

Open Source
MANO

OSM-MR#9 Remote
HD4.3 Closed-Loop Operations

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

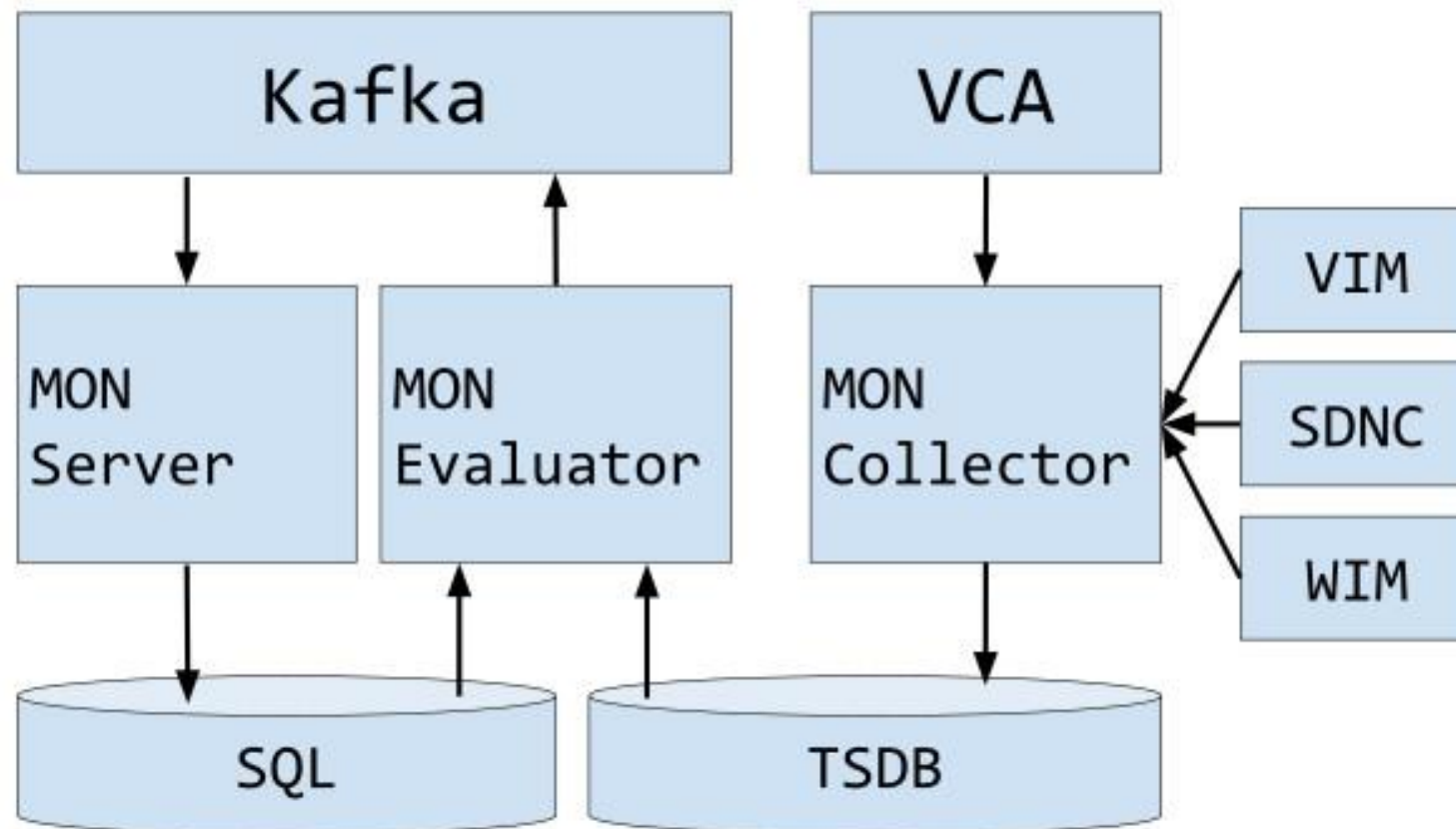
- Auto scaling allows to automatically scale VNFs with a VDU granularity and based on any available metric.
- Scaling descriptors can be included and be tied to automatic reaction to VIM/VNF metric thresholds.
- Supported metrics are both VIM and VNF metrics.

