Closing the loop
End to End Failure detection, RCA and auto healing

Ohad Shamir
Moshe Lavi
Andras Urban
Agenda

1. VIM Failures Impact
   VIM failure and its affect on VNF/VNFM

2. Demo part 1
   VNF alarm detection along with end to end RCA. No auto healing

3. Vitrage
   Expanding Vitrage with VNF topology and status information

4. VNFM
   Application life cycle management and automation

5. Demo part 2
   Closing the loop, Auto healing in action

6. Recap and Q&A
NFVI/ VIM Failures Impact

Planned Maintenance

Unplanned server reboot
Temporary outage

Unplanned server outage
HW recoverable operations

Unplanned server outage
Unrecoverable HW failure
## NFVI/ VIM Failures Impact

<table>
<thead>
<tr>
<th>Type</th>
<th>Planned Maintenance</th>
<th>Unplanned server reboot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Cases</td>
<td>• SW patching or upgrade</td>
<td>• SW glitched that caused an unexpected reboot</td>
</tr>
<tr>
<td></td>
<td>• Replacement of faulty HW such as FAN tray, RAM, disk</td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>• Migrate VMs</td>
<td>• Check VMs status</td>
</tr>
<tr>
<td></td>
<td>• Graceful shutdown</td>
<td>• Reboot server if error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Recreate server if still unhealthy</td>
</tr>
</tbody>
</table>
## NFVI/ VIM Failures Impact

| Type | **Unplanned server outage**  
|  | **HW recoverable operations** | **Unplanned server outage**  
|  |  | **Unrecoverable HW failure** |
| Use Cases | - OS Disk failure  
|  | - Total Power supply failure  
|  | - Tenant NIC HW failure  
|  | - Infrastructure NIC HW failure | - HW issues such as motherboard failure |
| Actions | - Disable scheduling new VMs  
|  | - Host auto evacuation  
|  | - Server return to operate normally after manual recoverable operations | - Disable scheduling new VMs  
|  |  | - Host auto evacuation  
|  |  | - Scale-in (remove server from cluster) |
Use Case – end-to-end flow
Assurance flow – Host NIC failure

NIC failure triggers alarms on impacted Hosts and VMs. VNFM’s monitoring function detects the fault.

NFVO correlates VNF resource fault to NS and notifies OSS

EMS correlates the VNF resource fault with app. fault, notifies OSS

VNFM identifies impact on VNFs and notifies NFVO and EMS

Service layer correlates failure impact with the relevant e2e service

Service layer correlates failure impact with the relevant e2e service

Service 1

Service 1

NS 1

NS 2

VNF 1

VNF 2

VNF 3

VNF 1 VNFC

VNF 2 VNFC

VNF VNF

VM 1

VM 2

VM 3

VM 4

VM 5

VM 6

VM 7

VM 8

Blade/Server 1

Blade/Server 2

OSS

CBND

NFVO

EMS

CBAM

G-VNF

CBIS

CBIS

VNF

VNF

© 2018 Nokia
Demo part 1
OpenStack Vitrage – an **official** project in OpenStack
Contributing to improve OpenStack operational simplicity

- Resources
- Events
- Alarms

**Organize**

**Physical to virtual mapping**

- Deduced alarms

**Analyze**

**Root cause alarm calculation**

- Alarm aggregation

**Visualize**

https://wiki.openstack.org/wiki/Vitrage

© 2018 Nokia
OpenStack Vitrage

Vitrage Functions

- **Deduced alarms and states** – raising alarms and modifying states based on system insights
- **Root Cause Analysis** – understand what causes faults to occur
- **Holistic & complete view** of the system

Architecture Highlights

- **Resource Topology Engine** - reflect how different entities relate to one another
- **Extendibility and configurability**: template driven configurable business logic utilizing multiple data sources
- **Multi-tenancy** supported
- **Clear Visualization** of Vitrage insights
Vitrage High Level Architecture

**Data Sources (extendible)**
- External monitoring tools
- OpenStack projects
- Physical topology configuration files (e.g. switch config)

**Horizon plug-in**
- Topology view ("sunburst")
- Vitrage alarm list
- RCA diagram per alarm
- Entity graph view

**Notifiers**
- Expose Vitrage alarms to other projects or external systems
- Execute actions like Mark host down in Nova

**Templates for deduced alarms & RCA**
- Provides customization facilities
- Examples: High CPU load, High memory load, Physical switch failure, Host NIC failure, OVS failures (load, DB, connectivity)
Vitrage Template – Example: Host High CPU Load

**Scenario 1 – Raise Alarm**
When high CPU load on host and host contains instance:
1. Raise deduced alarm on instance named cpu performance degradation
2. Set state to suboptimal on instance

**Scenario 2 – RCA**
When high CPU load on host and host contains instance and cpu performance degradation alarm on instance, add causal relationship

**Scenario 3 – Set host state**
When high CPU load on host, set host state to suboptimal
Sample use cases supported by Vitrage

