Open Source MANO

by ETSI

OSM installation

Gerardo García (Telefónica, OSM TSC Chair)

OSM Training Seminar - SLICES

13/02/2024

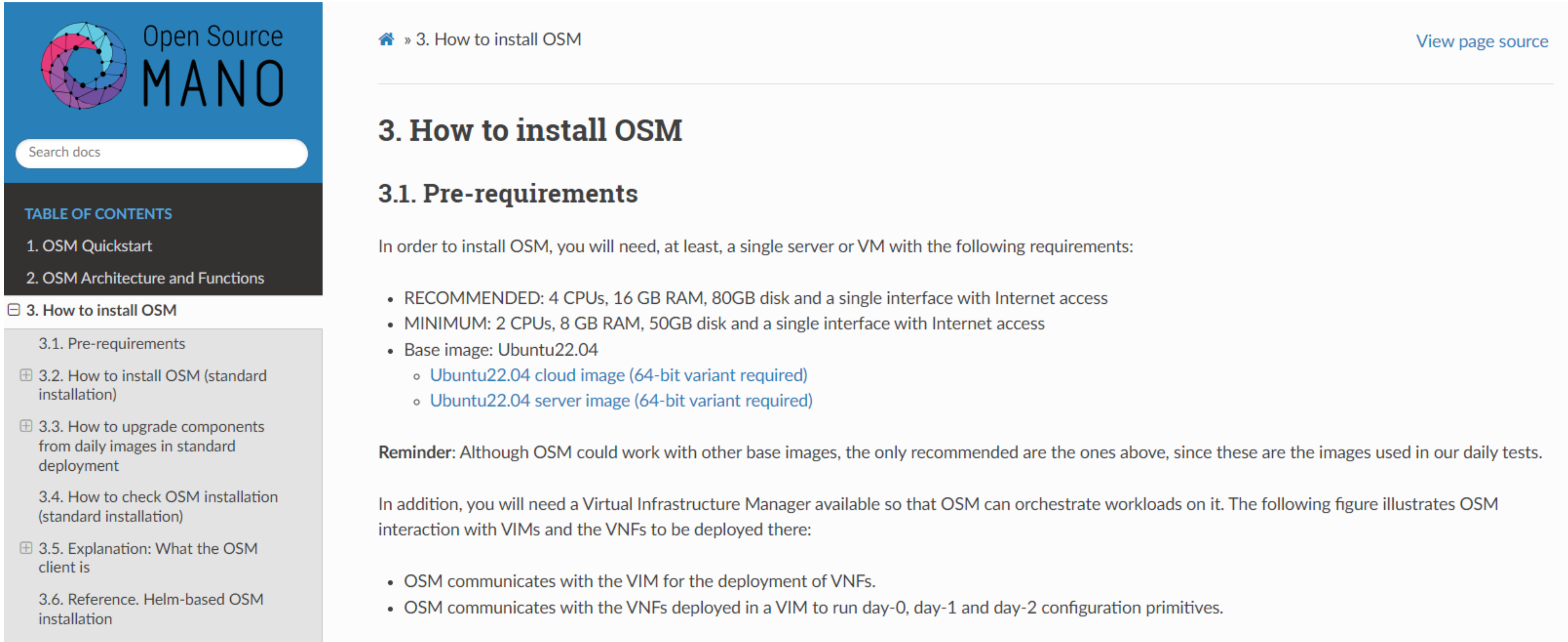
DISCLAIMER

Please do not run any installation now.

This is only an informative presentation.

OSM installation

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



3. How to install OSM

3.1. Pre-requirements

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

- RECOMMENDED: 4 CPUs, 16 GB RAM, 80GB disk and a single interface with Internet access
- MINIMUM: 2 CPUs, 8 GB RAM, 50GB disk and a single interface with Internet access
- Base image: Ubuntu22.04
  - Ubuntu22.04 cloud image (64-bit variant required)
  - Ubuntu22.04 server image (64-bit variant required)

Reminder: Although OSM could work with other base images, the only recommended are the ones above, since these are the images used in our daily tests.

