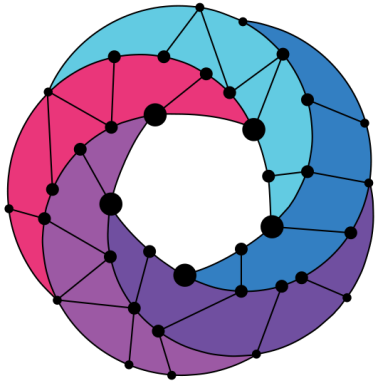


Open Source
MANO

OSM Hackfest – Session 6
Fault & Performance Management

Gianpietro Lavado (Whitestack)

© ETSI 2017

ETSI

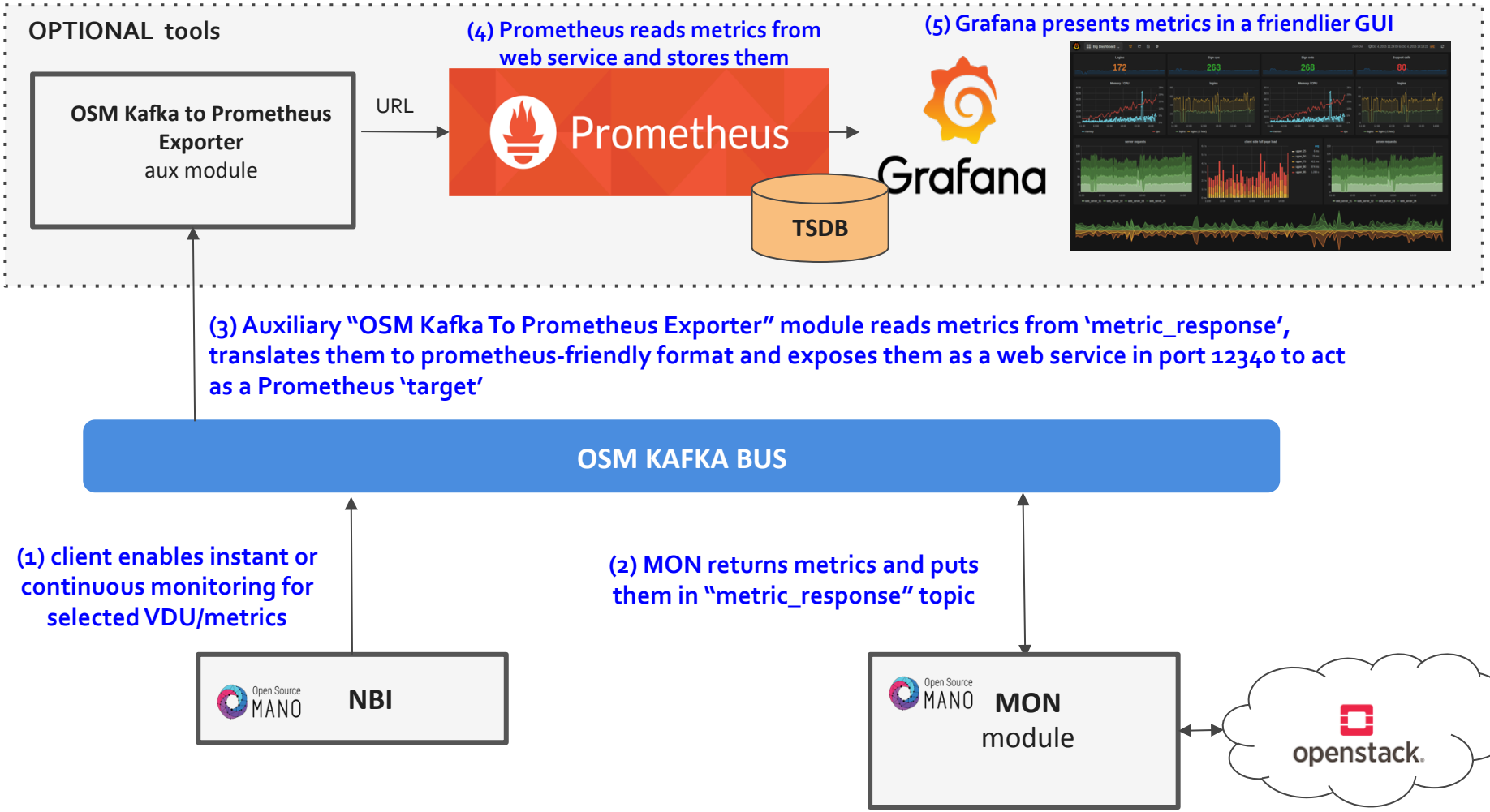
Introduction

Performance and Fault Management capabilities are in an evolving phase in OSM

Some of the features are in experimental phase and may present significant changes in the following release cycle.

© ETSI 2017

PM – What's available in Release 4.0.1?