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

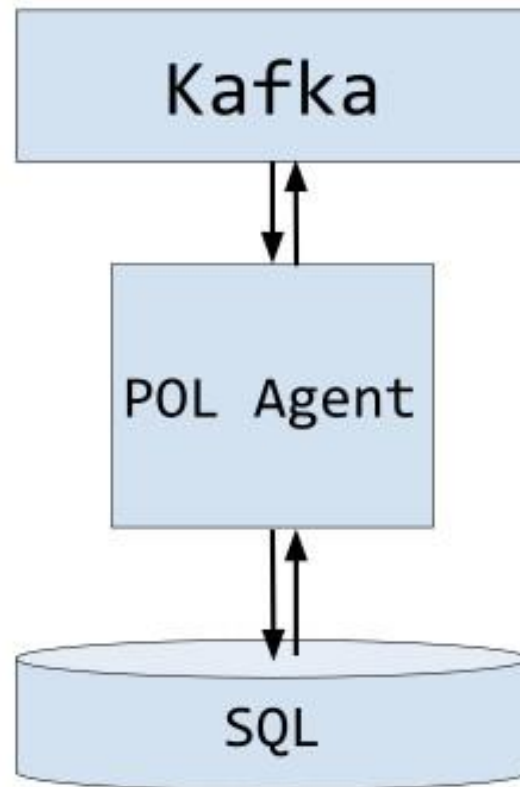
Revisiting MON Architecture

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



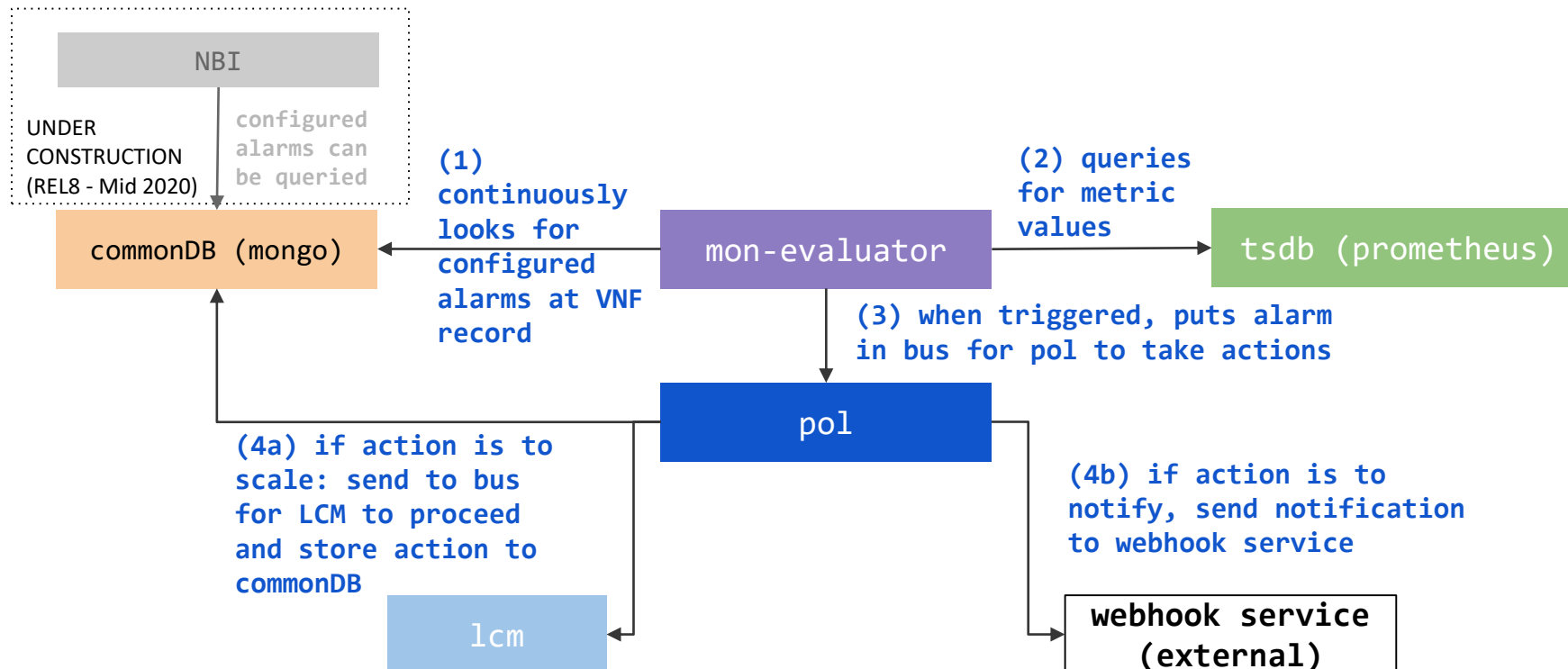
POL Architecture

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)



