Openstack Quantum:
Virtual Networks for OpenStack

Dan Wendlandt – dan@nicira.com
Outline

What is Quantum?
Why do we want Quantum?
Architecture Basics
Current project status
Roadmap?
What is Quantum?

A standalone Openstack service
Provides network connectivity between a set of network “interfaces” from other service (e.g., vNICs from compute service, interfaces on a load-balancer service).
Exposes API of logical abstractions for describing network connectivity + policy between interfaces.
Uses a “plug-in” architecture, so multiple technologies can implement the logical abstractions.
Provides a “building block” for sophisticated cloud network topologies.
What is Quantum: Overview

Nova: virtual server
Quantum: virtual network

Both:
- Expose a logical API for automated provisioning by cloud tenants.
- Manage physical devices in the data center to implement the logical model.
- Provide a “plugin” architecture to leverage support using different back-end technologies.
Why Quantum?

1) **Rich Topologies**: Cloud tenants deploying enterprise workloads often require flexibility to define sophisticated network topologies.

2) Cloud operators want to leverage **Advanced Technology** in their network infrastructure (i.e., more than L2 bridging + VLANs).

3) Integrate Advanced Services (LBaaS, FWaaS, etc.)
Why Quantum?: Rich Topologies

Existing Nova-only Network Provisioning Model:
Cloud operator statically determines a type of connectivity (flat, private) that will be statically applied to all VMs.

Tenant VMs get connectivity based on static model.
Why Quantum?: Rich Topologies

New Nova + Quantum Network Provisioning Model:
Tenant can dynamically create networks via API:

    create-network dmz-net, web-net, db-net

When creating VMs, tenant can specify the # vNICs and the networks they connect to:

    create-server image=fw-img nics=dmz-net,web-net
    create-server image=web-img nics=web-net,db-net
    create-server image=db-img nics=db-net

Try it yourself:
http://wiki.openstack.org/QuantumOVSDemo
Advanced Tech: QoS to meet SLAs

Cloud operators may want to be able to guarantee the a tenant VM or portion of the physical NIC capacity, just like they provide a portion of CPU + Memory.

Such properties can be exposed as additional port attributes using Quantum API extensions.

Cisco Quantum plugin enables leveraging properties of UCS hardware to provide QoS guarantees.
Advanced Tech: Avoid VLAN Limits

Many limitations when using VLANs for network isolation:
  Limited # of VLANs supported, limited to single physical subnet,
  requires configuring physical switches, etc….

Open vSwitch Quantum plugin can support L2-in-L3 tunneled
  networks without needing to use VLANs.

At summit, Cisco, Red Hat, Citrix & Nicira discussed plans to add
  support for VXLAN tunneling proposal.
Insert Additional Services

Other services want to be on the same networks, share the same address spaces with compute nodes.

FWaaS

LBaaS

VPNaaS
What is Quantum: Scope

Current focus is on defining Layer-2 Ethernet topologies.

Already (partially) integrated with Nova.

Higher level network services can be added:
  as new APIs within Quantum (e.g., Layer-3 routing)
  as stand-alone services (e.g., Atlas Load-balancing service)
Virtual Network Abstractions (1)

Services (e.g., nova, atlas) expose **interface-IDs** via their own tenant APIs to represent any device from that service that can be “plugged” into a virtual network.
   Example: nova.foo.com/<tenant-id>/server/<server-id>/eth0

Tenants use Quantum API to create **networks**, get back UUID:
    Example: quantum.foo.com/<tenant-id>/network/<network-id>

Tenants can create **ports** on a network, get a UUID, and associate config with those ports (APIs for advanced port config are TBD, initially ports give L2 connectivity):
    Example: quantum.foo.com/<tenant-id>/network/<network-id>/port/<port-id>

Tenants can “plug” an interface into a port by setting the **attachment** of a port to be the appropriate interface-id.
What is Quantum: Example Scenario:

Physical Server #1
- Nova i-23: 10.0.0.23
- Nova i-26: 10.0.0.26

Private Net #1

Data Center Network

Private Net #2

Physical Server #2
- Nova i-26: 10.0.0.26
- Nova i-24: 10.0.0.24

Tenant View

Provider View

Nova i-22: 10.0.0.2
Nova i-24: 10.0.0.24
What is Quantum: Example Architecture

Dashboard / Automation Tools

Tenant API

Quantum Service

Quantum API

Quantum Plugin

Nova Service

Nova-scheduler

 nova-api

Internal nova Communication

Internal Plugin Communication

Hypervisor

vswitch

Two Plugins Available:
- Open vSwitch
- Cisco UCS/Nexus
### Compute: Instances

Instances are virtual servers launched from images.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Image</th>
<th>Private IP</th>
<th>Public IP</th>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>instance1</td>
<td>10.0.0.3</td>
<td>N/A</td>
<td>Build</td>
<td>Terminate, Reboot, Log, VNC Console</td>
<td></td>
</tr>
</tbody>
</table>
Compute: Networks

Networks are virtual quantum networks that provide L2 connectivity to your instances.