© ETSI 2017

Basic Performance Management

- Main documentation is here:

  https://osm.etsi.org/wikipub/index.php/OSM_Performance_Management

- As of 4.0.1, the supported VIM is OpenStack with Gnocchi (others on its way!)
- Monitoring happens on a per-VDU basis.
- Supported metrics are:

  cpu_utilization, average_memory_utilization, disk_read_ops,disk_write_ops, disk_read_bytes, disk_write_bytes, packets_dropped_<nic number>, packets_received, packets_sent

© ETSI 2017

Basic Performance Management

- Exporting a metric to the bus
  - Information to the bus:

    osm ns-metric-export --ns [NS NAME] –vnf [VNF MEMBER INDEX] --vdu [VDU ID] --metric [METRIC NAME]

  - Information exported to the bus can be seen in the MON docker logs:

    docker logs [MON container id] *[--tail [XX lines]]*

  - Basic ‘continuous monitoring’ can be activated with an ‘--interval’ option:

    osm ns-metric-export --ns [NS NAME] –vnf [VNF MEMBER INDEX] --vdu [VDU ID] --metric [METRIC NAME] --interval [seconds]

  - Linux functionality can be used to send those metrics to the background (just don’t forget about them ☺)

© ETSI 2017

PM Experimental Features (1)

- We can enable a Prometheus + Grafana stack to visualize metrics.
  - Install the PM stack to your environment

    # download the installer, make it executable and…
    ./install_osm.sh -o pm_stack

    …if running Vagrant, you should forward ports 12340 and 3000

  - There is an adaptation layer provided by a 'kafka_exporter' container, which translates the metrics from the Kafka bus to a Prometheus 'target'

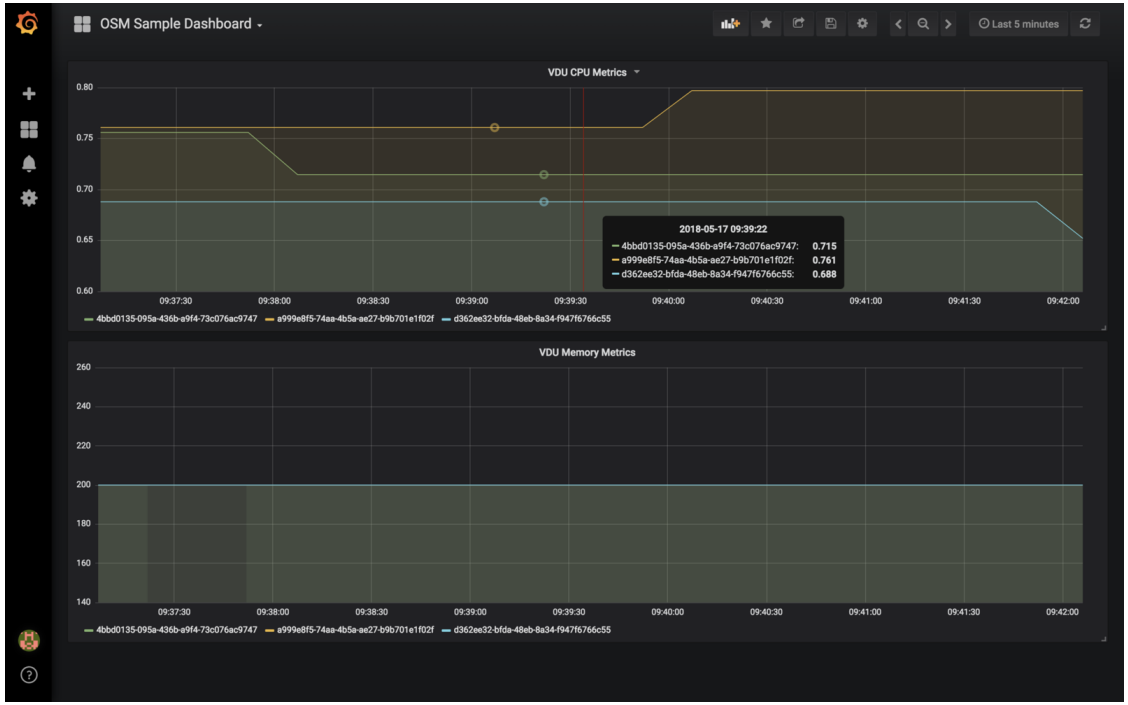
    # once it's there, go inside and make sure it sucessfully connected to Kafka
    docker exec [kafka_exporter_id] tail kafka-topic-exporter.log

  - You can then see active metrics being exported to Prometheus at http://[OSM_IP]:12340 (first time it takes some minutes)

