Multi-Cloud Practices with NFV and ONAP

Xinhui Li, VMware
Lingli Deng, China Mobile
Agenda

• Motivation
• Driven by Use Case
• Design and Architecture
• End2End Integration
Why Orchestration?

Global Service Orchestrator

SDNO

Domain Service Orchestrator

Legend: __ SDN-O
- NFP-O

Thin SD-WAN CPE

Thin SD-WAN CPE

Enterprise/Residential SDN

Overlay

Underlay

Option 1

Option 2

Option 3
ECOMP
• 2+ years of Deployment Maturity at AT&T
• Design + Orchestration + Control + Policy + Analytics
• Model-based design enabling self-service instantiation and close loop remediation

ONAP - OPEN-O and openECOMP

• Open TOSCA model
• Established Open Source process and tools
• Architected for ease of VNF insertion (SDK)
ONAP Architecture
ONAP Architecture

- Support multiple NFVI infrastructure environments: OpenStack, VMware VIO, Microsoft Azure, etc.
- DCAE - NFVI Telemetry and events for close loop remediation
- OOF - Smart placement of VNFs
- OOM - Container Orchestration and leveraging cloud-native technology advancements towards ONAP platform S3P
  - Monitoring the progress
  - Ensure compatibility with Multi-Cloud
  - Support multiple NFVI infrastructure environment --- OpenStack, VMware VIO, Microsoft Azure, etc.
Definition:
- enable ONAP to deploy and run on multiple infrastructure environments, including OpenStack and its different distributions, public and private clouds, and micro services containers, etc.
- provide a Cloud Mediation Layer supporting multiple infrastructures and network backends so as to effectively prevents vendor lock-in.
- decouple the evolution of ONAP platform from the evolution of underlying cloud infrastructure, and minimizes the impact on the deployed ONAP while upgrading the underlying cloud infrastructures independently.

Provided Interfaces:
- Resource LCM interface
  - Support VNF instantiation/termination (Resource create/delete/update/query)
- VIM registry/un-registry Interface
  - Support registry/un-registry of multiple clouds (VIM create/delete/update)
- VIM FCAPS management Interface
  - Support FCAPS required data collection, event/alert/metrics federation functions (message pub/sub)
- VIM capacity/capability query
  - Support placement and scheduling purpose (capacity/capability query/report)
Consumed Interfaces:
- ONAP
  - Inventory Service Interface from Available and Active Inventory
  - Message pub/sub interface from DMAAP
  - Logging interface from Logging project
- The third party Cloud API
  - OpenStack Interface
    - Ocata
    - Mitaka
  - VMware VIO interface (Ocata)
  - Wind River Titanium Interface (Mitaka)

Consumed Models:
- Existing
  - VES data format
- Underworking
  - Hierarchical Infrastructure Resource Information Model
  - FCAPS Data Model
Multi VIM/Cloud Architecture

Release 1 scope
Enable Vanilla OpenStack as well as VMware Integrated OpenStack, Wind River Titanium OpenStack
Align with two key use cases VoLTE and vCPE

Release 2 scope
Event/Alert Federation
Service onboarding
HPA enabling
Enabling smart placement
Image service
Performance and concurrent processing improvement
API Design Principles

- All API starts with **MultiCloud Name Space**, followed by **functional module name space**

- Support existing **OpenStack APIs** as default functional modules. Minimal code changes to existing ONAP modules that already use OpenStack.

- New name space for **common cloud functionality**, beyond common functionalities
createVfModule

heat.queryStack (cloudSiteId, tenantId, vfModuleName)

cloudConfig.getCloudSite (cloudSiteId)

getHeatClient (cloudSite, tenantId)

cloudIdentity.getKeystoneUrl

keystoneTenantClient.tokens

msb.onap.org:80/api/multicloud/v0/[cloud owner]_{region}/identity/v3

return token and service catalog

heatUrl = KeystoneUtils.findEndpointURL

“Returned heat endpoint (SBI decided by different Cloud providers)” +/stacks

queryHeatStack (heatClient, stackName)

return heat stack list
• Different Cloud Backend -> different Alerts/Events/Meters, two roles:
  • Multi Tenancy workload
  • Admin