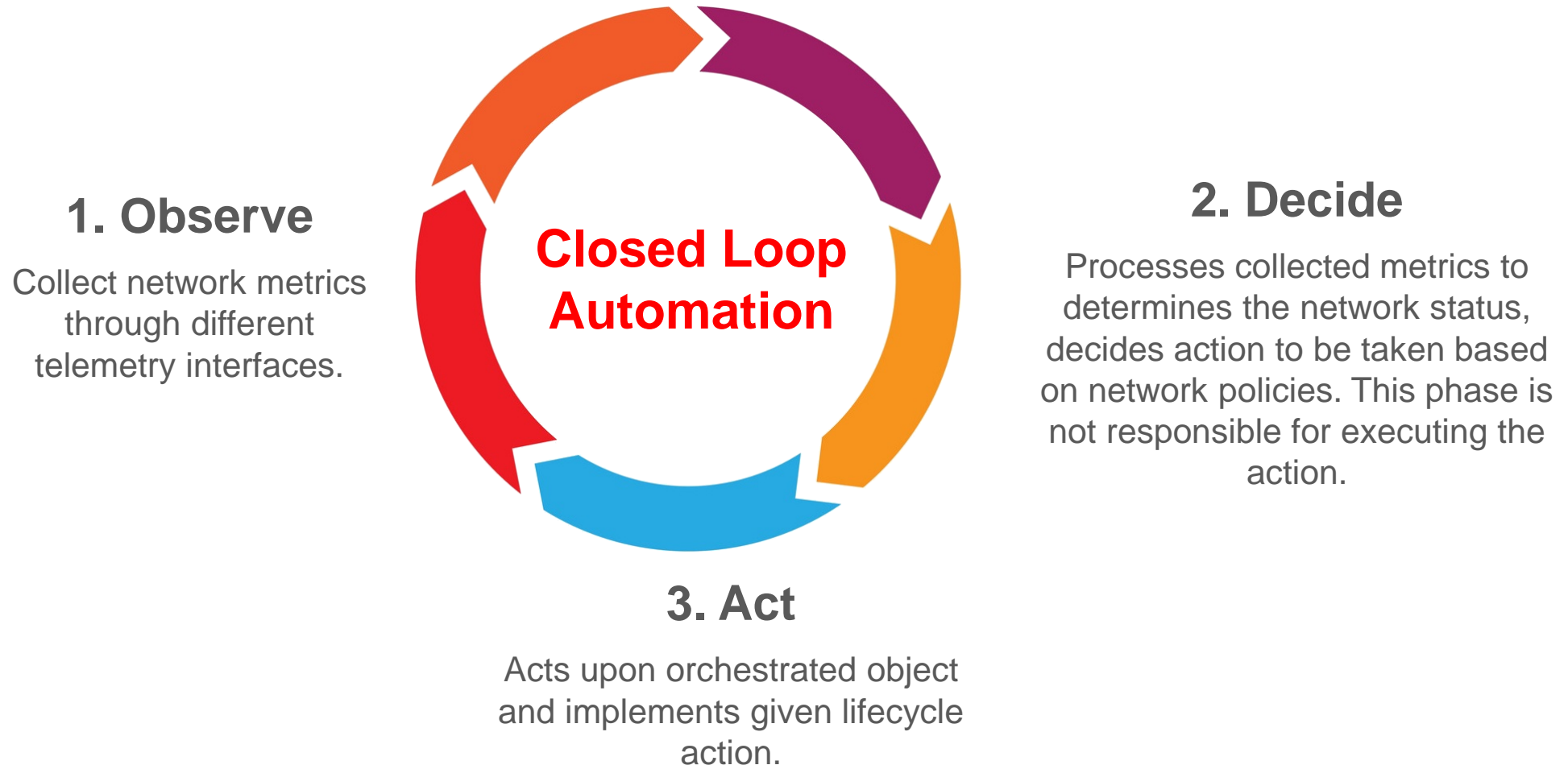
Open Source
MANO

New Proposals

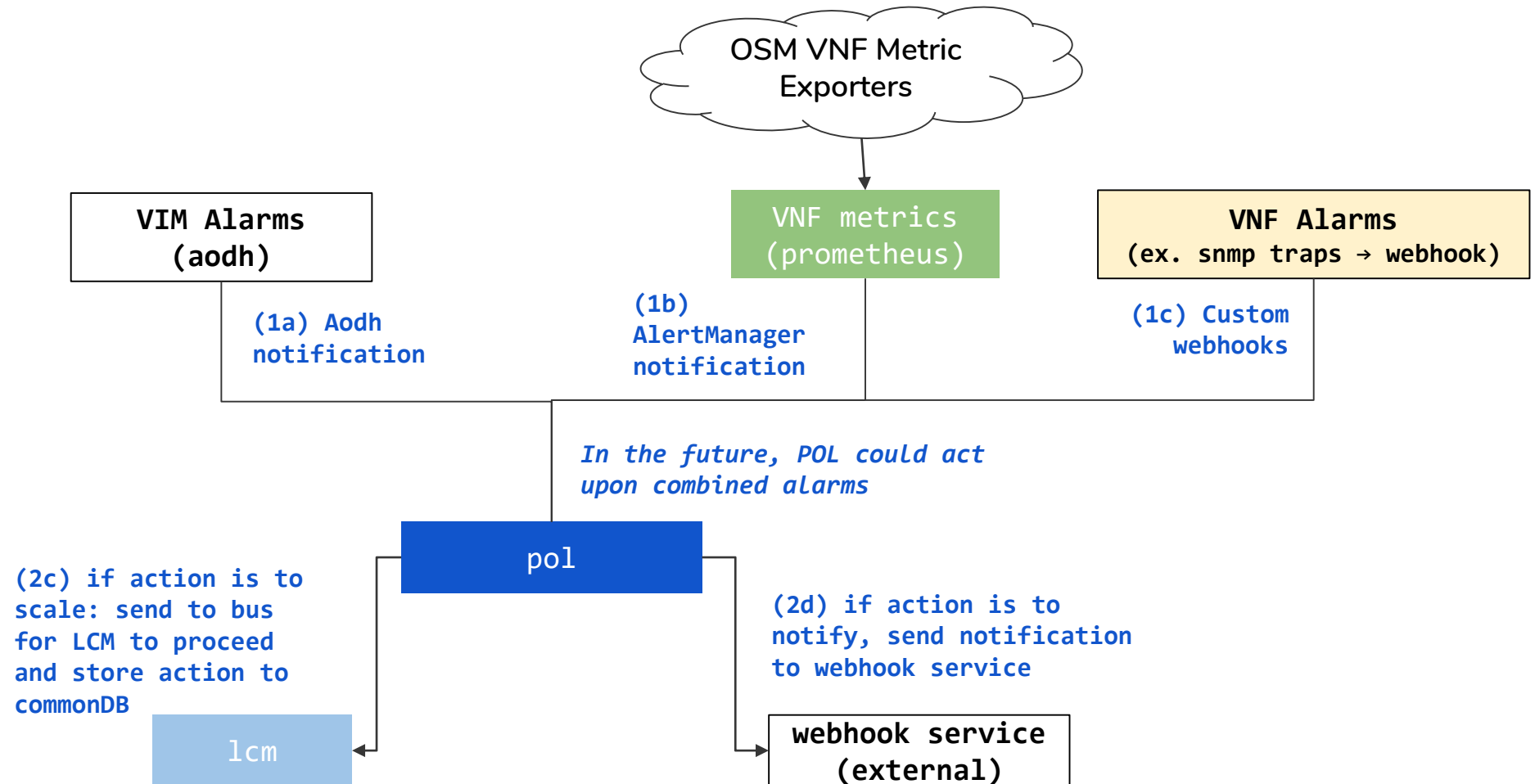
OSM Service Assurance

Future Vision - Closed Loop Automation

Closed-loop automation powers autonomous networks.



Auto-Scaling & Alarms – New Architecture



Other Features in Roadmap

- Move away from threshold to ML based anomaly detection
- Improved Closed Loop Operation
 - Dynamic Thresholds
 - Predictive Alerts/ Actions
 - Auto Healing
 - Setting threshold on correlated metrics (multiple metrics)