© ETSI 2017

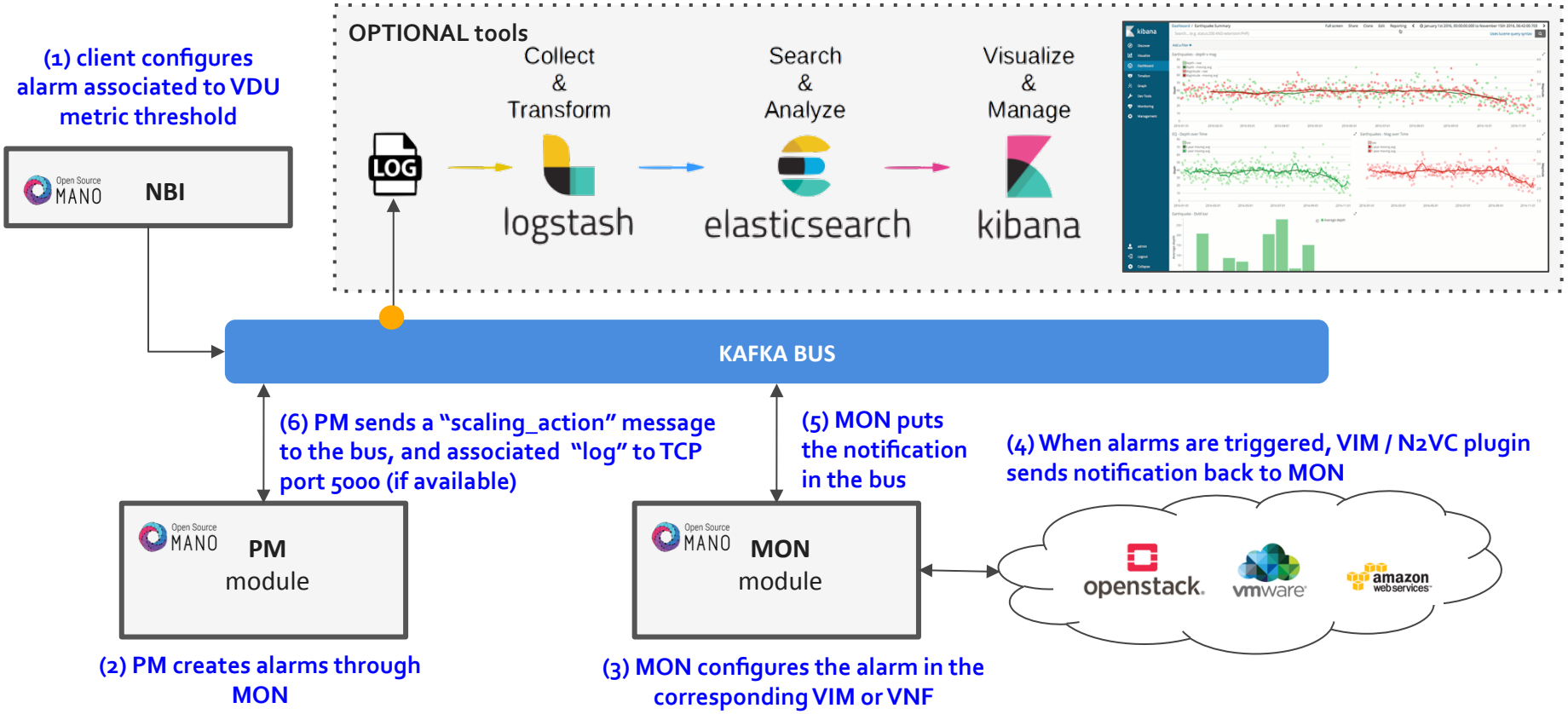
PM Experimental Features (2)

- Once you see metrics being exported for Prometheus, Grafana will be able to build graphs out of them.
- If you are monitoring CPU or Memory, you will find the associated graphs already configured at the “OSM Sample Dashboard”



© ETSI 2017

FM – What's available in Release 4.0.1?



© ETSI 2017

Basic Fault Management

- Main documentation is here:

  https://osm.etsi.org/wikipub/index.php/OSM_Fault_Management

- Regarding logs, docker containers store their own logs for now.
- Regarding alarms, as of 4.0.1:
  - the supported VIM is OpenStack with Aodh (others on its way!)
  - we can configure alarms at the VIM level, that are not associated to a specific actions (scaling, etc)
- Alarming happens on a per-VDU basis, and are controlled by the VIM components.