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

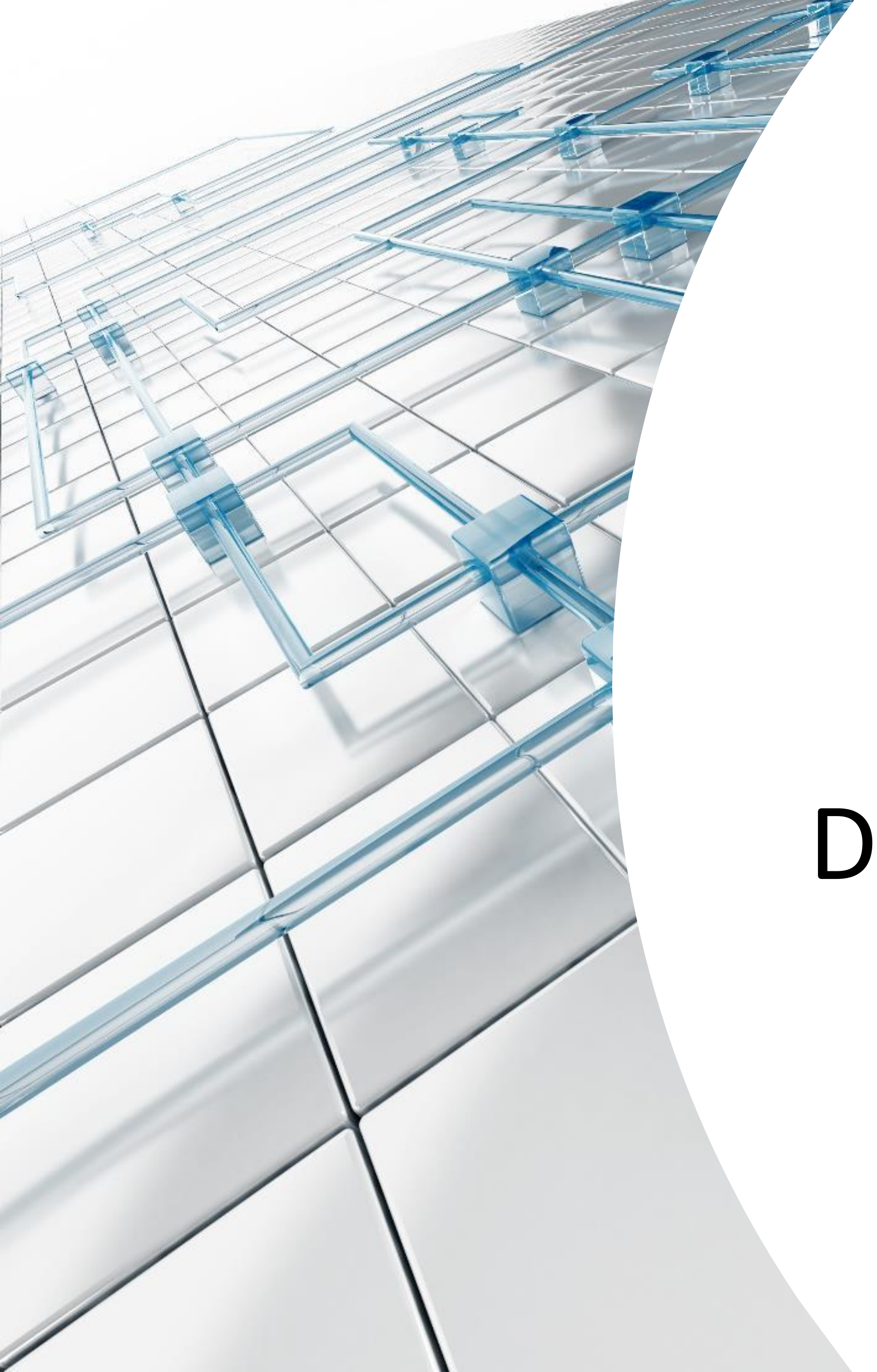
- Default installation of current release (Release THIRTEEN)

```
wget https://osm-download.etsi.org/ftp/osm-15.0-fifteen/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh 2>&1 | tee osm_install_log.txt
```

On Ubuntu 22.04 (jammy)