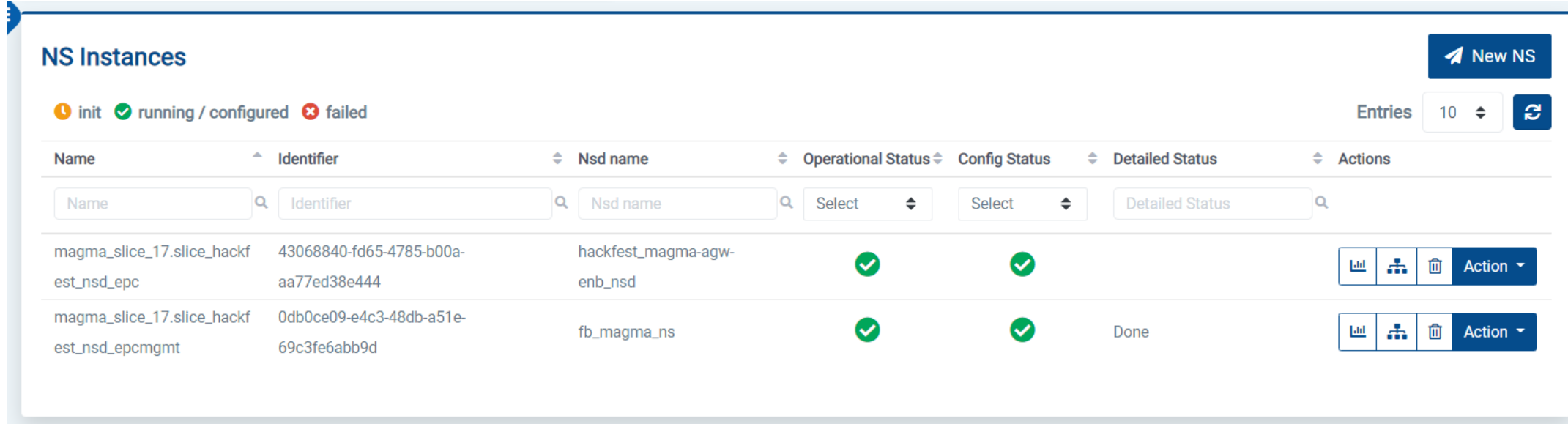
Open Source
MANO

Hands-on!







VNF Monitoring

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

- We will use slice created in previous session and stress the VDU of AGW VNF



The screenshot shows the 'NS Instances' management interface. At the top right is a 'New NS' button. Below the title, there are status filters: 'init' (orange circle), 'running / configured' (green checkmark), and 'failed' (red cross). To the right of these filters is an 'Entries' dropdown set to '10' and a refresh icon. The main table has columns: Name, Identifier, Nsd name, Operational Status, Config Status, Detailed Status, and Actions. Each column has a search input field below it. The table contains two rows of data, both with green checkmarks in the 'Operational Status' and 'Config Status' columns. The first row's 'Detailed Status' is empty, while the second row's is 'Done'. Each row has an 'Action' button with a dropdown menu.

Name	Identifier	Nsd name	Operational Status	Config Status	Detailed Status	Actions
magma_slice_17.slice_hackf est_nsd_epc	43068840-fd65-4785-b00a- aa77ed38e444	hackfest_magma-agw- enb_nsd	✓	✓		   Action
magma_slice_17.slice_hackf est_nsd_epcmgmt	0db0ce09-e4c3-48db-a51e- 69c3fe6abb9d	fb_magma_ns	✓	✓	Done	   Action

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

- 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_17.slice_hackfest_nsd_epc-MagmaAGWsrsLTE-srsLTE-vdu-1	srsLTEzmqrF_hf9	magma_slice_17.slice_hackfest_nsd_epc-internalS1 192.168.100.10 osm-ext 172.21.248.49	srsLTE-vdu-flv	-	Active	nova	None	Running	2 hours, 11 minutes	Create Snapshot
<input type="checkbox"/>	magma_slice_17.slice_hackfest_nsd_epc-MagmaAGWsrsLTE-magma-agw-vdu-1	magma101_hfmr9	magma_slice_17.slice_hackfest_nsd_epc-internalS1 192.168.100.254 sgi 192.168.239.15 osm-ext 172.21.248.14	magma-agw-vdu-flv	-	Active	nova	None	Running	2 hours, 11 minutes	Create Snapshot

Displaying 2 items

This is your IP

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



- Login to AGW VM from management VM (172.21.248.4) command line.

```
$ ssh magma@172.21.248.14
```