• Motivation:
  • Despite Alerts, Events and metrics are needed from VIM controller
    • Close the Service Resilience loop in ONAP needs underlying events (from not only resources but also vim controller)
    • Close the Auto-Scaling Resilience loop in ONAP needs underlying Meters/Alerts etc.
  • Aligning/translating FCAPS modeling is needed from different VIM controllers
  • Framework enhancement is needed
    • Allowing for meaningful close loop for handling time-sensitive event/alerts
    • Allowing streaming mechanism for handling accuracy-sensitive metrics
    • Extending ves agent implemented in R1 which only emit vm abnormal alert for more period data collection
  • Visualization on Portal
Multi-Cloud Control Plane for Event/Alert/Streaming/Agent Services

- **Event Service**
  - Federate events from different VIM providers with ONAP Message/Data bus services
  - Allow to be configured by the control plane about listener and endpoints
  - Not only events of different backend clouds, but also events from VIM controllers

- **Streaming Service**
  - Federate meters from different VIM providers with ONAP Message/Data bus services
  - Allow to be configured by the control plane about gate rate and water mark
  - Achieve an ideal long term output rate which should be faster or at least equal to the long term data input rate

- **Alert Service**
  - Alert comes from meters or events, or pre-defined in different backend Clouds
  - Allow to be customized
- **Extensible MC framework**
  - Pluggable framework
  - Intelligence contained
  - Information models standardized
  - Customizable to work with upper layers
  - Policy-driven data distribution

- **Compensate each other with different Telemetry:**
  - OpenStack control plane service status
  - Underlying OS/hypervisor status
  - Customizable by policy
Listening to VIM events

VIM policy driven healing actions

Multi Cloud

DMAap

DCAE

Holmes

Policy

Controllers

Multi Cloud

Pub

Sub

Pub

Sub

Pub

Recover

Stop VM

Start VM

VM actions
**Multi-Cloud Engine pool Framework Improvement**

- **Performance improvement**
  - The number of Handler Engines is configurable, and is the number of CPU cores by default.
  - Each Handler Engine will use scheduler to do multiple jobs.

- **Stability improvement**
  - Bootstrap process will watch handler engine and respawn it if any exits unexpectedly.
  - Handler engines are run as processes so that memory and resources are isolated.
  - Store logs into files

- **Scalability improvement**
  - Bootstrap process uses fork to create multiple Handler Engines
  - Multiple Handler Engines share a single socket with bind to the `<IP address>:<port>` of the API server
  - Extend northbound API by using YAML files

---

**Diagram:**
- **Multi-Cloud**
- **Bootstrap process**
- **API Handler**
- **Event Handler**
- **Alert Handler**
- **Scheduler**
- **Service Engine 1**
- **Handler Engine 2**
- **Handler Engine N**
- **Engine Scheduler**
- **Log File**
- **Extensible API Framework**: `http://<IP address>:<port>`
- **Controllers**:
  - Portal
  - Other controllers
  - SDC
  - VNFSDK
  - Image management
  - AAI
- **Platforms**:
  - OpenStack
  - VMware
  - Wind River
  - Kubernetes
  - Azure

---
• VNF on-boarding and upgrade are:
  • Current No formal way for a VNF to ask for resources and ensure they can be and will be granted at install and reserved thru operation
  • Limited security, isolation, scalability, self healing – VIM & VNF specific
  • Efficiency: in many cases, no other VNF runs on the same platform
  • Operator, VNF vendor and VIM specific – procedures are specifically tailored to each VIM infrastructure, operator and VNF vendor

• Multi Cloud will help:
  • Automatically ensure capacity and capability of required VIM & infrastructure resources
  • Verify/Validate that
    • VIM Image(qcow2 or vmdk)
    • VIM requirements can be met
Image service

- **Image service to handle**
  - Different kinds of storage for image file and meta reservoir
  - Support both Docker or VM images
  - Help VNF onboarding and runtime optimization

- **Primary operations**
  - Image upload/download
  - Image registry/un-registry
  - Copy from one VIM to another VIM
  - Copy from one data store to another data store
  - Launch instances

