OSM-MR#9 Hackfest
Day 1: Adding Monitoring to VNFs

Fabián Bravo
(Whitestack)
Current Architecture & Features
OSM Service Assurance
Service Assurance MDG

Main components

- **MON**: Covers the basic uses 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.
- **PLA**: Provides computation of optimal placement of NFs over VIMs.
  - Considers cost of compute/network.
- **Prometheus**: OSM’s TSDB for metrics since REL5.
  - Opportunities to enhance multi-tenancy to match new RBAC capabilities.
- **Grafana**: Integrates seamlessly with Prometheus.
  - Great tool for enhancing usability of the system’s Service Assurance.

Auxiliary/Optional

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

© ETSI 2020
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.
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**

  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
## Metrics collection

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>SNMP Metrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Features

OSM Service Assurance
New Execution environments

A new way of interaction between OSM and VNFs

- Helm charts to manage dependencies and new companion pods
- Useful for collecting metrics like SNMP
- New API to send primitives (Day 0/1 configuration)
- Fully customizable new endpoints for the API
New methods for VNF Indicator Collection

**NEW JOB**
- mySnmpMetrics:
  - VDU Mgmt IP
  - SNMP Exporter IP
  - NS/VNF/VDU ID/Names (labels)
  - Metrics prefix

MIB files + generator.yaml
SNMP Exporter + Config API

VNF PACKAGE

1. Instantiation
2. HELM Deployment
3. (Re) Configure
4. Config API
5. LCM
6. Prometheus
7. SNMP Exporter

GET & Export

VNF

Prometheus

VNF

NBI

LCM

EE

Vendor provided (generator template provided by OSM)
Base chart provided by OSM
Hands-on!

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

- On the already created VNF package 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 after line #8:

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

- In the VNF Package editor add the following lined in YAML after line #35/ 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.yaml
  ```
Let’s play with metrics and (auto)dashboards!

Metrics collection is starts (5 to 10 minutes due to current collection period)
Let’s play with metrics and (auto)dashboards!
Let’s play with SNMP metrics!

• First you need to download and install an SNMP exporter Helm Chart:

  $ helm repo add whitestack http://helm.whitestack.com
  $ helm repo update

  $ cd ~/magma/hackfest_magma-agw-enb_vnf/
  $ mkdir helm-charts
  $ cd helm-charts

  $ helm fetch whitestack/eecart --untar
Let’s play with SNMP metrics!

- In `helm-charts/eechart` edit `values.yaml` to have `snmpexporter` as `enabled: true` (last line)
Let’s play with SNMP metrics!

 execution-environment-list:
   - id: monitor
     helm-chart: eechart
     metric-service: snmpexporter
     connection-point-ref: vnf-mgmt
 initial-config-primitive:
   - seq: 1
     name: generate_snmp
     execution-environment-ref: monitor
 config-primitive:
   - name: generate_snmp
     execution-environment-ref: monitor

• Edit the VNFD to reference the chart
Let’s play with SNMP metrics!

Or even faster:

```bash
$ cp /home/ubuntu/examples/02-snmp/hackfest_magma-agw-enb_vnfd.yaml ~/magma/hackfest_magma-agw-enb_vnf/hackfest_magma-agw-enb_vnfd.yaml
```
Let’s play with metrics and (auto)dashboards!

• Delete your previous instance and packages:

  $ osm ns-list
  $ osm ns-delete ac51ab3d-3972-49c8-9748-a3c22a67a553
  $ osm nsd-delete hackfest_magma-agw-enb_nsd
  $ osm vnfd-delete hackfest_magma-agw-enb_vnfd

• Recreate the packages with SNMP enabled:

  $ tar -czf hackfest_magma-agw-enb_vnfd.tar.gz hackfest_magma-agw-enb_vnf/
  $ tar -czf hackfest_magma-agw-enb_nsd.tar.gz hackfest_magma-agw-enb_ns/
  $ osm upload-package hackfest_magma-agw-enb_vnfd.tar.gz
  $ osm upload-package hackfest_magma-agw-enb_nsd.tar.gz
Let’s play with SNMP metrics!

- Launch the network service again and watch its metrics at Prometheus

```bash
$ osm ns-create --ns_name magmaAGW_x --nsd_name hackfest_magma-agw-enb_nsd --vim_account etsi-openstack-x --config_file params.yaml
```
Let’s play with SNMP metrics!