

Open Source
MANO

OSM Hackfest - Installation and first use

Guillermo Calviño (Altran)
Gerardo García (Telefónica)

[Click HERE](#)

Before installing OSM (1/2)

- Instructions
 - https://osm.etsi.org/wikipub/index.php/LXD_configuration_for_OSM_Release_TREE
- Steps:
 - **Run following commands as a non-root user**
 - Install lxd.
 - `sudo apt-get update`
 - `sudo apt-get install -y lxd`
 - `newgrp lxd`
 - Init LXD. Exec following command as a non-root user:
 - `sudo lxd init`
 - Storage backend: dir
 - LXD available over network: no
 - New network bridge: yes
 - IPv4 configuration: yes (default values apply)
 - IPv6 configuration: no

Before installing OSM (2/2)

- Steps (cont):
 - Get the MTU of your main interface:
 - ifconfig
 - Configure lxc profile to use the same MTU:
 - `sudo lxc profile device set default eth0 mtu <MTU>`
 - Test LXD **as a non-root user**:
 - `lxc launch ubuntu:16.04 test`
 - `lxc exec test bash`
 - `root@test:~# apt-get update`

- OSM installation from LXD images.

Run following commands as a non-root user:

- `wget https://osm-download.etsi.org/ftp/osm-3.0-three/install_osm.sh`
 - `chmod +x install_osm.sh`
 - `./install_osm.sh --lxdimages -l http://172.21.8.10/lxd`
-
- Other available options to install from:
 - `<NO OPTIONS>` → from binaries
 - `--source` → from source

After installing OSM

- Test OSM client
 - Add env variables to your .bashrc to use OSM client locally
 - Try 'osm'

- Test UI:
 - Google Chrome is the recommended browser
 - Access UI: `https:<IP_OSM>:8443`

For people using your own OSM

Test OSM client



- Add some env variables to your `.bashrc`:
 - `export OSM_HOSTNAME=<SO-ub_container_IP>`
 - `export OSM_RO_HOSTNAME=<RO_container_IP>`

- Try 'osm':

Usage: `osm [OPTIONS] COMMAND [ARGS]...`

Options:

`--hostname TEXT` hostname of server. Also can set `OSM_HOSTNAME` in environment

`--so-port INTEGER` hostname of server. Also can set `OSM_SO_PORT` in environment

...

For people using your own OSM Test UI



- Google Chrome installation (linux)
 - `wget -c https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`
 - `sudo apt-get update && sudo apt-get install libappindicator1`
 - `sudo dpkg -i google-chrome-stable_current_amd64.deb`

- Test UI:
 - Open web browser: `https:<IP_OSM>://8443`

Local OSM instances @ETSI

- osm-demo: 172.21.1.3
- osm-plugtest1: 172.21.1.4
- osm-plugtest2: 172.21.1.5
- osm-hackfest1: 172.21.1.9
- osm-hackfest2: 172.21.1.10

For people using ETSI OSM servers Install OSM client



- Instructions in the main page for Rel THREE:
 - https://osm.etsi.org/wikipub/index.php/OSM_Release_THREE#Install_OSM_client
- Steps:
 - `curl http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE/OSM%20ETSI%20Release%20Key.gpg | sudo apt-key add -`
 - `sudo add-apt-repository -y "deb [arch=amd64] http://osm-download.etsi.org/repository/osm/debian/ReleaseTHREE stable osmclient"`
 - `sudo apt-get update`
 - `sudo apt-get install -y python-osmclient`

For people using ETSI OSM servers

Test OSM client



- Add some env variables to your `.bashrc`:
 - `export OSM_HOSTNAME=<OSM_IP>`
 - `export OSM_RO_HOSTNAME=<OSM_IP>`

- Try 'osm':

Usage: `osm [OPTIONS] COMMAND [ARGS]...`

Options:

`--hostname TEXT` hostname of server. Also can set `OSM_HOSTNAME` in environment

`--so-port INTEGER` hostname of server. Also can set `OSM_SO_PORT` in environment

...