- **Transparent to the up layer**
  - Auto handling across vim copy and capacity management

- **Multi-Cloud**
  - ONAP Message/Data Bus Layer
  - Control plane for Resource Insurance
  - Event Service
  - Image Service

- **Controllers**
  - Portal
  - Other controllers
  - SDC
  - VNFSK
  - Image management
  - AAI

- **Layers**
  - ONAP Message/Data Bus Layer
  - DMaaS
  - Control Plane for Image Service

- **Storages**
  - OpenStack
  - VMware
  - Wind River
  - Kubernetes
  - Azure
  - Metadef
  - FileSystem
  - VSAN
  - Ceph
  - Swift
Mismatching NUMA requirement to NUMA resource, case 1

• VNF usually consumes big chunk of memory
• Un-balanced usage of NUMA resource in NFVI hosts
• Waste usage of memory usage
• What usually observed: The avail. memory are more than request, but no NFVI host could be placed for a VNF

Case 1: Single NUMA node request memory more than avail. memory of any single NUMA node resource, but less than total avail. Memory resource

VM 1 WORKLOAD

NUMA node 1:
- vCPU: 8
- memory: 32G

NFVI HOST 1

NUMA node 1:
- pCPU: 3.75
- Avail. memory: 18G

NUMA node 2:
- pCPU: 5.15
- Avail. memory: 30G
• There is no single predefined NUMA requirement fit for dynamically changed NUMA resource

Case 2: Single NUMA node request memory more than avail. memory of any single NUMA node resource, but less than total avail. Memory resource
- MultiCloud tune the NUMA requirement to fit the VM workloads into NFVI Host NUMA resource
- NFVI Host NUMA resource must be exposed/updated into ONAP data store

Mismatching NUMA requirement

VM 1 WORKLOAD
- NUMA node 1: vCPU: 4 memory: 16G
- NUMA node 2: vCPU: 4 memory: 16G

VM 2 WORKLOAD
- NUMA node 1: vCPU: 4 memory: 16G
- NUMA node 2: vCPU: 4 memory: 4G

NFVI HOST 1
- NUMA node 1: pCPU: 3.75 Avail. memory: 18G
- NUMA node 2: pCPU: 5.15 Avail. memory: 30G

MultiCloud EPA matching logic
- Transform

pass-through
ONAP End2End Integration
External system Registration
## Service Lifecycle Management

The image shows a UI for managing services and network packages through a Life Cycle Manager interface. The interface includes a table with columns for Service Instance ID, Service Name, Service Type, and Action. The table lists various network service instances with their respective details and options to delete each entry.

### Table Example

<table>
<thead>
<tr>
<th>Service Instance Id</th>
<th>Service Name</th>
<th>Service Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>f9036540-c4a8-4b5f-bbae-461c7b99b4f</td>
<td>demo_test5</td>
<td>VoLTE type</td>
<td>Delete</td>
</tr>
<tr>
<td>d560252f-d012-40e8-8e8d-7f592b2602ba</td>
<td>demo_lms LTE</td>
<td>E2E Service</td>
<td>Delete</td>
</tr>
<tr>
<td>3c27db3a-12e4-4d42-8809-04c85276d5cb</td>
<td>test3</td>
<td>VoLTE type</td>
<td>Delete</td>
</tr>
<tr>
<td>7b5999-506d-4d13-c81e-db2e5e7e3fa</td>
<td>vIMS_demo_test5</td>
<td>Network Service</td>
<td>Delete</td>
</tr>
<tr>
<td>7c117751-16d-420e-865b-587b96dbcbbe</td>
<td>vIMS_test3</td>
<td>Network Service</td>
<td>Delete</td>
</tr>
<tr>
<td>f11866c-4830-47a-8b5a-9c6b65b9591</td>
<td>vIMS_onap_r1_demo_d</td>
<td>Network Service</td>
<td>Delete</td>
</tr>
<tr>
<td>9d8df8c-4a08-41f4-8a69-8645e4641f8</td>
<td>vIMS demo test10</td>
<td>Network Service</td>
<td>Delete</td>
</tr>
</tbody>
</table>

