

Open Source  
**MANO**

OSM#9 Hackfest

Hack 3.3 : Adding Auto-Scaling & Alerting to VNFs

Subhankar Pal  
(Altran)



Open Source  
**MANO**

# Current Auto Scaling & Alarms Feature

## OSM Service Assurance

# Revisiting Service Assurance MDG

## Main components

## Auxiliary/ Optional

**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.

**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!**

## Auto Scaling

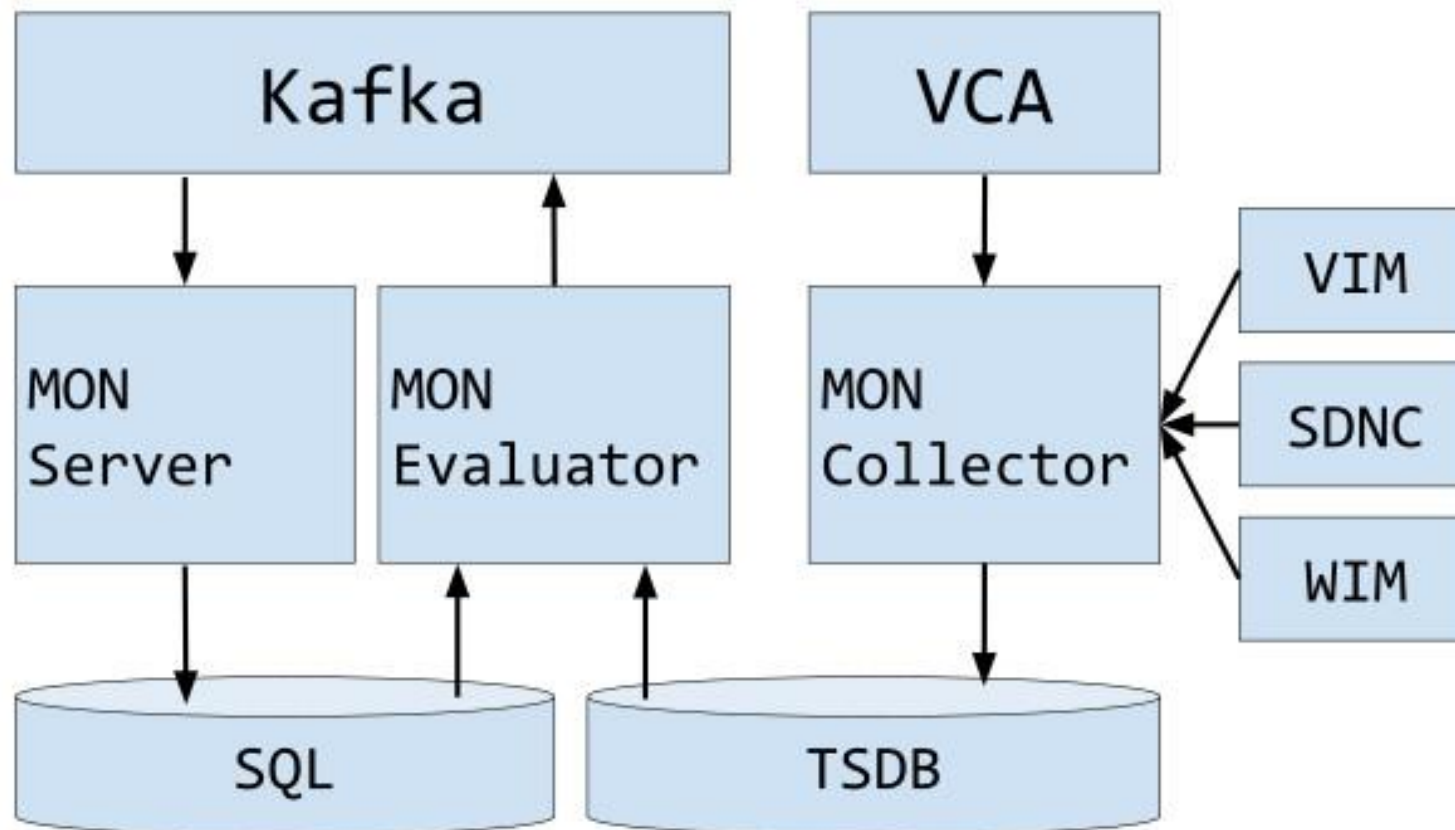
- Auto scaling allows to automatically scale VNFs with a VDU granularity and based on any available metric.
- Scaling descriptors can be included and be tied to automatic reaction to VIM/VNF metric thresholds.
- Supported metrics are both VIM and VNF metrics.