Create Network

Network Name

Create Network

Networks

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Ports</th>
<th>Available</th>
<th>Used</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd6a6195-e646-4a51-9bb8-1aa74105b57d</td>
<td>test network</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>Delete, Rename</td>
</tr>
</tbody>
</table>
### Create Ports

**Number of Ports**

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Attachment</th>
<th>Actions</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>b2bb8996-677c-415b-9f83-4ff8365eg920</td>
<td>ACTIVE</td>
<td>None</td>
<td>Attach</td>
<td>QoS</td>
</tr>
<tr>
<td>ae6f79b6-bac7-49bf-8670-26b16bb2533</td>
<td>DOWN</td>
<td>None</td>
<td>Attach, Delete, Port UP</td>
<td>QoS</td>
</tr>
<tr>
<td>11a61d72-02e1-4ab5-957b-875b712e57c</td>
<td>DOWN</td>
<td>None</td>
<td>Attach, Delete, Port UP</td>
<td>QoS</td>
</tr>
<tr>
<td>29964ed5-3579-4e15-8821-a40c0623f74</td>
<td>DOWN</td>
<td>None</td>
<td>Attach, Delete, Port UP</td>
<td>QoS</td>
</tr>
</tbody>
</table>
Project Status: Diablo Release

Started at Diablo summit
6 months of dev resulted in Diablo release:
  v1.0 of the Quantum L2 API
  API extensibility framework
  API client library and CLI
  Nova Integration via the QuantumManager
  OpenStack Dashboard Integration
  Plugin framework & two publicly available plugins:
    Open vSwitch Plugin
    Cisco UCS/Nexus Plugin

Available at: http://launchpad.net/quantum
Docs at: http://docs.openstack.org/incubation/
Project Status: Who should use Quantum?

“Early adopters” are already putting Quantum into their OpenStack trial deployments. **Caution:** deployments are by people who understand the limitations of Quantum and are looking to improve the platform by running/testing it.

Some aspects of the traditional Nova networking model are not yet integrated with Quantum (e.g., floating IPs, DHCP). This will improve significantly over the Essex time frame, watch for updates on the OpenStack list + IRC meetings.
Quantum Essex Roadmap

Top Priority: production quality + usability:
 Automated functional, integration & scale testing
 Parity/integration with all existing Nova network capabilities.
 Dashboard + client enhancements for ease of use.
 API auth with Keystone (mostly done)
 Packaging & distribution

Essex is target for production quality. Some will deploy pre-production using Essex milestones.

Strategic Directions:
 Implementing L3 service for “virtual routers”
 Advanced RBAC for API
 VXLAN/NVGRE, and much, much more…
Questions / Comments?

Thanks!

Come join us:

http://wiki.openstack.org/Quantum
netstack@lists.launchpad.net
Openstack Quantum:
Virtual Networks for OpenStack

Dan Wendlandt – dan@nicira.com
Outline

What is Quantum?
Why do we want Quantum?
Architecture Basics
Current project status
Roadmap?
What is Quantum?

A standalone Openstack service
Provides network connectivity between a set of network “interfaces” from other service (e.g., vNICs from compute service, interfaces on a load-balancer service).
Exposes API of logical abstractions for describing network connectivity + policy between interfaces.
Uses a “plug-in” architecture, so multiple technologies can implement the logical abstractions.
Provides a “building block” for sophisticated cloud network topologies.
Why Quantum?

1) **Rich Topologies**: Cloud tenants deploying enterprise workloads often require flexibility to define sophisticated network topologies.

2) Cloud operators want to leverage Advanced Technology in their network infrastructure (i.e., more than L2 bridging + VLANs).

3) Integrate Advanced Services (LBaaS, FWaaS, etc.)
Why Quantum?: Rich Topologies

Existing Nova-only Network Provisioning Model:
Cloud operator statically determines a type of connectivity (flat, private) that will be statically applied to all VMs.

Tenant VMs get connectivity based on static model.
Why Quantum?: Rich Topologies

New Nova + Quantum Network Provisioning Model:
Tenant can dynamically create networks via API:

create-network dmz-net, web-net, db-net

When creating VMs, tenant can specify the # vNICs and the networks they connect to:

create-server image=fw-img nics=dmz-net,web-net
create-server image=web-img nics=web-net,db-net
create-server image=db-img nics=db-net

Try it yourself:
http://wiki.openstack.org/QuantumOVSDemo
Advanced Tech: QoS to meet SLAs

Cloud operators may want to be able to guarantee the a tenant VM or portion of the physical NIC capacity, just like they provide a portion of CPU + Memory.

Such properties can be exposed as additional port attributes using Quantum API extensions.

Cisco Quantum plugin enables leveraging properties of UCS hardware to provide QoS guarantees.
Advanced Tech: Avoid VLAN Limits

Many limitations when using VLANs for network isolation:
  Limited # of VLANs supported, limited to single physical subnet,
  requires configuring physical switches, etc....

