Agenda

- Gaps in multi-site OpenStack for NFV
- Tacker
- Kingbird
- Tricircle
- Putting it all together
Disaster recovery of application...
- App level consistency snapshot
- Volume level cross site replication

OpenStack VIM

vRouter 1 (IPv6)
VNF2

NFV Orchestrator

vRouter 2 (IPv6)
VNF2

OpenStack VIM

OpenStack
Gaps in Multi-site OpenStack for NFV

- NFV Orchestrator
  - Tenant level L2/L3 networking for E-W traffic
  - Tenant level quota control and resources view
  - Tenant level image, SSH key, security group, IP/mac, metering, monitoring...

OpenStack VIM

VNF1
vRouter 1 (IPv6)

OpenStack VIM

vRouter 2 (IPv6)
VNF2
Gaps in Multi-site OpenStack for NFV

- NFV Orchestrator
- Distributed VNF
- Distributed Session DB
- Stateless Session Process Unit
- Distributed LB
- OpenStack

- Tenant level L2/L3 networking for E-W traffic
- Tenant level quota control and resources view
- Tenant level image, SSH key, security group, IP/mac, metering, monitoring...
| Use case 1 | Multisite identity service management | • Bug reported to KeyStone (Liberty) |
| Use case 2 | VNF high availability across VIM | • Cross OpenStack L2 networking requirements to L2GW (Newton) |
| Use case 3 | Multisite VNF Geo-Site disaster recovery | • Group VMs consistency snapshot via quiesce/unquiesce API (Newton)  
• Volume level replication |
| Use case 4 | Resources management and synchronization. | • Kingbird  
• Tricircle  
( for different deployment scenario ) |
| Use case 5 | Centralized monitoring service. | |
End-to-end Service Orchestration

Monitor and heal service chain in case of VNF failures
Modify/update service chains
Resilient to WAN bandwidth and delay
Tacker Project Overview

- Open NFV Orchestrator
- VNF life cycle management
  - monitoring framework
  - configuration framework
- VNF catalog management
  - TOSCA template support
- EPA support
  - CPU-Pinning, Huge Page, NUMA awareness, SR-IOV
- Auto resource creation
  - Flavor, Network and Image creation
Multisite VIM Support

- Unified view of VIM management using a single dashboard
- Ease of deployment for operators to integrate Tacker with their existing OpenStack installations with minimal overhead
- Fulfillment of a certain level of RO functionality, by combining necessary existing OpenStack modules such as heat and keystone
Unified View of VIM Management

Liberty

Tacker

VIM 0
(local OpenStack Controller)

Mitaka

Tacker

Site 1 - VIM
(default – local OpenStack Controller)

Site 2 - VIM
(OpenStack Controller)

Site 3 - VIM
(OpenStack Controller)

Tacker can instantiate VNFs only on same OpenStack controller

VNF placement on specific target OpenStack VIM
Explicit Region support
Multiple OpenStack Versions

- Operators have multiple OpenStack VIMs with different OpenStack versions
- Detect KeyStone and HOT template version and translate accordingly
- Gracefully downgrade resource requests on older VIMs
Tacker Multisite Architecture

- Multisite VIM module part of orchestrator component
- Pluggable driver framework
  - (aka Bring your own VIM)
- VIMs shared across tenants
- Horizon dashboard and CLI support
Tacker Resources

- https://wiki.openstack.org/wiki/Tacker
- https://github.com/openstack/tacker
Kingbird Project

Resource synchronization and management for multi-region OpenStack deployments.

Based on OPNFV Multisite UC5, UC6.

Aggregated view of distributed resources.

Synchronization (ssh-keys, security groups, images, flavors, etc.)

Centralized quota management.
Quotas in OpenStack

- Defined on a per-region basis.
- Fragmented quota management in nova, neutron, cinder.
- Assigned per region.
  - E.g. max 50 instances in R1, max 30 instances in R2
- No process for synchronizing allocated quotas across regions.
Kingbird Quota Management

- New centralized quota management function.
- Global quota limits across multiple regions.
- Minimal impact on the existing OpenStack services.
Kingbird Quota Management

- Use existing APIs to dynamically balance quota values.
- Calculate resource usage upon synchronization.
- Store the default/tenant quota limits in Kingbird.
  - Provide CRUD operations for the known quota limits
  - Expose similar APIs (os-quota-set, os-quota-class-sets)
Kingbird Architecture

**kingbird-api**

Provides an API for managing global quota limits and on-demand quota synchronization.

**kingbird-engine**

Responsible for communicating with OpenStack services in regions, fetching tenant resource usages, periodically rebalancing quota limits.
Kingbird Resources

- Project status, bugs, and blueprints are tracked on Launchpad
  https://launchpad.net/kingbird
- Source code on Github
  https://github.com/openstack/kingbird/
Tricircle is OpenStack API gateway with added value like cross OpenStack L2/L3 networking, volume/VM movement, image distribution, global resource view, distributed quota management …

This makes massive distributed edge clouds work like one inter-connected cloud, one OpenStack.
OpenStack API Gateway:
- Tenant level L2/L3 networking automation for E-W traffic isolation
Tricircle-request forwarding

Forwarding request to regarding bottom OpenStack according to AZ.

If more than one bottom openstack in one AZ, then schedule one.

VM/Volume co-location through the binding of tenant-id and bottom OpenStack.
Tricircle—extend the network to where the VM is, automatically.
OpenStack API Gateway:
- Move tenant’s data (VM, Volume, Image, etc) across site leverage the cross site tenant L2/L3 networking

Create VM with transportation tool, and attach the volume (data to be moved) to the VM, move the data across OpenStack through tenant level L2/L3 networking.