For people using ETSI OSM servers

Test UI



- Google Chrome installation (linux)
 - `wget -c https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`
 - `sudo apt-get update && sudo apt-get install libappindicator1`
 - `sudo dpkg -i google-chrome-stable_current_amd64.deb`
- Access UI:
 - `https://<IP_OSM>:8443`

Adding VIM accounts

- VIMs:

Name	Type	AUTH URL	tenant	user	Password	SDN controller
openstack1	openstack	http://172.21.2.20:5000/v2.0	xxx	xxx	xxx	NO
openstack2-epa	openstack	http://172.21.2.22:5000/v2.0	xxx	xxx	xxx	YES
openstack3	openstack	http://172.21.5.4:5000/v3	xxx	xxx	xxx	NO

- Test VIMs:

- ping <IP>
- curl http://<IP>:5000/v2.0
- Install python-openstackclient
 - sudo apt-get install python-openstackclient

Adding VIM accounts

- Load Openstack credentials and run some commands for testing:
 - export OS_AUTH_URL=xxx
 - export OS_USERNAME=xxx
 - export OS_TENANT_NAME=xxx
 - export OS_PASSWORD=xxx
 - openstack image list
 - openstack network list
 - openstack flavor list
 - openstack server list

Adding VIM accounts

- Add your first VIM 'openstack1' with the OSM client:
 - `osm vim-create --name openstack1 --account_type openstack \`
`--auth_url http://172.21.2.20:5000/v2.0 \`
`--user xxx --password xxx --tenant xxx \`
`--description "ETSI openstack site 1, with tenant xxx"`
 - `osm vim-list`

Adding VIM accounts

- Add a second VIM 'openstack3' with the OSM client:
 - `osm vim-create --name openstack3 --account_type openstack \`
`--auth_url http://172.21.5.4:5000/v3 \`
`--user xxx --password xxx --tenant xxx \`
`--description "Windriver ETSI openstack site 3, with tenant osm" \`
`--config '{region_name: RegionOne, project_domain_name: Default,`
`user_domain_name: Default}'`
 - `osm vim-list`

Adding images to the VIM

- Image management is not implemented in OSM today. It has to be done independently on each VIM.
- **IMAGES HAVE BEEN ALREADY ADDED TO THE REMOTE VIMS IN THE HACKFEST**
- Example for Openstack:
 - `openstack image create --file="./cirros-0.3.4-x86_64-disk.img" --container-format=bare --disk-format=qcow2 --public cirros034`

Adding images to the VIM

Image name in descriptors	Filename
ubuntu1604	xenial-server-cloudimg-amd64-disk1.img (you can get it from https://cloud-images.ubuntu.com/xenial/current/)
US1604	US1604.qcow2
hackfest3-mgmt	hackfest3-mgmt-qcow2
hackfest-pktgen	hackfest-pktgen-qcow2
cirros034	cirros-0.3.4-x86_64-disk.img

Deploying our first NS with OSM UI



A screenshot of the Open Source MANO (OSM) user interface. The top navigation bar is black with white text for "LAUNCHPAD", "CATALOG", "ACCOUNTS", and "ADMINISTRATION". The "LAUNCHPAD" tab is highlighted in blue. Below the navigation bar, the main content area is titled "LAUNCHPAD" in large, bold, black letters. On the left side, there is a panel titled "NETWORK SERVICES" with a back arrow icon. Below this title, there is a status summary: "TOTAL: 0 RUNNING: 0 FAILED: 0 SCALING OUT: 0 SCALING IN: 0 INITIALIZING: 0". Below the status summary is a button with a plus sign and the text "Instantiate Service". Below the button is a table with the following headers: "NS NAME", "NSD", "STATUS", and "UPTIME". The table body is currently empty. On the right side, there is a panel titled "NETWORK SERVICE DETAILS" which is currently blank.

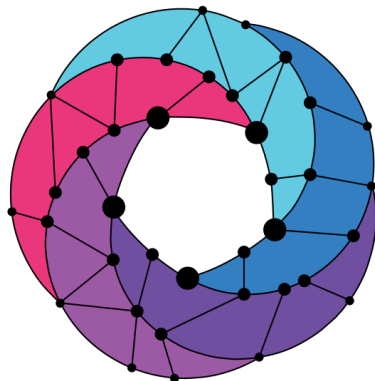
Deploying our first NS with the UI

- Add VNF package
- Add NS package
- Instantiate
- Get VNF record and obtain mgmt IP address
- Access to the VNF via SSH
- Delete NS instance
- Delete NS
- Delete VNF

Deploying our first NS with OSM client



- Add VNF and NS package
 - `osm upload-package cirros_vnf.tar.gz`
 - `osm vnfd-list`
 - `osm upload-package cirros_2vnf_ns.tar.gz`
 - `osm nsd-list`
- Instantiate
 - `osm ns-create --nsd_name cirros_2vnf_ns --ns_name <ns-instance-name> --vim_account <data-center-name>`
 - `osm ns-list`
- Delete NS instance
 - `osm ns-delete <ns-instance-name>`
 - `osm ns-list`
- Delete VNF and NS package
 - `osm nsd-delete cirros_2vnf_ns`
 - `osm nsd-list`
 - `osm vnfd-delete cirros_vnfd`
 - `osm vnfd-list`



Open Source
MANO

Find us at:

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