OSM#9 Hackfest
Hack 1: Adding Monitoring to VNFs
Subhankar Pal
(Altran)
Current Architecture & Features

OSM Service Assurance
Service Assurance MDG

Main components

MON
- Covers the basic use cases, with a solid architecture to expand them easily.
- Opportunities to enhance usability.

POL
- Designed around the autoscaling use case.
- Starting to cover VNF alarms.
- Architecture needs a revisit based on expected use cases.

Prometheus
- OSM’s TSDB for metrics since REL5
- Opportunities to enhance multi-tenancy to match new RBAC capabilities.

Auxiliary/Optional

Grafana
- Integrates seamlessly with Prometheus.
- Great tool for enhancing usability of the system’s Service Assurance

ELK
- Proved seamless integration with OSM.
- Main use case remains at log processing where stack is used.

And an upcoming Placement module!
MON Architecture

Formal documentation: https://osm.etsi.org/gitlab/osm-architecture/osm-arch-doc/blob/master/04-mon.md
When launching a new instance of a Network Service or Slice Instance (n x VNFs) which is described with the collection of VNF Metrics that come from infrastructure (NFVI), the following components interact.

1. lcm - metrics-collection primitive
2. Juju-metrics primitive
3. Continuously looks for active VNFs with metrics
4. API calls to VIM/VCA, to collect metrics
5. Looks for active metrics
6. Reads & presents metrics
When creating Projects or Network Services, Grafana dashboards are created automatically and the following elements interact.

1. continuously looks for new projects or NS with metrics

2a. if project: create Project dashboard

2b. if NS: create NS dashboard (with sample graphs)

(*) continuously delete obsolete dashboards
metrics collection

- VDU Metric Collection from VIM

```yaml
monitoring-param:
  - aggregation-type: AVERAGE
    id: agw_cpu_util
    name: agw_cpu_util
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_cpu_util
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_memory_util
    name: agw_memory_util
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_memory_util
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_packets_received
    name: agw_packets_received
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_packets_received
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_packets_sent
    name: agw_packets_sent
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_packets_sent
      vdu-ref: magma-agw-vdu
```
### Prometheus collects the following metrics from “MON Exporter”

<table>
<thead>
<tr>
<th>Metric</th>
<th>Collection type</th>
<th>Behavior</th>
<th>KPI</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIM Status</td>
<td>Infrastructure</td>
<td>By default</td>
<td>status (up/down)</td>
<td>vim_id</td>
</tr>
<tr>
<td>SDNC Status</td>
<td></td>
<td></td>
<td>status (up/down)</td>
<td>sdnc_id</td>
</tr>
<tr>
<td>VM Status</td>
<td></td>
<td></td>
<td>status (up/down)</td>
<td></td>
</tr>
<tr>
<td>VDU CPU Utilization</td>
<td>VNF</td>
<td>Enabled by descriptor</td>
<td>utilization, rate, etc.</td>
<td>nsr_id, vnf_member_index, vdu_name</td>
</tr>
<tr>
<td>VDU Memory Utilization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDU Packet forwarding</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VNF Metrics through Juju (to be deprecated)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Proposals

OSM Service Assurance
New methods for VNF Indicator Collection

Objective: Evolve the way OSM collects VNF indicators to allow for more compatibility with VNFs, real-time collection and standards alignment.

A first approach is using additional “Prometheus exporters”
Objective: OSM Operators can install OSM and immediately and permanently know the health of the system.

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Feature 7898</th>
<th>Feature 8132</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSM on Kubernetes</td>
<td>OSM on Docker Swarm</td>
</tr>
<tr>
<td></td>
<td>● Prometheus Operator Chart (New prometheus instance, Grafana and different exporters: node, cadvisor, etc.)</td>
<td>● Grafana promoted to OSM stack.</td>
</tr>
<tr>
<td></td>
<td>● Other charts: MongoDB, MySQL and Kafka exporters</td>
<td>● Node exporter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● CAdvisor exporter</td>
</tr>
<tr>
<td>Additional components</td>
<td>Multiple Grafana dashboards for a comprehensive health check of the system.</td>
<td>Single Grafana dashboard with the most important system metrics.</td>
</tr>
</tbody>
</table>
Project-scoped VIM/VNF Metrics

Objective: Follow RBAC structure for metric consumption.

- Prometheus does not support multi-tenancy, other projects need to be explored (e.g. Cortex)
- Short-term proposal is to add a label for `project_id` in all Prometheus metrics
Grafana Dashboard Automation

Objective: adding to the previous feature, a new “MON Dashboarder” component will take care of dashboard “lifecycle”.

<table>
<thead>
<tr>
<th>Updates in...</th>
<th>...automates these dashboards...</th>
<th>...and these Grafana resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM installation</td>
<td>System Metrics, Admin Project-scoped</td>
<td>Admin-privileges</td>
</tr>
<tr>
<td>OSM Projects</td>
<td>Project-scoped (Grafana “team” privileges)</td>
<td>Grafana “team” privileges</td>
</tr>
<tr>
<td>OSM Users</td>
<td>-</td>
<td>Grafana users to teams</td>
</tr>
<tr>
<td>OSM Network Services</td>
<td>NS-scoped</td>
<td>-</td>
</tr>
</tbody>
</table>

...Let’s play with the prototype!
Hands-on!

VNF Monitoring
Let’s play with metrics and (auto)dashboards!

- On the already created packages make the following changes for VNF package ‘hackfest_magma-agw-enb_vnfd’
Let’s play with metrics and (auto)dashboards!

monitoring-param:
  - aggregation-type: AVERAGE
    id: agw_cpu_util
    name: agw_cpu_util
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_cpu_util
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_memory_util
    name: agw_memory_util
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_memory_util
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_packets_received
    name: agw_packets_received
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_packets_received
      vdu-ref: magma-agw-vdu
  - aggregation-type: AVERAGE
    id: agw_packets_sent
    name: agw_packets_sent
    vdu-monitoring-param:
      vdu-monitoring-param-ref: agw_packets_sent
      vdu-ref: magma-agw-vdu

• In the VNF Package editor add the following lined in YAML at line #8
Let’s play with metrics and (auto)dashboards!

- In the VNF Package editor add the following lined in YAML at line #61 and update.

```
  monitoring-param:
    - id: agw_cpu_util
      nfvi-metric: cpu_utilization
    - id: agw_memory_util
      nfvi-metric: average_memory_utilization
    - id: agw_packets_received
      nfvi-metric: packets_received
    - id: agw_packets_sent
      nfvi-metric: packets_sent
```
Let’s play with metrics and (auto)dashboards!

• Delete your previous instance and launch a new one!
  
  $ osm ns-list
  $ osm ns-delete ac51ab3d-3972-49c8-9748-a3c22a67a553

• Recreate the network service with monitoring enabled.
  
  $ osm ns-create --ns_name magmaAGW_x --nsd_name hackfest_magma-agw-enb_nsd --vim_account etsi-openstack-x --config_file params.yml
Let’s play with metrics and (auto)dashboards!

Metrics collection is starts (5 to 10 minutes due to current collection period