- 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
```



Demo

Other installation procedures

- Previous releases of OSM
  - Release FOURTEEN

    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 2>&1 | tee osm_install_log.txt
    ```

  - Release TWELVE

    On Ubuntu 20.04 (focal)

    ```
    wget https://osm-download.etsi.org/ftp/osm-12.0-twelve/install_osm.sh
    chmod +x install_osm.sh
    ./install_osm.sh 2>&1 | tee osm_install_log.txt
    ```

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
    ```

  - Release FIFTEEN

    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 ReleaseFIFTEEN-daily -t releasefifteen-daily -r testing –y
    ```

  - Release FOURTEEN

    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 ReleaseFOURTEEN-daily -t releasefourteen-daily -r testing -y
    ```

What is done by the installer?

- Install Docker CE
- Install and initialize a local Kubernetes cluster based on kubeadm, including a CNI (Flannel), container storage (OpenEBS) and a Load Balancer (MetalLB)
- Optionally install LXD and Juju client
  - Includes bootstrap of juju controller to allow the deployment of Execution Environments in local LXD server and local LXD cluster
- Deploy OSM with a helm chart, together with other required helm charts
  - Additional helm charts: airflow, alertmanager, push-gateway, cert-manager, mongodb
- Install OSM client

What can be found after OSM installation?

```
$ helm list --all-namespaces
NAME           NAMESPACE       REVISION     UPDATED                 STATUS          CHART                           APP VERSION
airflow        osm             1            2024-02-12 15:35:11 UTC deployed        airflow-1.9.0                   2.5.3
alertmanager   osm             1            2024-02-12 15:38:40 UTC deployed        alertmanager-0.22.0             v0.24.0
cert-manager   cert-manager    1            2024-02-12 15:33:46 UTC deployed        cert-manager-v1.9.1             v1.9.1
metallb        metallb-system  1            2024-02-12 15:33:42 UTC deployed        metallb-0.13.10                 v0.13.10
mongodb-k8s    osm             1            2024-02-12 15:35:05 UTC deployed        mongodb-13.9.4                  6.0.5
openebs        openebs         1            2024-02-12 15:33:38 UTC deployed        openebs-3.7.0                   3.7.0
osm            osm             1            2024-02-12 15:35:06 UTC deployed        osm-0.0.1                       15
pushgateway    osm             1            2024-02-12 15:38:35 UTC deployed        prometheus-pushgateway-1.18.2   1.4.2
```

What can be found after OSM installation?

```
$ kubectl -n osm get services
NAME                                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
airflow-postgresql                   ClusterIP   10.108.237.168   <none>        5432/TCP                     5h22m
airflow-postgresql-hl                ClusterIP   None             <none>        5432/TCP                     5h22m
airflow-redis                        ClusterIP   10.108.34.198    <none>        6379/TCP                     5h22m
airflow-statsd                       ClusterIP   10.111.9.151     <none>        9125/UDP,9102/TCP            5h22m
airflow-webserver                    NodePort    10.107.112.103   <none>        8080:5873/TCP                5h22m
airflow-worker                       ClusterIP   None             <none>        8793/TCP                     5h22m
alertmanager                         NodePort    10.100.234.112   <none>        9093:9093/TCP                5h18m
alertmanager-headless                ClusterIP   None             <none>        9093/TCP                     5h18m
grafana                              NodePort    10.107.108.119   <none>        3000:3000/TCP                5h22m
kafka                                ClusterIP   10.111.255.157   <none>        9092/TCP                     5h22m
kafka-controller-headless            ClusterIP   None             <none>        9094/TCP,9092/TCP,9093/TCP   5h22m
keystone                             ClusterIP   None             <none>        5000/TCP                     5h22m
mongodb-k8s                          ClusterIP   None             <none>        27017/TCP                    5h22m
mongodb-k8s-arbiter-headless         ClusterIP   None             <none>        27017/TCP                    5h22m
mysql                                ClusterIP   10.100.112.14    <none>        3306/TCP                     5h22m
mysql-headless                       ClusterIP   None             <none>        3306/TCP                     5h22m
nbi                                  NodePort    10.108.215.12    <none>        9999:9999/TCP                5h22m
ng-ui                                NodePort    10.99.141.251    <none>        80:80/TCP                    5h22m
prometheus                           NodePort    10.107.43.136    <none>        9090:9091/TCP                5h22m
pushgateway-prometheus-pushgateway   ClusterIP   10.96.208.7      <none>        9091/TCP                     5h19m
ro                                   ClusterIP   None             <none>        9090/TCP                     5h22m
webhook-translator                   NodePort    10.104.171.208   <none>        9998:9998/TCP                5h22m
zookeeper                            ClusterIP   10.108.49.158    <none>        2181/TCP,2888/TCP,3888/TCP   5h22m
zookeeper-headless                   ClusterIP   None             <none>        2181/TCP,2888/TCP,3888/TCP   5h22m
```

What can be found after OSM installation?

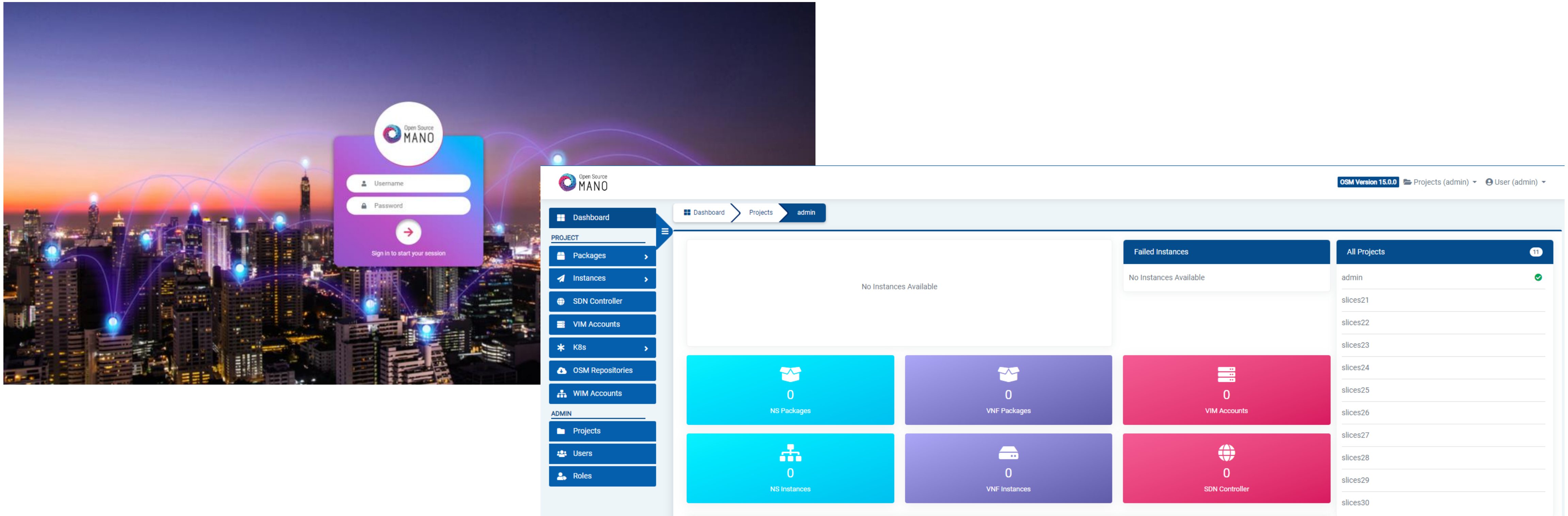
```
$ kubectl -n osm get deployments
NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
airflow-scheduler                    1/1     1            1           5h22m
airflow-statsd                       1/1     1            1           5h22m
airflow-triggerer                    1/1     1            1           5h22m
airflow-webserver                    1/1     1            1           5h22m
grafana                              1/1     1            1           5h23m
keystone                             1/1     1            1           5h23m
lcm                                  1/1     1            1           5h23m
mon                                  1/1     1            1           5h23m
nbi                                  1/1     1            1           5h23m
ngui                                 1/1     1            1           5h23m
pushgateway-prometheus-pushgateway   1/1     1            1           5h19m
ro                                   1/1     1            1           5h23m
webhook-translator                   1/1     1            1           5h23m
```

What can be found after OSM installation?

```
$ kubectl -n osm get statefulsets
NAME                  READY   AGE
airflow-postgresql    1/1     5h23m
airflow-redis         1/1     5h23m
airflow-worker        1/1     5h23m
alertmanager          1/1     5h19m
kafka-controller      3/3     5h23m
mongodb-k8s           2/2     5h23m
mongodb-k8s-arbiter   1/1     5h23m
mysql                 1/1     5h23m
prometheus            1/1     5h23m
zookeeper             1/1     5h23m
```

Using your OSM installation

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



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!