Note- Password is same as the user name i.e. magma

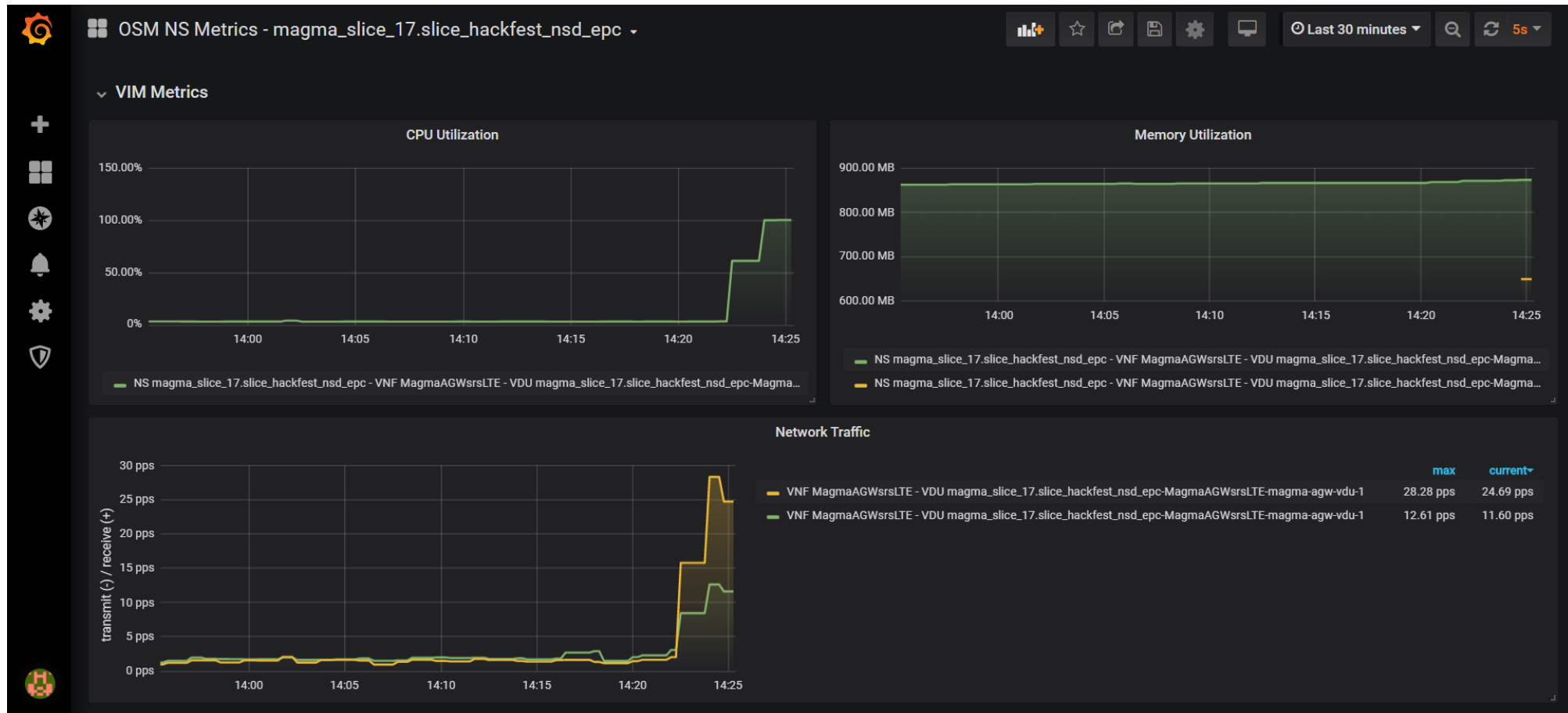
- Increase CPU load with this command. Not down the process id.

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

- Check CPU metrics in Grafana <http://172.21.248.xx:3000/>
- Observe increase in CPU load and eventually a new VDU is created through auto scaling.