## Alarms

- An internal alarm manager has been added to MON through the 'mon-evaluator' module, so that both VIM and VNF metrics can also trigger threshold-violation alarms and scaling actions

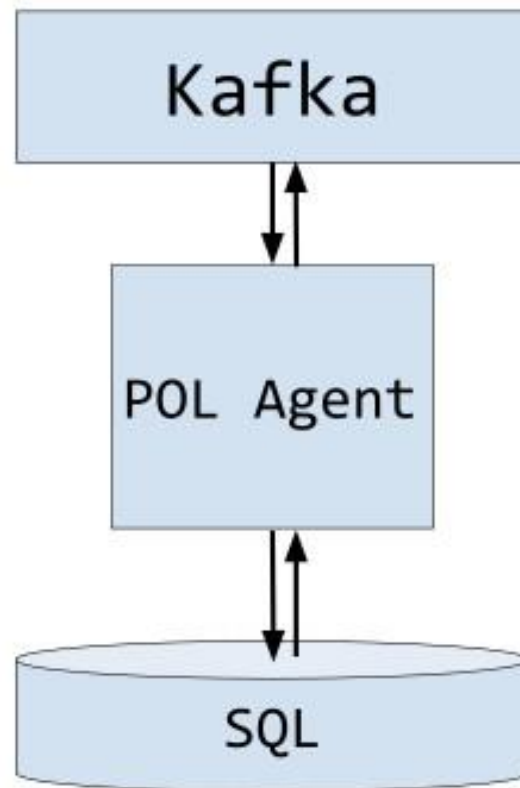
# Revisiting MON Architecture

Formal documentation: <https://osm.etsi.org/gitlab/osm-architecture/osm-arch-doc/blob/master/04-mon.md>



