

Open Source
MANO
by ETSI

OSM installation

Gerardo García (Telefónica, OSM TSC Chair)

OSM#15

12/06/2023

Please do not run any installation now.

This is only an informative presentation.

- Details can be found in OSM user guide: <https://osm.etsi.org/docs/user-guide/latest/03-installing-osm.html>



Open Source
MANO

Search docs

TABLE OF CONTENTS

1. OSM Quickstart
2. OSM Architecture and Functions
3. Installing OSM
 - 3.1. Pre-requirements
 - 3.2. Installation procedure
 - 3.3. Other installation methods
 - 3.4. Checking your installation
 - 3.5. Installing standalone OSM Client
4. How to Set Up Virtual Infrastructure Managers (VIMs)
5. OSM Usage
6. OSM platform configuration

Other Versions v: latest

» 3. Installing OSM

[View page source](#)

3. Installing OSM

3.1. Pre-requirements

In order to install OSM, you will need, at least, a single server or VM with the following requirements:

- MINIMUM: 2 CPUs, 6 GB RAM, 40GB disk and a single interface with Internet access
- RECOMMENDED: 2 CPUs, 8 GB RAM, 40GB disk and a single interface with Internet access
- Base image:
 - Ubuntu20.04 cloud image (64-bit variant required) (<https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-amd64.img>)
 - Ubuntu20.04 server image (64-bit variant required) (<http://releases.ubuntu.com/20.04/>)

In addition, you will need a Virtual Infrastructure Manager available so that OSM can orchestrate workloads on it. The following figure illustrates OSM interaction with VIMs and the VNFs to be deployed there:

- OSM communicates with the VIM for the deployment of VNFs.
- OSM communicates with the VNFs deployed in a VIM to run day-0, day-1 and day-2 configuration primitives.



Installation of current release

- Default installation of current release (Release THIRTEEN)

```
wget https://osm-download.etsi.org/ftp/osm-13.0-thirteen/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh 2>&1 | tee osm_install_log.txt
```

On Ubuntu 20.04 (focal)

- Common options in installer:

```
-h / --help:    print this help
-y:            do not prompt for confirmation, assumes yes
-r <repo>:     use specified repository name for osm packages
-R <release>:  use specified release for osm binaries (deb packages)
-t <docker tag> specify osm docker tag (default is latest)
-D <devops path> use local devops installation path
-p <docker proxy URL> set docker proxy URL as part of docker CE configuration
--charmed:     Deploy and operate OSM with Charms on k8s
```

Demo

Other installation procedures

- Testing daily versions of OSM

- Master (current development branch)

On Ubuntu 22.04 (jammy)

```
wget https://osm.etsi.org/gitlab/osm/devops/-/raw/master/installers/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh -R testing-daily -t testing-daily -r testing -y
```

- ReleaseTWELVE

On Ubuntu 20.04 (focal)

```
wget https://osm.etsi.org/gitlab/osm/devops/-/raw/master/installers/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh -R ReleaseTWELVE-daily -t releasetwelve-daily -r testing -y
```

- Charmed OSM

On Ubuntu 20.04 (focal)

On Ubuntu 22.04 (jammy)

- <https://charmed-osm.com/>

```
wget https://osm-download.etsi.org/ftp/osm-13.0-thirteen/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh --charmed
```

Next OSM release: Release FOURTEEN (not yet available)

- **Release FOURTEEN (not yet available)**

On Ubuntu 22.04 (jammy)

```
wget https://osm-download.etsi.org/ftp/osm-14.0-  
fourteen/install_osm.sh  
chmod +x install_osm.sh  
./install_osm.sh
```

- Some differences that you will find between Release THIRTEEN and Release FOURTEEN

- OSM services are deployed with helm instead of kubectl
- New Service Assurance architecture:
 - Airflow, Alert Manager and Prometheus Push Gateway are deployed
 - POL is not deployed
 - Simplified MON is deployed (only runs Grafana dashboards)

What is done by the installer?

- Install local LXD server (required for LXD-based proxy charms)
- Install Docker CE
- Install and initialize a local Kubernetes cluster, including a CNI (Flannel), container storage (OpenEBS) and a Load Balancer (MetalLB)
- Install Juju client and juju controller
 - Includes bootstrap of juju controller to allow the deployment of Execution Environments in local LXD server and local LXD cluster
- Deploy OSM
- Install OSM client

What can be found after OSM installation?

```
$ kubectl -n osm get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
airflow-postgresql	ClusterIP	10.105.14.130	<none>	5432/TCP	4d8h
airflow-webserver	NodePort	10.97.190.241	<none>	8080:28281/TCP	4d8h
airflow-worker	ClusterIP	None	<none>	8793/TCP	4d8h
alertmanager	NodePort	10.96.97.229	<none>	9093:9093/TCP	4d8h
alertmanager-headless	ClusterIP	None	<none>	9093/TCP	4d8h
grafana	NodePort	10.102.165.200	<none>	3000:3000/TCP	4d8h
kafka	ClusterIP	None	<none>	9092/TCP	4d8h
keystone	ClusterIP	None	<none>	5000/TCP	4d8h
modeloperator	ClusterIP	10.110.127.70	<none>	17071/TCP	4d8h
mon	ClusterIP	None	<none>	8662/TCP	4d8h
mongodb-k8s	ClusterIP	10.102.39.40	<none>	27017/TCP	4d8h
mongodb-k8s-endpoints	ClusterIP	None	<none>	<none>	4d8h
mongodb-k8s-operator	ClusterIP	10.109.218.187	<none>	30666/TCP	46h
mysql	ClusterIP	None	<none>	3306/TCP	4d8h
nbi	NodePort	10.102.252.3	<none>	9999:9999/TCP	4d8h
ng-ui	NodePort	10.98.95.162	<none>	80:80/TCP	4d8h
prometheus	NodePort	10.97.21.236	<none>	9090:9091/TCP	4d8h
pushgateway-prometheus-pushgateway	ClusterIP	10.97.0.67	<none>	9091/TCP	4d8h
ro	ClusterIP	None	<none>	9090/TCP	4d8h
webhook-translator	NodePort	10.99.161.5	<none>	80:9998/TCP	4d8h
zookeeper	ClusterIP	None	<none>	2181/TCP	4d8h

What can be found after OSM installation?