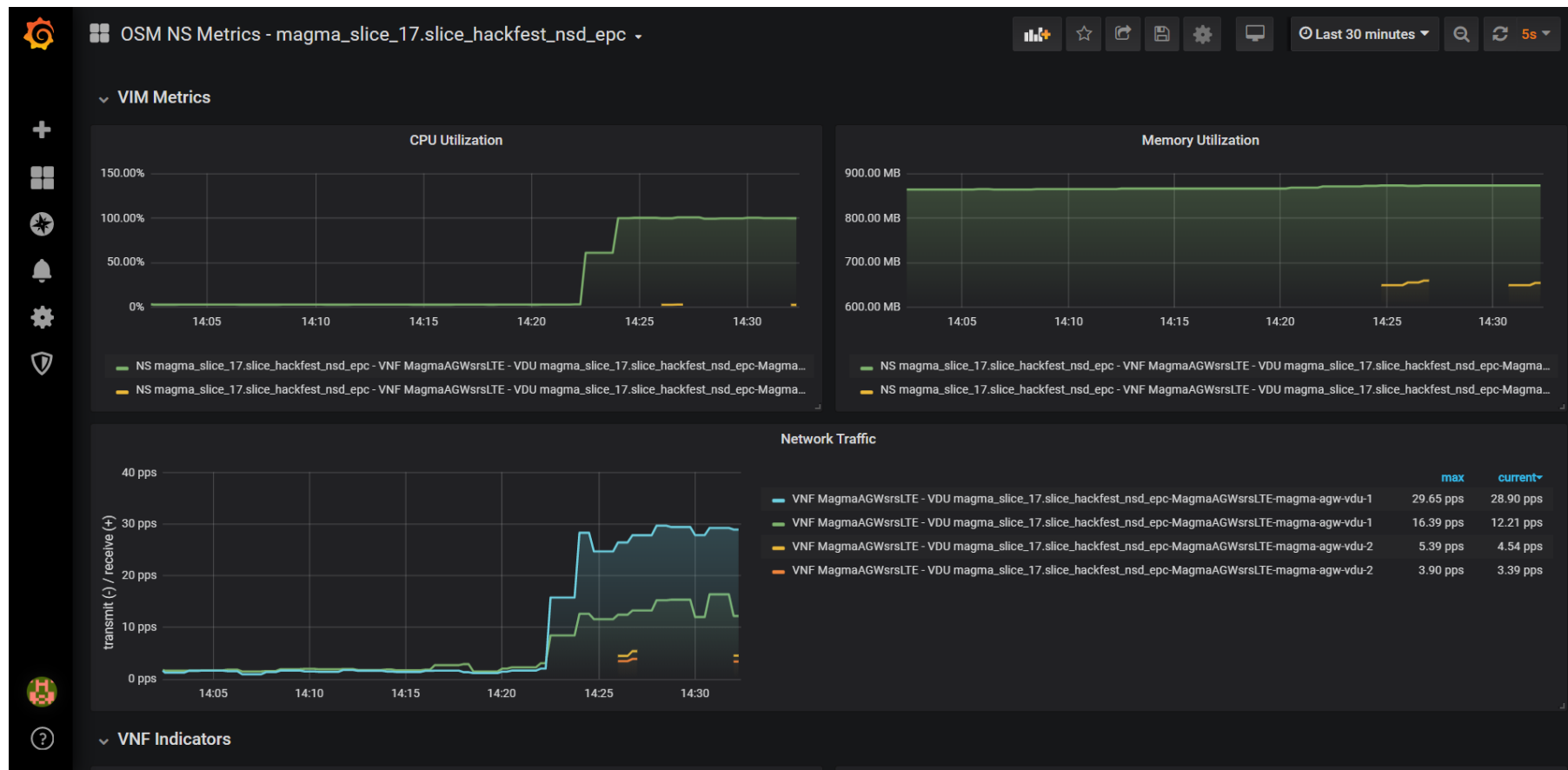
Let's play with metrics and (auto)dashboards!

- Increase in load visible in Grafana



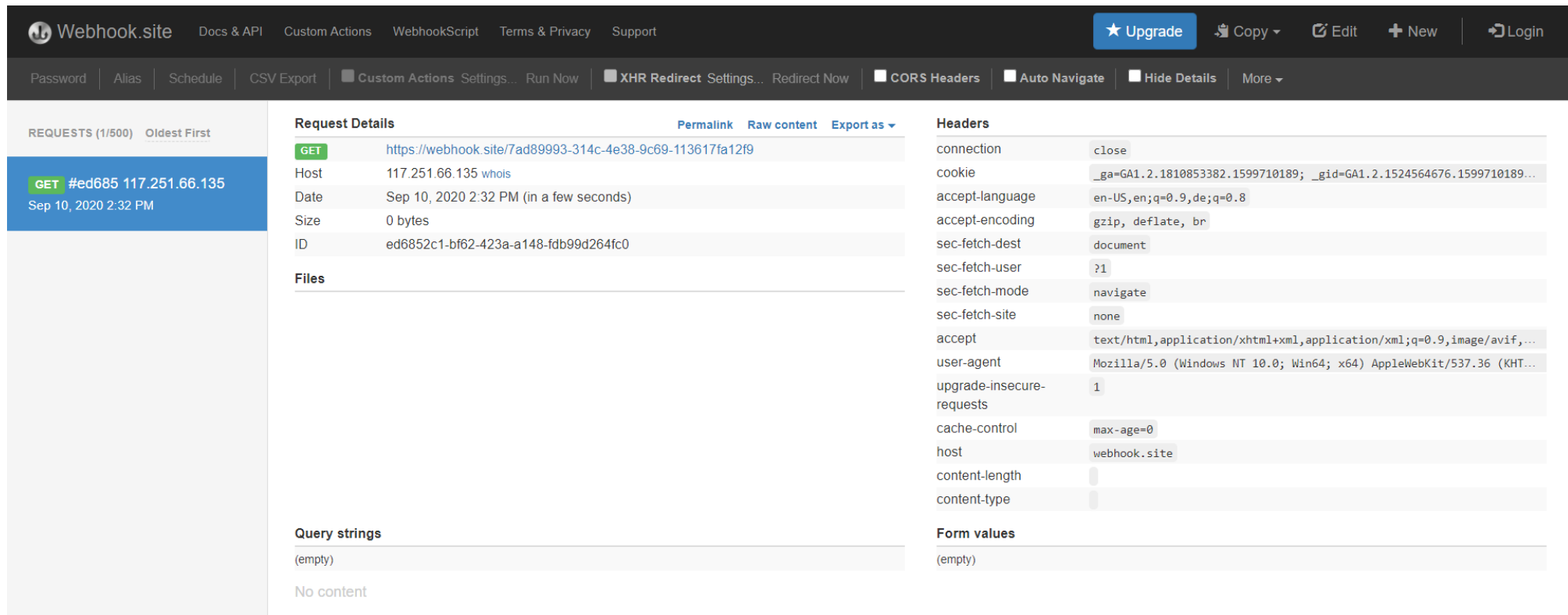
Let's play with metrics and (auto)dashboards!