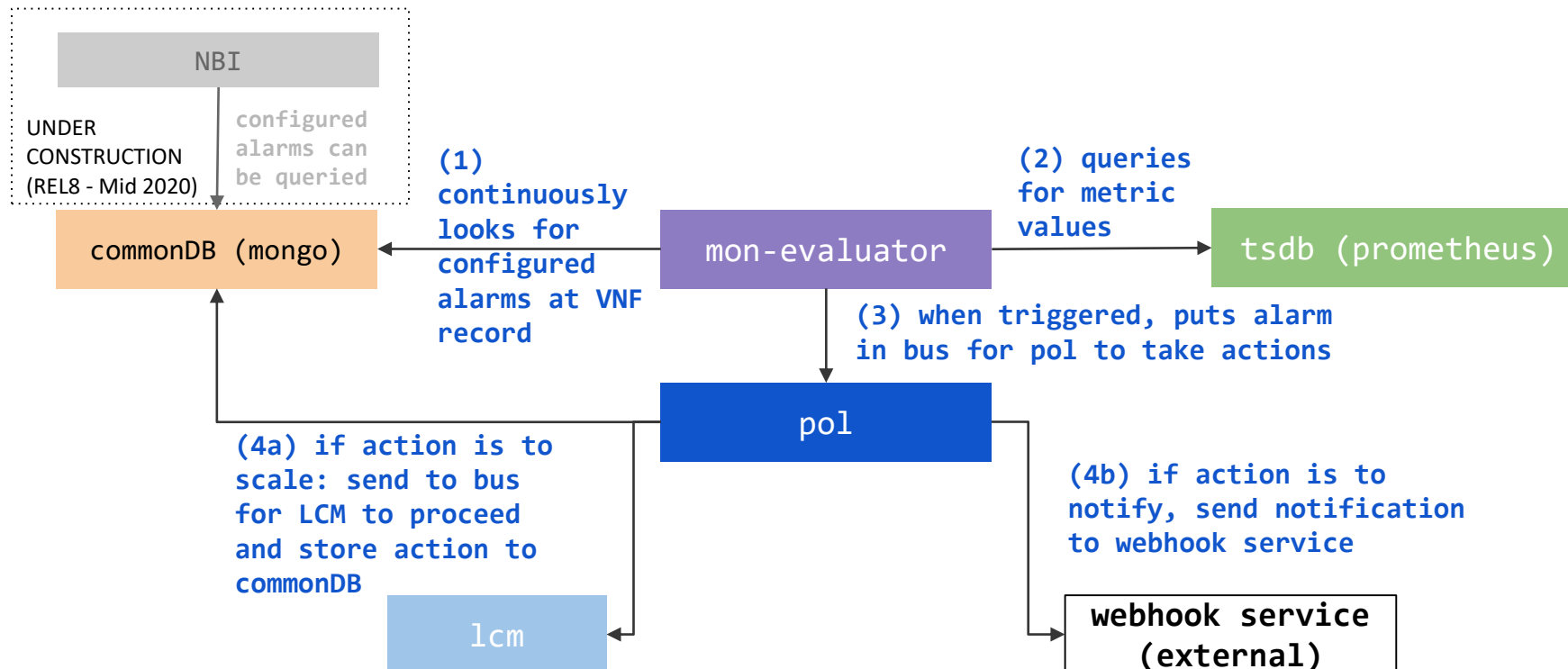
# POL Architecture

Formal documentation: <https://osm.etsi.org/gitlab/osm-architecture/osm-arch-doc/blob/master/05-pol.md>



# Auto Scaling & Alarms Architecture

When configuring alarms associated to scaling actions or just webhook notifications (through the VNFD), the following components interact.



- 'mon-evaluator' evaluates thresholds related to metrics
- Policy Manager module (POL) takes actions such as auto-scaling.
- Whenever a threshold is crossed and an alarm is triggered, the notification is generated by MON and put in the Kafka bus so other components, like POL can consume them.

# Scaling Group Descriptor

scaling-group-descriptor:

- **max-instance-count: 1**  
min-instance-count: 0  
name: vdu\_autoscale  
scaling-policy:
  - **cooldown-time: 120**  
name: cpu\_util\_above\_threshold  
scaling-criteria:
    - name: cpu\_util\_above\_threshold  
scale-in-relational-operation: LT  
scale-in-threshold: 10  
**scale-out-relational-operation: GT**  
**scale-out-threshold: 60**  
**vnf-monitoring-param-ref: agw\_cpu\_util**
- scaling-type: automatic  
threshold-time: 10
- vdu:
  - count: 1  
**vdu-id-ref: magma-agw-vdu**

The scaling descriptor is part of a VNFD. Like the example shows, it mainly specifies:

- An existing metric to be monitored, which should be pre-defined in the monitoring-param list (vnf-monitoring-param-ref).
- The thresholds to monitor (scale-in/out-threshold)
- The minimum and maximum amount of scaled instances to produce.
- The minimum time it should pass between scaling operations (cooldown-time)
- The VDU to be scaled (vdu-id-ref) and the amount of instances to scale per event (count)



# Alarm Descriptor

```
- alarm:
  - actions:
    alarm:
      - url: https://webhook.site/5706da10-04a0-4ab0-819b-cb524f71a367
    alarm-id: cpu-above-threshold
    operation: GT
    value: 80
    vnf-monitoring-param-ref: agw_cpu_util
```

