailing list:
  – http://openvswitch.org/pipermail/dev/
  – #openvswitch on Freenode
• Neutron plugin for OVN is being developed here:
  – http://git.openstack.org/openstack/networking-ovn.git
  – openstack-dev mailing list
  – #openstack-neutron-ovn on Freenode
Watch out during the 9am Wednesday morning keynote for an OVN appearance ...
Thank you!

Russell Bryant (@russellbryant)
Kyle Mestery (@mestery)
Justin Pettit (@Justin_D_Pettit)
Ben Pfaff (@Ben_Pfaff)