- Metrics collection from scaled out VDU is also visible after sometime.



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

- Check webhook invoked at <https://webhook.site/> when alarm is generated.



The screenshot displays the Webhook.site web application. The top navigation bar includes links for Docs & API, Custom Actions, WebhookScript, Terms & Privacy, and Support, along with an Upgrade button and utility icons for Copy, Edit, New, and Login. A secondary bar contains various settings like Password, Alias, Schedule, CSV Export, Custom Actions, XHR Redirect, CORS Headers, Auto Navigate, and Hide Details. The main interface is divided into three panels. The left panel, titled 'REQUESTS (1/500)', shows a list of requests with the most recent one highlighted: a GET request from 117.251.66.135 on Sep 10, 2020 at 2:32 PM. The middle panel, 'Request Details', provides information for this specific request, including the URL (https://webhook.site/7ad89993-314c-4e38-9c69-113617fa12f9), Host (117.251.66.135), Date (Sep 10, 2020 2:32 PM), Size (0 bytes), and ID (ed6852c1-bf62-423a-a148-fdb99d264fc0). It also has sections for Files, Query strings (empty), and Headers. The right panel, 'Headers', lists various request headers such as connection (close), cookie, accept-language, accept-encoding, sec-fetch-dest, sec-fetch-user, sec-fetch-mode, sec-fetch-site, accept, user-agent, upgrade-insecure-requests, cache-control, host, content-length, and content-type.

Request	Method	Host	Date	Size	ID
#ed685	GET	117.251.66.135	Sep 10, 2020 2:32 PM	0 bytes	ed6852c1-bf62-423a-a148-fdb99d264fc0

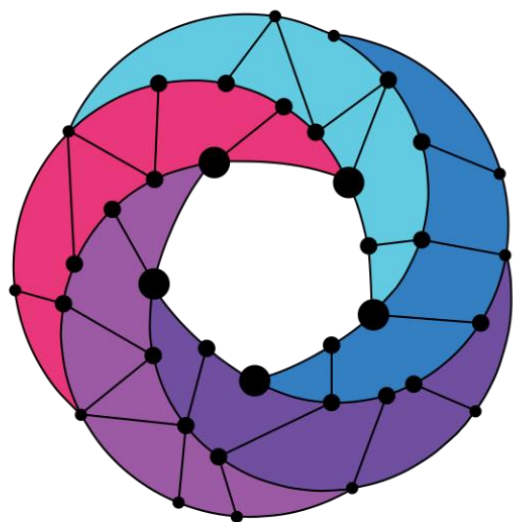
Header	Value
connection	close
cookie	_ga=GA1.2.1810853382.1599710189; _gid=GA1.2.1524564676.1599710189...
accept-language	en-US,en;q=0.9,de;q=0.8
accept-encoding	gzip, deflate, br
sec-fetch-dest	document
sec-fetch-user	?1
sec-fetch-mode	navigate
sec-fetch-site	none
accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,...
user-agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML...
upgrade-insecure-requests	1
cache-control	max-age=0
host	webhook.site
content-length	
content-type	

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

- Now locate the IP of the process and kill it to reduce the extra CPU load

```
$ kill <process-id>
```

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



Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub