

Open Source  
**MANO**

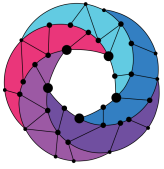
# **OSM Release TEN**

## **Release Notes**

Open Source MANO  
*Technical Steering Committee*  
June 2021

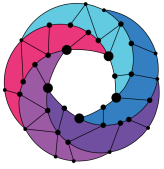
ETSI  
06921 Sophia Antipolis CEDEX, France  
Tel +33 4 92 94 42 00  
info@etsi.org  
www.etsi.org





## Authors

<b>Name</b>	<b>Organization</b>
Alejandro García	Whitestack
Andy Reid	BT
Antonio Marsico	BT
Atul Agarwal	Altran
Barath Kumar R.	Tata ELXSI
David García	Canonical
Eduardo Sousa	Canonical
Fabian Bravo	Whitestack
Francisco-Javier Ramón	Telefónica
Francisco Rodriguez	Indra
Gerardo García de Blas	Telefónica
Gianpietro Lavado	Whitestack
Isabel Lloret	Indra
José Miguel Guzmán	Whitestack
Mark Beierl	Canonical
Mark Shuttleworth	Canonical
Pål Grønsund	Telenor
Preethika P	Tata ELXSI
Ramesh Ramanathan	Tata ELXSI
Silvia Almagia	ETSI
Subhankar Pal	Altran
Wajehaa Hamid	Canonical



# Content

[Introduction](#)

[New Features in Release TEN](#)

[Scale-in/Scale-out functionality](#)

[Support for VNFs deployed in Microsoft Azure](#)

[Juju operational dashboard](#)

[Distributed VCA](#)

[Scaling for native Kubernetes charms](#)

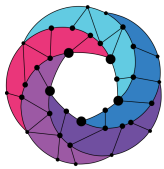
[Subscription Management](#)

[Enhanced alarm management](#)

[Improved release process](#)

[Conclusions](#)

[To go further](#)



## Introduction

We are proud to announce Open Source MANO Release TEN!

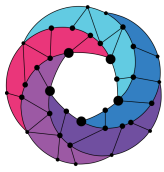
Yes, OSM v10.0 is available, which in fact is the 11th release counting from the initial Release ZERO, [announced 5 years ago](#).

As the number of OSM solutions running in production networks keeps growing, the focus of the project has shifted from implementing the vision, to orchestrating commercial VNFs and CNFs. So in this release, an important number of optimizations have been implemented, based on experiences collected from real deployments. That feedback loop is extremely important to keep alignment with real world requirements, and a high priority for the OSM Community.

OSM is also putting a lot of focus on the integration to existing systems. The positioning of ETSI NFV-SOL005 as the North-Bound API to integrate OSS to the NFV orchestrator provides a standardized mechanism that protects the investment of the integration efforts.

On the other side, and in order to adapt to edge computing scenarios, we are working on distributing our components, enabling fully decentralized deployments.

Since OSM was launched, the community has grown from 30 initial members to over 145 organizations today, without which it would have been impossible to get to this point. Many thanks to all these organizations, OSM developers and users, participating directly in building OSM!

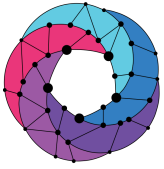


## New Features in Release TEN

Release TEN brings a set of new features:

- Distributed VNF Configuration and Abstraction, allowing to implement VNF management in a distributed fashion, which gives better resiliency for different failure scenarios and enables Geo-Redundant deployments.
- Scaling for Kubernetes Applications Using Juju, which improves scalability in Kubernetes deployments.
- Improved Scale-in/Scale-out functionality, allowing operators to trigger scaling operations from the Graphical User Interface in addition to the Command Line.
- A Juju operational dashboard, fully integrated into the Graphical User Interface, providing operators with a more interactive view about the status of day-1 and day-2 operations
- Subscription Management to allow Operation Support Systems (OSS) to subscribe to Life Cycle Management events by using a library (or Command Line Interface), and facilitate the integration of 3rd party systems.
- Enhanced alarm management allows users to have better visibility and interaction with the alarms registered in OSM.
- Support for Microsoft Azure, which gives additional public cloud options for deploying Network Functions
- Reduced footprint and improved release process, relying on Python packages with locked dependency versions and greatly reduced image sizes.

The following sections describe in further detail these major new features coming with Release TEN.



## Scale-in/Scale-out functionality

In previous releases, manual Scale-In/Scale-Out commands could only be triggered from the Command Line Interface. Now Scaling operations can also be triggered from the NG-UI.