*Conveyor, a project built above Tricircle will help to do this: https://launchpad.net/conveyor
Tricircle Resources

- Project status, bugs, and blueprints are tracked on Launchpad
  https://launchpad.net/tricircle
- Source code on Github
  https://github.com/openstack/tricircle
Multi-site NFV orchestration

Tenant level cross site L2/L3 networking automation, service chaining, data movement
Thanks, Q&A
Tacker Multisite Horizon Workflow
Tacker Multisite Horizon Workflow
Tacker Multisite Horizon Workflow
Tacker Multisite Horizon Workflow
## Tacker Multisite Horizon Workflow

### VNF Manager

<table>
<thead>
<tr>
<th>VNF Name</th>
<th>Description</th>
<th>Deployed Services</th>
<th>VIM</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>openwrt_vnf</td>
<td>OpenWRT with services</td>
<td></td>
<td>AUS-Site</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>vrouter</td>
<td>vRouter</td>
<td></td>
<td>SFO-Site</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>load balancer</td>
<td>ADC</td>
<td></td>
<td>NYC-Site</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>
Need and Gap for Multi-site OpenStack in NFV (3)

A new work item ETSI NFV GS IFA-020 was approved on Apr 7th to provide a report on the NFVO Split, where a first time ever try for a holistic view on multiple site NFVO architecture will be discussed and introduced.

- The NFVO split architecture options should be designed to be complementary and not contradict each other.
- Reference points should be reused as much as possible.
- A potential option is shown on the left.
- In this context, OpenStack would be expected to provide certain level of RO functionality:
  - Could be a combination of modules
  - Could be a not full-fledged RO entity

Disclaimer: The architecture options have not been discussed in IFA yet.
Gaps in these solutions and what different teams are doing to fix it

Landscape for projects working together
Gaps in these solutions and what different teams are doing to fix it

- **Kingbird**
  - Site 1 - VIM (OpenStack Controller)
- **Tacker**
  - Site 2 - VIM (OpenStack Controller)
- **Tricircle**
  - Site 4 - VIM (OpenStack Controller)
  - Site 5 - VIM (OpenStack Controller)
  - Site 6 - VIM (OpenStack Controller)

If tenant level cross site L2/L3 networking automation, service chaining, data movement are needed
Gaps in these solutions and what different teams are doing to fix it

If tenant level quota control, resource sync..etc are required for multi-region deployment
Gaps in these solutions and what different teams are doing to fix it

If tenant level quota control, resource sync..etc are required for multi-region deployment

Kingbird

Region1 Tacker
Site 1 - VIM (OpenStack Controller)

Region2 Tacker
Site 2 - VIM (OpenStack Controller)

Region3 Tacker
Tricircle

Site 4 - VIM (OpenStack Controller)
Site 5 - VIM (OpenStack Controller)
Site 6 - VIM (OpenStack Controller)
Gaps in these solutions and what different teams are doing to fix it

If tenant level cross site L2/L3 networking automation, service chaining, data movement in all sites are needed
Existing projects and solutions

- Multisite
- Tacker
- Kingbird
- Tricircle
Tricircle

OpenStack API Gateway:
- Tenant level quota control and resources view
- Tenant level image, SSH key, security group, IP/mac, metering, monitoring...

Tricircle

OpenStack API

Site 1 - VIM (OpenStack Controller)

OpenStack API

Site 2 - VIM (OpenStack Controller)

OpenStack API

Site 3 - VIM (OpenStack Controller)
Quota sync algorithm

Three regions A, B, C and the global quota limit is $\text{vcpus}=10$ set in Kingbird

**S0**: Initial state.

```
>--------------------------
> |   | A  | B  | C  | Kb |
>--------------------------
> | S0 |    |    |    | 10 |
>--------------------------
```

**S1**: Kingbird syncs the limit in all regions:

```
>--------------------------
> |   | A  | B  | C  | Kb |
>--------------------------
> | S0 | 10 | 10 | 10 | 10 |
>--------------------------
```

>--------------------------
Quota sync algorithm

**S2**: Allocate 2 vcpus in A and 1 in C.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Kb</th>
<th>Au</th>
<th>Bu</th>
<th>Cu</th>
<th>Su</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

**S3**: Tenant allocates 2 more in A, 2 in B, and 2 more in C: so now the usage will be 4 in A, 2 in B, 3 in C (total 9).

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Kb</th>
<th>Au</th>
<th>Bu</th>
<th>Cu</th>
<th>Su</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>S3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>9</td>
</tr>
</tbody>
</table>

Global_remaining_limit = Kingbird_global_limit - Su(sum of all usages)
Region_new_limit = Global_remaining_limit + resource_usage_in_that_region.
1 Create Network1

2 Create VM1(Network1, AZ1)

3 Create Network1-1

4. update Network1( segment1 = Network1-1@ AZ1)

5. Create Port1 for VM1

6 Create VM1(Port1, Network1-1)

*support from Networking L2GW project
1. Create Network1-1
2. Create VM1 (Network1-1, VLAN1)
3. Create Network1-2

7. Create VM2 (Network1, AZ2)
   - Nova API-GW
   - Tricircle Plugin
   - L2GW Driver
   - Neutron API
   - 9. update Network1 (segment2 = Network1-2 @ AZ2)
   - 10. Create Port2 for VM2

8. Create Network1-2

11. Create VM2 (Port2, Network1-2)

bottom OpenStack