Alarms based on metric thresholds can be sent to webhooks. The alarm descriptor is also part of a VNFD. Like the example shows, it mainly specifies:

- An existing metric to be monitored, which should be pre-defined in the monitoring-param list (vnf-monitoring-param-ref).
- The thresholds to monitor (alarm-threshold)
- The web hook to be invoked (url)



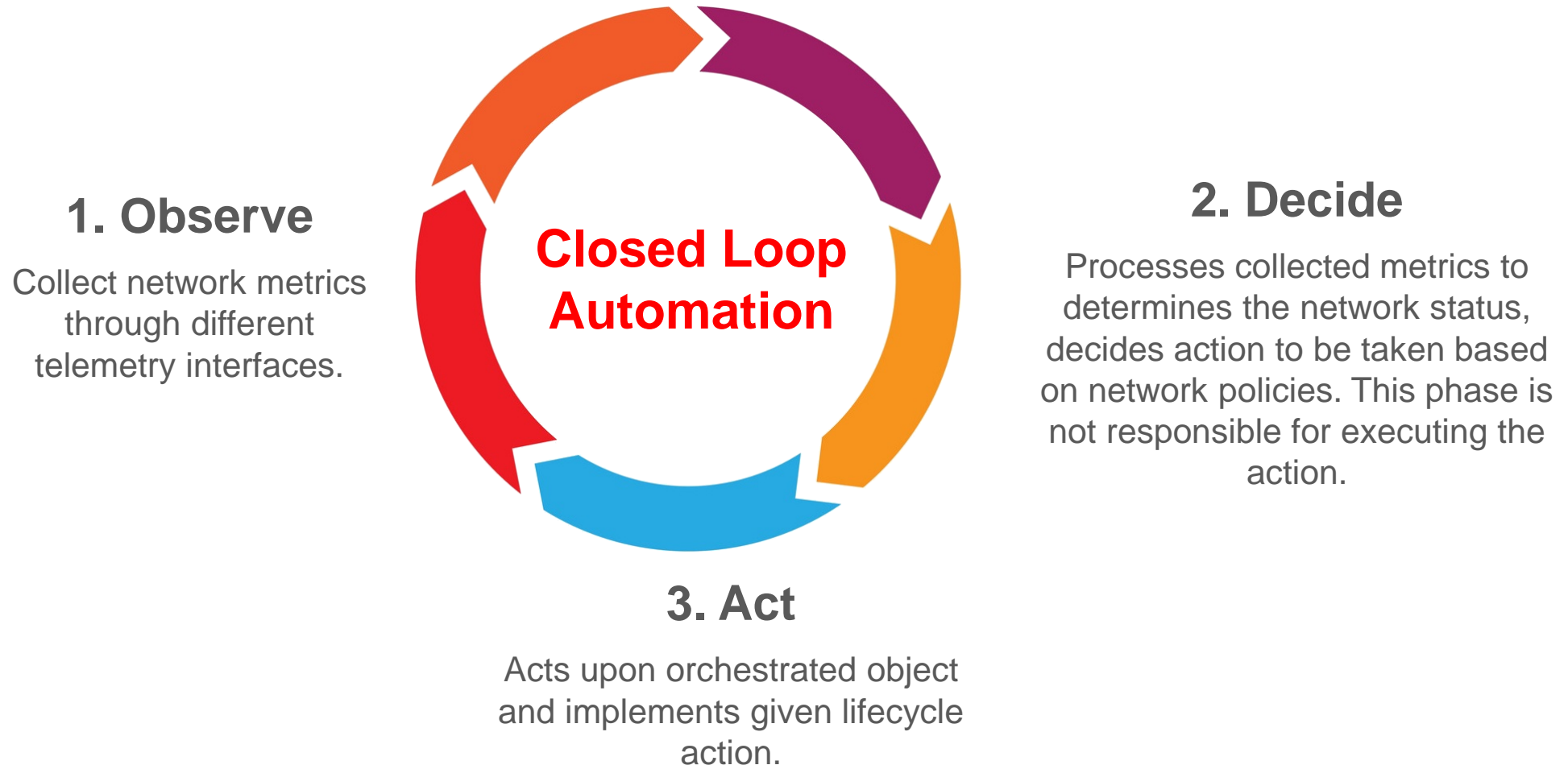
Open Source  
**MANO**

# **New Proposals**

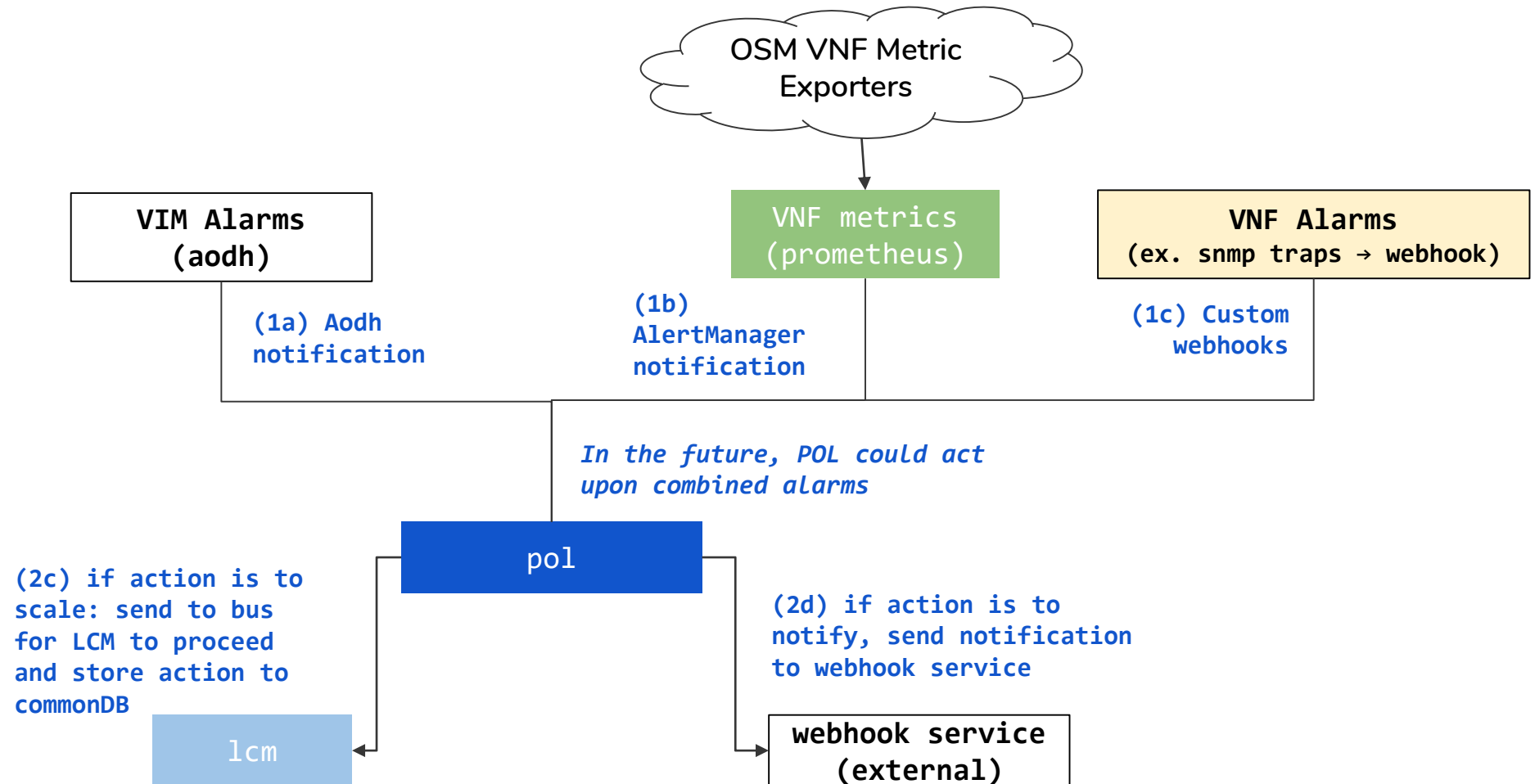
OSM Service Assurance

# Future Vision - Closed Loop Automation

Closed-loop automation powers autonomous networks.



# Auto-Scaling & Alarms – New Architecture



# Other Features in Roadmap

- Automatically monitor auto scaled VDU
- Setting threshold on correlated metrics (multiple metrics)
- Move away from threshold to ML based anomaly detection