After Launching the NS Instances. Go to the **NS Instances** page. Click **Actions** of the NS Instance row. From the list select **Manual Scaling** link. Then a new pop-up page will open if the scaling-aspect key is found in the VNFD package.

A screenshot of a 'Manual Scaling' pop-up window. The window has a title bar with the text 'Manual Scaling' and a close button (red circle with an 'x'). Below the title bar, there is a note: 'Mandatory fields are marked with an asterisk (\*)'. There are three dropdown menus, each with an asterisk: 'Select Member VNF Index \*', 'Select scaling-aspect \*', and 'Select Scale Type \*'. At the bottom right of the form, there are two buttons: a red 'Cancel' button and a blue 'Apply' button.

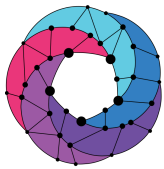
In the above figure pop-up there will be an option to select appropriate details of Member VNF Index, For the selected member VNF Index the scaling-aspect list will be displayed select any one option. Select the option whether the scaling-type is going to perform the "**Scale-In/Scale-Out**" and click the "**Apply**" button to trigger scaling action, Once triggered, the pop-up window will be closed and the page is redirected to "History of operations" page for this NS.

## Support for VNFs deployed in Microsoft Azure

Release TEN updates the support of Azure VIM to include the latest API calls from Azure, the use of Azure AD service principals, the parallelization of the creation of resources (virtual machines, networks, disks, etc.) and improved logging. This guarantees future-proof compatibility of OSM with Azure.

## Juju operational dashboard

Juju is a key technology within OSM, allowing it to operate network functions (ie., VNFs, CNFs or PNFs) during their entire life cycle. Until now, there was a powerful command set to



monitor the status of Charms, responsible for running operational primitives on the VNFs, in this release, we are taking Juju to the User Interface, by providing a Dashboard that allows operators to understand exactly what is going on, with no need to use the command line:

- Juju models a charm of a VNF to an Application.
- Juju shows status of deployed App and units in it.
- Juju lists status of operational status, scale, version, etc.
- All information are limited to be accessed by CLI.

Juju's Operational view is exposed to NG-UI for better readability and user experience, avoiding interacting with the terminal, and providing instantaneous status after executing an action. Easy click button to execute Day-2 operations on the charm directly.

## Distributed VCA

The VCA (Juju controller) has been a core component of OSM since Release ONE. Combined with the use of charms, it allows for the operation and integration of Network Functions of different natures: physical, virtual, and containerized. In previous releases, OSM could only talk with one VCA, but that has changed.

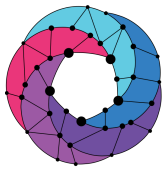
In this release, OSM has taken a step further including the VCA as a first class citizen, allowing a user with admin privileges to manage VCAs in a project. As a result, OSM is now able to talk to multiple VCAs, and they can be distributed across different points of presence (PoP), meaning charms deployed by one OSM cluster can be managed by different Juju controllers, providing a more scalable, robust architecture with the separation of fault domains.

A fault domain is a set of functionality that has a shared point of failure. In the case where there is only one VCA, a single fault would result in the loss of all day 2 operations. With multiple, distributed VCAs the loss of any one does not impact the functionality provided by the others. This lends itself to advanced use cases requiring greater autonomy (less cross-dependencies) and reduced latency such as edge deployments, or geographical redundancy.

## Scaling for Kubernetes Applications Using Juju

Containerized Network Functions were added in OSM Release SEVEN. Since then, OSM can model the deployment of Kubernetes workloads using Juju bundles or Helm.

The importance of scaling in production environments is undeniable. Scaling provides the ability to adapt the workload to the demand. As usage increases, applications can scale-out



to provide additional capacity and release their resources, or scale in, as usage returns to the baseline.

In Release TEN, we have added the ability for on-demand horizontal scaling of Juju bundle-based Kubernetes workloads. Juju bundles are collections of charms that link applications together, which allow administrators to deploy large, complex systems from one file, where all of the configuration and integrations are pre-defined. Charms are the best way of implementing customized day-2 operations for containerized workloads in Kubernetes.

For this feature, we have added the concept of Kubernetes Deployment Unit (KDU) resource that identifies a specific application of a KDU, each of which can be scaled at run-time. The OSM user can express in the VNF descriptor the maximum and minimum number of units per KDU resource, and the number of units to be added or removed in scale-out or scale-in operations.

## Subscription Management

Release TEN comes with new tools for managing event subscription through the OSM Northbound API, based on ETSI NFV-SOL005. This feature is key to facilitate integration with third-party Operations Support Systems..

New options have been added directly in the OSM client, to allow developers to interact with the subscription API more easily. In addition, a new python library has been made available, to facilitate the integration with other systems.