This UI is part of a larger Open Networking environment, as indicated by the logo and the text "ONS - NORTH AMERICA - OPEN NETWORKING // Integrate, Automate, Accelerate."
### VNFs on Multi VIM/Cloud

#### Overview

#### Usage Summary

Select a period of time to query its usage:

- **From:** 2017-12-07
- **To:** 2017-12-08

**Active Instances:** 52

**RAM:** 64GB

**This Period's VCPU Hours:** 7446.67

**This Period's GB Hours:** 54432.16

**This Period's RAM Hours:** 121594016.57

#### Usage

<table>
<thead>
<tr>
<th>Project Name</th>
<th>VCPUs</th>
<th>Disk</th>
<th>RAM</th>
<th>VCPU Hours</th>
<th>Disk GB Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>huaweionap</td>
<td>64</td>
<td>2000GB</td>
<td>2040B</td>
<td>2108.79</td>
<td>6598.97</td>
</tr>
<tr>
<td>ZTE_ONAP_PRO</td>
<td>152</td>
<td>1.2TB</td>
<td>4160B</td>
<td>5008.38</td>
<td>41253.22</td>
</tr>
<tr>
<td>admin</td>
<td>10</td>
<td>2000GB</td>
<td>200B</td>
<td>329.50</td>
<td>6598.97</td>
</tr>
</tbody>
</table>

*Displaying 3 items*
VNFs on Multi VIM/Cloud 2

Overview

Usage Summary

Select a period of time to query its usage:

From: 2017-10-07 To: 2017-12-08

Active Instances: 55
Active RAM: 775GB
This Period's VCPU-Hours: 337406.60
This Period's GB-Hours: 3171966.35
This Period's RAM-Hours: 1007910450.37

Usage

<table>
<thead>
<tr>
<th>Project Name</th>
<th>VCPUs</th>
<th>Disk</th>
<th>RAM</th>
<th>VCPU Hours</th>
<th>Disk GB Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>huawelonap</td>
<td>156</td>
<td>870GB</td>
<td>513GB</td>
<td>220875.14</td>
<td>126306.57</td>
</tr>
<tr>
<td>admin</td>
<td>6</td>
<td>400GB</td>
<td>128GB</td>
<td>7528.84</td>
<td>562447.30</td>
</tr>
<tr>
<td>Nokia_INS</td>
<td>80</td>
<td>522GB</td>
<td>150GB</td>
<td>64127.52</td>
<td>449966.19</td>
</tr>
<tr>
<td>ZTE_EPC_PRO</td>
<td>42</td>
<td>818GB</td>
<td>100GB</td>
<td>44904.76</td>
<td>896518.92</td>
</tr>
</tbody>
</table>
Closed Loop for NS resilience

Alarm from VIM via MultiCloud (Power off)

Alarm from VNF via EMS (ZTE_xGW_39_PFU_1 VM of SPGW)

Alarm Collection & Correlation (VES collector/Holmes)

Policy Matching (Policy)

Healing Restart VM (VFC)

Recover
Across layered Alerts and Metrics
• Backup
• **Use Cases**
  - VoLTE, VCP, 5G

• **Architecture**
  - Member of R2 Architecture tiger team

• **ONAP Mini Summit**

• **ONAP Developer Event**
  - Multi-Cloud project incubation, Beijing event, [https://wiki.onap.org/display/DW/ONAP+June+2017+F2F+in+Beijing+China](https://wiki.onap.org/display/DW/ONAP+June+2017+F2F+in+Beijing+China)
FCAPS Modeling

• Different categories of Data: resource status, alerts, performance, and so on
  • Interface protocol
  • Data format
  • Metrics
  • Logs

• All the data should be gotten based on the same pre-condition
  • Period or event?
  • NTP and facilitation requirements
  • Delay sensitive
  • Different data sources

• Need spec and modeling efforts to formulate these detentions
Alert/Event/Meter Federation Framework

• Different Cloud Backend -> different Alerts/Events/Meters, two roles:
  • Multi Tenancy workload
  • Admin

• Use cases:
  • Close the Service Resilience loop in ONAP needs underlying events(from not only resources but also vim controller)
  • Fcaps propose the requirements for monitoring and management of Alerts/Events/Meters
  • Visualization on Portal