Open Source  
**MANO**

# Hands-on!

VNF Monitoring

# Let's play with metrics and (auto)dashboards!

- We will stress the VDU of AGW VNF
- Check the AGW VM IP in the VIM <http://172.21.247.1/>

Displaying 2 items

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	magma_slice_1_154.slice_hackfest_t_nsd_epc-1-srsLTE-vm-1	srsLTE-zmqRF_hf9	magma_slice_1_154.slice_hackfest_nsd_epc-internalS1 192.168.100.10 osm-ext 172.21.248.37	srsLTE-vm-flv-2	-	Active	nova	None	Running	2 hours, 57 minutes	Create Snapshot
<input type="checkbox"/>	magma_slice_1_154.slice_hackfest_t_nsd_epc-1-magma-agw-vm-1	magma101_hf9	sgi 192.168.239.2 magma_slice_1_154.slice_hackfest_nsd_epc-internalS1 192.168.100.254 osm-ext 172.21.248.106	magma-agw-vm-flv-19	-	Active	nova	None	Running	2 hours, 58 minutes	Create Snapshot

Displaying 2 items



This is your IP

# Let's play with metrics and (auto)dashboards!

- Login to AGW VM from your OSM command line

```
$ ssh magma@172.21.248.106
```

- Increase CPU load with this command. Not down the process id.

```
$ yes > /dev/null &
```

- Check CPU metrics in Grafana <http://172.21.248.57:3000/>
- Observe increase in CPU load and eventually a new VDU is created through auto scaling.

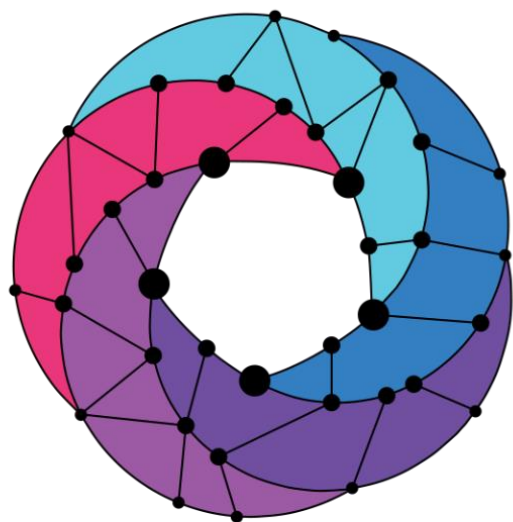


# Let's play with metrics and (auto)dashboards!

- Check webhook invoked at <https://webhook.site/> when alarm is generated.
- Now locate the IP of the process and kill it to reduce the extra CPU load

```
$ kill 3904
```

- Observe decrease in CPU load and eventually a additional VDU is deleted.



# Open Source MANO

Find us at:

[osm.etsi.org](https://osm.etsi.org)  
[osm.etsi.org/wikipub](https://osm.etsi.org/wikipub)