By using these API calls, it is possible for a external system to subscribe for events related to:

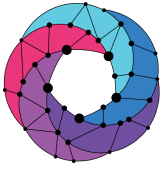
- Network Service Lifecycle Management
- Network Service Descriptor Management
- VNF Descriptor Management

## Enhanced alarm management

Release TEN has implemented ETSI NFV-SOL005 Fault Management interface to provide OSM users better visibility and interaction with the registered alarms. Each alarm corresponds to a change in VNF's or NS's operation conditions (like high CPU utilization) that requires an action to be performed (for e.g. auto-scale-out of VNF).

OSM users can now list the current alarms, get details of a specific alarm, update the alarm threshold and enable or disable the alarms. This feature helps OSM users to easily verify if



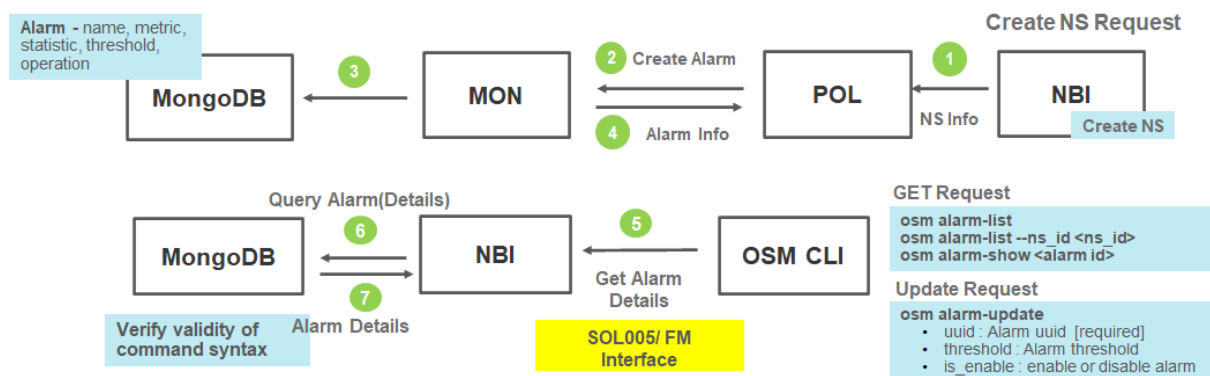


the alarms are registered correctly, and also allows them to change some of the alarm parameters after the alarm is registered.

With this feature, OSM users will get role based access to the following commands using OSM client.

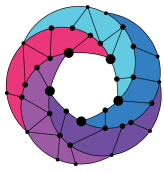
1. **osm alarm-list:** This command will list all the alarms registered in OSM and their current state.
2. **osm alarm-show:** This command will show the details of a specific alarm
3. **osm alarm-update:** This command will allow - to change the threshold of a specific alarm or change the webhook address of the alarm or enable or disable the alarm. Alarm is only triggered if the threshold criteria is met and the alarm is enabled.

The end to end message flow is shown in the figure below. Message 1 to 6 shows how a new alarm is registered when a new network service is instantiated. Message 5 to 7 shows how the alarm is accessed through the OSM CLI.



## Smaller Artifact Footprint

With Release TEN, OSM has taken a new approach to the creation and distribution of its container image artifacts. The container builds have moved away from using upstream Apt sources for python dependencies, instead using pinned versions directly from PyPI itself. This allows for full control and reproducibility of our final container images. Along with this, intermediate layers for docker images are being discarded, resulting on average in each container being 50% smaller as only the final binary layers are being published.



## Conclusions

Release TEN defines an inflection point in OSM evolution, shifting from implementing a vision, to bringing concrete efficiencies to operators. Nowadays no operator is questioning the rationale behind NFV. On the contrary, they are looking for ways to move those efficiencies further in their networks.

The redesign of some of the key modules in OSM, allows to decouple and distribute functions across the network, which together with the natural efficiencies of Kubernetes, makes MEC or OpenRAN the natural next steps for the project.

Finally, we are happy to witness operators not only conducting POCs of OSM in different regions of the world, but with OSM deployed in their production networks, which confirms the value of multi-vendor network services orchestrated by open solutions based on interoperable standards.

ETSI OSM is an open community in its fifth year of growth that welcomes the contribution of people and organisations sharing the vision of open source orchestration. If you want to know how to join us, please visit [How To Get Involved In OSM](#)

## To go further

For more information, please visit the following links:

- [Open Source MANO's user documentation](#)
- [Open Source MANO's developer's guide](#)
- [Open Source MANO's VNF Onboarding guide](#)
- [OSM Ecosystem](#)
- [OSM Members and Participants](#)