© ETSI 2017

Basic Fault Management: Logs

- To access logs, as of 4.0.1, it depends on the container.
  - LCM logs
    - Path /app/log/lcm.log
  - RO logs
    - Path /var/log/osm/openmano.log
  - VCA logs
    - Run ‘juju debug-logs’ from the host
  - MON/PM logs
    - docker logs [MON/PM id]

© ETSI 2017

Basic Fault Management: Alarms

- Creating an alarm
  - Creating an alarm through the client:

    osm ns-alarm-create --ns [NS NAME] –vnf [VNF MEMBER INDEX] --vdu [VDU ID] --metric [METRIC NAME] --severity [SEVERITY LEVEL] --threshold_value [VALUE] --threshold_operator [GT|LT] --statistic [AGGREGATOR, ex. AVERAGE] [ALARM NAME]

  - This will configure an alarm at OpenStack, which can be seen with the following command:

    openstack alarm list
    openstack alarm show [ID]

  - Once the alarm triggers, it will notify MON about it, and a log will be shown in that container.

© ETSI 2017

FM Experimental Features (1)

- We can enable a ELK stack stack to visualize logs and kafka events.
  - Install the ELK stack to your environment

    # download the installer, make it executable and…
    ./install_osm.sh –o elk_stack

    …if running Vagrant, you should forward port 5601 (Kibana)

  - Make sure a default pattern was created, otherwiser you would have to create one through Kibana or by using the following script:

```
curl -f -XPOST -H "Content-Type: application/json" -H "kbn-xsrf: anything" \
      "http://localhost:5601/api/saved_objects/index-pattern/logstash-*" \
      -d"{\"attributes\":{\"title\":\"logstash-*\",\"timeFieldName\":\"@timestamp\"}}”

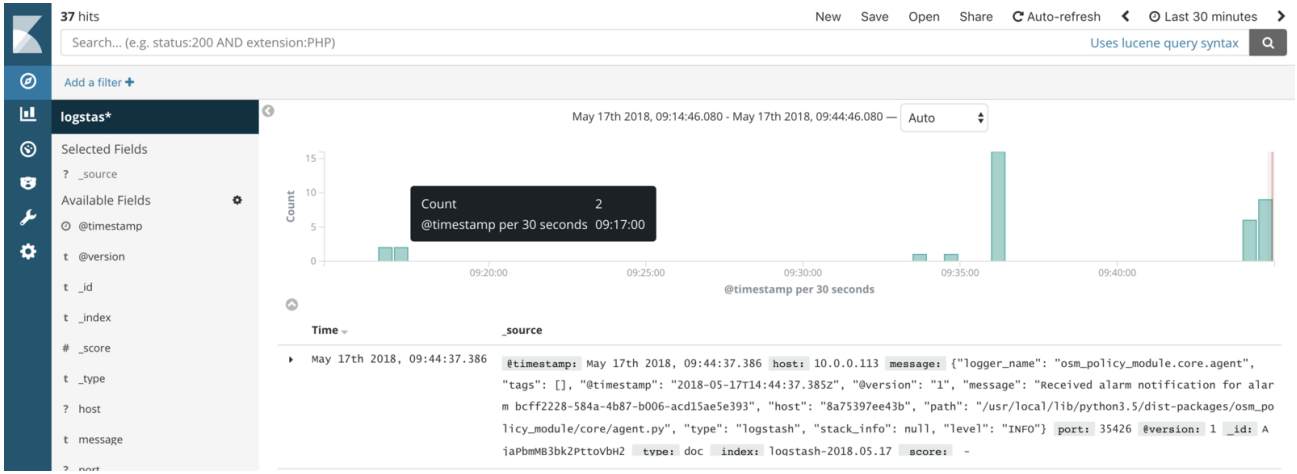
curl -XPOST -H "Content-Type: application/json" -H "kbn-xsrf: anything" \
      "http://localhost:5601/api/kibana/settings/defaultIndex" \
      -d"{\"value\":\"logstash-*\"}"
```

© ETSI 2017

FM Experimental Features (2)

- By default, Logstash will gather:
  - Any log sent to TCP port 5000
    - You can enable Policy Manager to send logs with:

      docker service update --env-add LOGSTASH_URI=logstash:5000 osm_pm
  - Kafka topic: ‘alarm_response’
- Most alarm-related activity will appear at Kibana

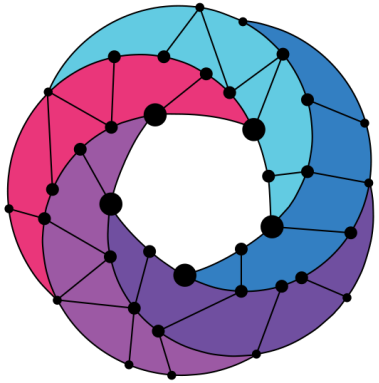


© ETSI 2017

The near future…

- PM
  - OSM to have its own TSDB to store metrics selectively, in a more VNF-aware fashion and without the need of adaptation layers.
  - This also introduces historical metrics while minimizing impact to the bus.
- FM
  - Centralize logs and notifications
  - Every log to be exported to ELK (or other compatible systems)

© ETSI 2017



Open Source
MANO

Find us at:

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

© ETSI 2017

ETSI