```
$ kubectl -n osm get deployments
```

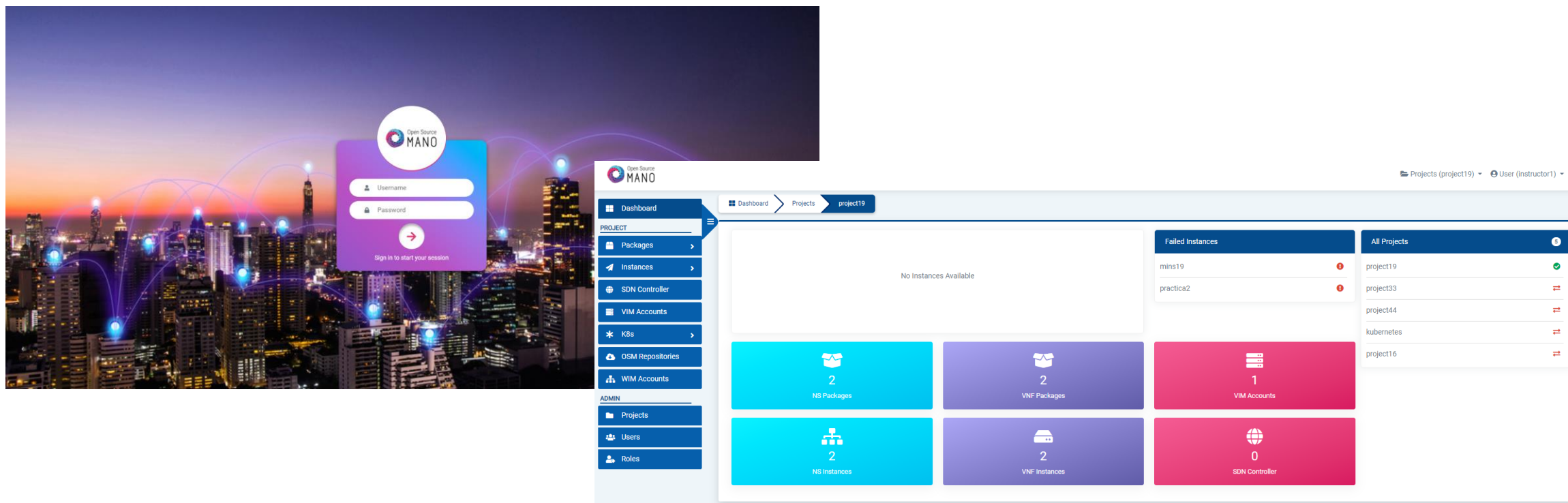
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
airflow-scheduler	1/1	1	1	4d8h
airflow-statsd	1/1	1	1	4d8h
airflow-triggerer	1/1	1	1	4d8h
airflow-webserver	1/1	1	1	4d8h
grafana	1/1	1	1	4d8h
keystone	1/1	1	1	4d8h
lcm	1/1	1	1	4d8h
modeloperator	1/1	1	1	4d8h
mon	1/1	1	1	4d8h
nbi	1/1	1	1	4d8h
ngui	1/1	1	1	4d8h
pushgateway-prometheus-pushgateway	1/1	1	1	4d8h
ro	1/1	1	1	4d8h
webhook-translator	1/1	1	1	4d8h

What can be found after OSM installation?

```
$ kubectl -n osm get statefulsets
NAME                READY   AGE
airflow-postgresql  1/1    4d8h
airflow-redis       1/1    4d8h
airflow-worker      1/1    4d8h
alertmanager        1/1    4d8h
kafka                1/1    4d8h
mongodb-k8s         1/1    4d8h
mongodb-k8s-operator 1/1    4d8h
mysql                1/1    4d8h
prometheus           1/1    4d8h
zookeeper            1/1    4d8h
```

Using your OSM installation

- You can access to the UI in the following URL (user:admin, password: admin): `http://<HOST_IP_ADDRESS>`



The image shows the Open Source MANO user interface. On the left, there is a login card with fields for Username and Password, and a 'Sign in to start your session' button. The main dashboard displays a navigation menu on the left with categories like PROJECT and ADMIN. The central area shows a 'No Instances Available' message. On the right, there are summary cards for NS Packages (2), VNF Packages (2), VIM Accounts (1), NS Instances (2), VNF Instances (2), and SDN Controller (0). A 'Failed Instances' table lists 'mins19' and 'practica2' with error icons. An 'All Projects' table lists various projects with status indicators.

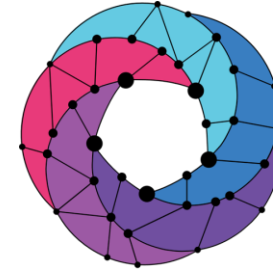
Failed Instances	
mins19	!
practica2	!

All Projects	
project19	✓
project33	!
project44	!
kubernetes	!
project16	!

Using your OSM installation

- OSM client will be available as well in the host machine. Via the OSM client, you can manage NF and NS packages, deploy NS and operate them.

```
osm --help
```



Open Source
MANO
by ETSI

Thank You!