<table>
<thead>
<tr>
<th>Event alarms</th>
<th>Metric alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neutron OVS failures (on host)</td>
<td>Host high CPU load</td>
</tr>
<tr>
<td>Host unreachable</td>
<td>Host high memory usage</td>
</tr>
<tr>
<td>Ceph unhealthy</td>
<td>Host low disk space</td>
</tr>
<tr>
<td>Disk smart test scenarios</td>
<td>Ceilometer failure – unable to collect metrics</td>
</tr>
<tr>
<td>Physical NIC failure</td>
<td></td>
</tr>
</tbody>
</table>
Manually intensive tasks to be automated

1. User 1 creates cloud resources
   User 1 verifies that everything OK and hands over

2. User 2 does initial configuration of the VNF
   User 2 hands over to EMS configuration

3. User 3 integrates the VNF with the EMS

4. When scale or upgrade request arrives, step 1-3 repeats

5. NOC operator detect and RCA a failure
   User 1-2-3 require to heal the application in a joint work

Higher scale and reduced human errors drive the need for automation
CBAM’s (G-VNFM) main functionality

One-click operations through the entire VNF lifecycle

Full scale VNF lifecycle management

- Create
- Instantiate
- Scale
- Upgrade
- Heal
- Terminate
- Delete
CBAM Auto-Operations

Overview

- Near real-time monitoring for detecting the need to heal / scale by scheduled Mistral workflows and Ansible playbooks
- Ask VNF about “scale me”, “heal me” or collecting indicator KPIs
- Configurable algorithms for KPI evaluation
- Can be switched on / off at VNFM or VNF level

Execution of the operation happens as if triggered via the NBI
CBAM Basic Auto-Operations
Architecture

CloudBand Application Manager

- Ansible
  - Ve-Vnfm-vnf

- Mistral Workflow engine
  - Ve-Vnfm-em

- VNF Topology & Lifecycle Manager
  - Or-Vnfm
  - Operation trigger

- Workflow scheduler & operation trigger

- Indicator KPI collection via Ansible

OpenStack

- VNF

- VIM controllers

- NFVO

- EM

- CloudBand Application Manager

- Operation trigger

- Scheduled workflow management

- Trigger lifecycle operations (Heal, Scale, etc.) can.
Closing the Loop – end to end failure detection and auto healing
CBAM & Vitrage

1. VNF failure reported
2. Correlate VIM to VNF failure
3. VIM failure and RCA reported
4. Auto healing scheduled action Activated
5. Update VNF Topology
6. Instantiate VNF Template
VNF-M (CBAM)

VNF Lifecycle Management

VM Resource FM

VIM Resource FM in CBAM

E2E Fault Management supported by Alarm Correlation in EMS

Virtual Infrastructure Manager

EMS

Alarm Correlation

Virtualized network functions

CSCF

TAS

HSS

SDL

Hypervisor

VMware ESXi

KVM

SDN

SAN

Data center

Hardware (x86)
Demo part 2
We now have a closed loop
Summary

Network Function and physical infrastructure are separated

H/W failure impact on application is not as straightforward

Need for alarm correlation & RCA at multiple levels

Split operational responsibility needs to be automated to maintain high SLA

The VNFM plays a major role to ensure

Day to day automated LCM to reduce burden and prevent error mistakes

Alarm correlation at multiple levels (VIM, EMS, NFVO) to simplify RCA

Framework for auto action to reduce failure time and ensure SLA

Hierarchical monitoring increases modularity, simplifies network changes, and eases the path to automated operations
Closing the loop
End to End Failure detection, RCA and auto healing

More info:

CBAM: https://networks.nokia.com/products/cloudband

Vitrage: https://wiki.openstack.org/wiki/Vitrage
Vitrage example: Host NIC failure

- Monitor host NIC (public/tenant network) by Zabbix, raise an alarm when failed
- Vitrage will receive alarm from Zabbix, add it to the entity graph, connected to the Host NIC vertex
- Find matching scenario (template) and perform the following actions:
  - Raise deduce alarm on host (and add it to the entity graph)
  - Change host state in Vitrage (may trigger also calling Nova API to modify state)
  - Add causal link between alarms
- Once the deduced alarm on host is added, a similar flow will occur on the hosted instances
- Notify VNFM on failure. VNFM to decide and take recovery actions
Agenda

• Intro, introducing 3 Nokia team members – Ohad 1 min
• Intro about multiple VIM failure and the outcome it will have on VNF/VNFM – Ohad 5 min
• Nic failure (no healing) demo and how it effects on VNF/VNFM – Andras 5 min
• Vitrage review how to detect failure and provide RCA – Ohad 10 min
• CBAM in general and CBAM role doing VNF auto healing – Moshe 7 min
• Full integration demo – failure + auto healing – Moshe and Andreas 5 min
• Summary – Moshe 3 min
• QA – 4 min