Open vSwitch Quantum plugin can support L2-in-L3 tunneled networks without needing to use VLANs.

At summit, Cisco, Red Hat, Citrix & Nicira discussed plans to add support for VXLAN tunneling proposal.
Insert Additional Services

Other services want to be on the same networks, share the same address spaces with compute nodes.

FWaaS
LBaaS
VPNaaS
What is Quantum: Scope

Current focus is on defining Layer-2 Ethernet topologies.

Already (partially) integrated with Nova.

Higher level network services can be added:
  as new APIs within Quantum (e.g., Layer-3 routing)
  as stand-alone services (e.g., Atlas Load-balancing service)
Virtual Network Abstractions (1)

Services (e.g., nova, atlas) expose **interface-IDs** via their own tenant APIs to represent any device from that service that can be “plugged” into a virtual network.
Example: nova.foo.com/<tenant-id>/server/<server-id>/eth0

Tenants use Quantum API to create **networks**, get back UUID:
Example: quantum.foo.com/<tenant-id>/network/<network-id>

Tenants can create **ports** on a network, get a UUID, and associate config with those ports (APIs for advanced port config are TBD, initially ports give L2 connectivity):
Example: quantum.foo.com/<tenant-id>/network/<network-id>/port/<port-id>

Tenants can “plug” an interface into a port by setting the **attachment** of a port to be the appropriate interface-id.
What is Quantum: Example Scenario:

Physical Server #1
Nova i-26
10.0.0.26

Data Center Network

Private Net #1
Nova i-23
10.0.0.23

Private Net #2
Nova i-24
10.0.0.24

Physical Server #2
Nova i-26
10.0.0.26
Nova i-24
10.0.0.24

Tenant View

Provider View

Physical Server #1
Nova i-26
10.0.0.26
Nova i-24
10.0.0.24

Data Center Network
## Compute: Instances
Instances are virtual servers launched from images.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Image</th>
<th>Private IP</th>
<th>Public IP</th>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>instances1</td>
<td>10.0.0.3</td>
<td>N/A</td>
<td>Build</td>
<td></td>
<td>• Terminate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Reboot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Log</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• VNC Console</td>
</tr>
</tbody>
</table>
OpenStack Dashboard - Compute: Networks

Create Network

Network Name

Create Network

Networks

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Ports</th>
<th>Available</th>
<th>Used</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd5a6195-6546-4a51-9bb8-1aa74103557d</td>
<td>test network</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
### Create Ports

**Number of Ports:**

- **Create Ports**

### Ports

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Attachment</th>
<th>Actions</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960656-8203-470b-446f-701d-4f58-3f5e-345857624870</td>
<td>ACTIVE</td>
<td>None</td>
<td>Attach</td>
<td>QoS</td>
</tr>
<tr>
<td>1960656-8203-470b-446f-701d-4f58-3f5e-345857624870</td>
<td>DOWN</td>
<td>None</td>
<td>Detach</td>
<td>QoS</td>
</tr>
<tr>
<td>1960656-8203-470b-446f-701d-4f58-3f5e-345857624870</td>
<td>DOWN</td>
<td>None</td>
<td>Detach</td>
<td>PortUP</td>
</tr>
<tr>
<td>1960656-8203-470b-446f-701d-4f58-3f5e-345857624870</td>
<td>DOWN</td>
<td>None</td>
<td>Detach</td>
<td>PortUP</td>
</tr>
</tbody>
</table>

- **Attach port to a virtual interface**
Project Status: Diablo Release

Started at Diablo summit
6 months of dev resulted in Diablo release:
  v1.0 of the Quantum L2 API
  API extensibility framework
  API client library and CLI
  Nova Integration via the QuantumManager
  OpenStack Dashboard Integration
  Plugin framework & two publicly available plugins:
    Open vSwitch Plugin
    Cisco UCS/Nexus Plugin

Available at: http://launchpad.net/quantum
Docs at: http://docs.openstack.org/incubation/
Project Status: Who should use Quantum?

“Early adopters” are already putting Quantum into their OpenStack trial deployments.
Caution: deployments are by people who understand the limitations of Quantum and are looking to improve the platform by running/testing it.
Some aspects of the traditional Nova networking model are not yet integrated with Quantum (e.g., floating IPs, DHCP)
This will improve significantly over the Essex time frame, watch for updates on the OpenStack list + IRC meetings.
Quantum Essex Roadmap

Top Priority: production quality + usability:
   Automated functional, integration & scale testing
   Parity/integration with all existing Nova network capabilities.
   Dashboard + client enhancements for ease of use.
   API auth with Keystone (mostly done)
   Packaging & distribution

Essex is target for production quality.
Some will deploy pre-production using Essex milestones.

Strategic Directions:
   Implementing L3 service for “virtual routers”
   Advanced RBAC for API
   VXLAN/NVGRE, and much, much more...
Questions / Comments?

Thanks!

Come join us:
http://wiki.openstack.org/Quantum
netstack@lists.launchpad.net