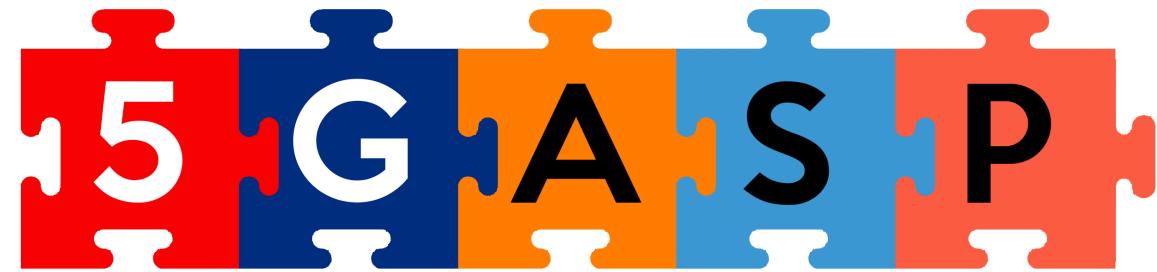


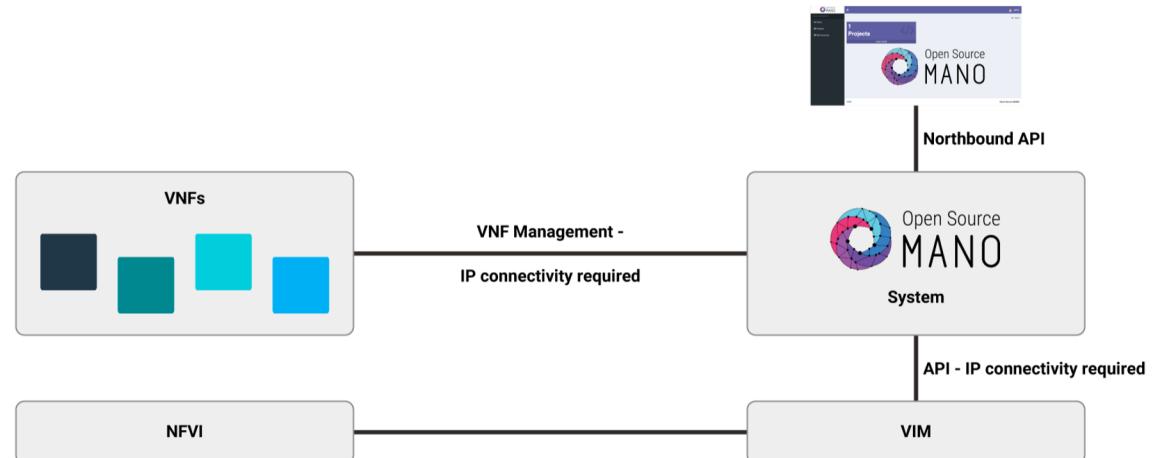
Build your VNF from scratch



Basic architecture

- Open Source MANO (OSM – version 10.0.3) will be used
- OpenStack (version wallaby stable) will be our VIM account

The OSM interaction with VIMs and VNFs



OSM

Virtual Machine requirements:

- 2 CPUs, 6 (8 recommended) GB RAM, 40GB disk and a single interface with Internet access
- Base image: [Ubuntu20.04 \(64-bit variant required\)](https://ubuntu.etsi.org/ftp/ubuntu-20.04.1 LTS (Hirsute Hippo) - 64-bit variant required)

Installing OSM in the VM:

```
$ wget https://osm-download.etsi.org/ftp/osm-10.0-ten/install\_osm.sh
```

```
$ chmod +x install_osm.sh
```

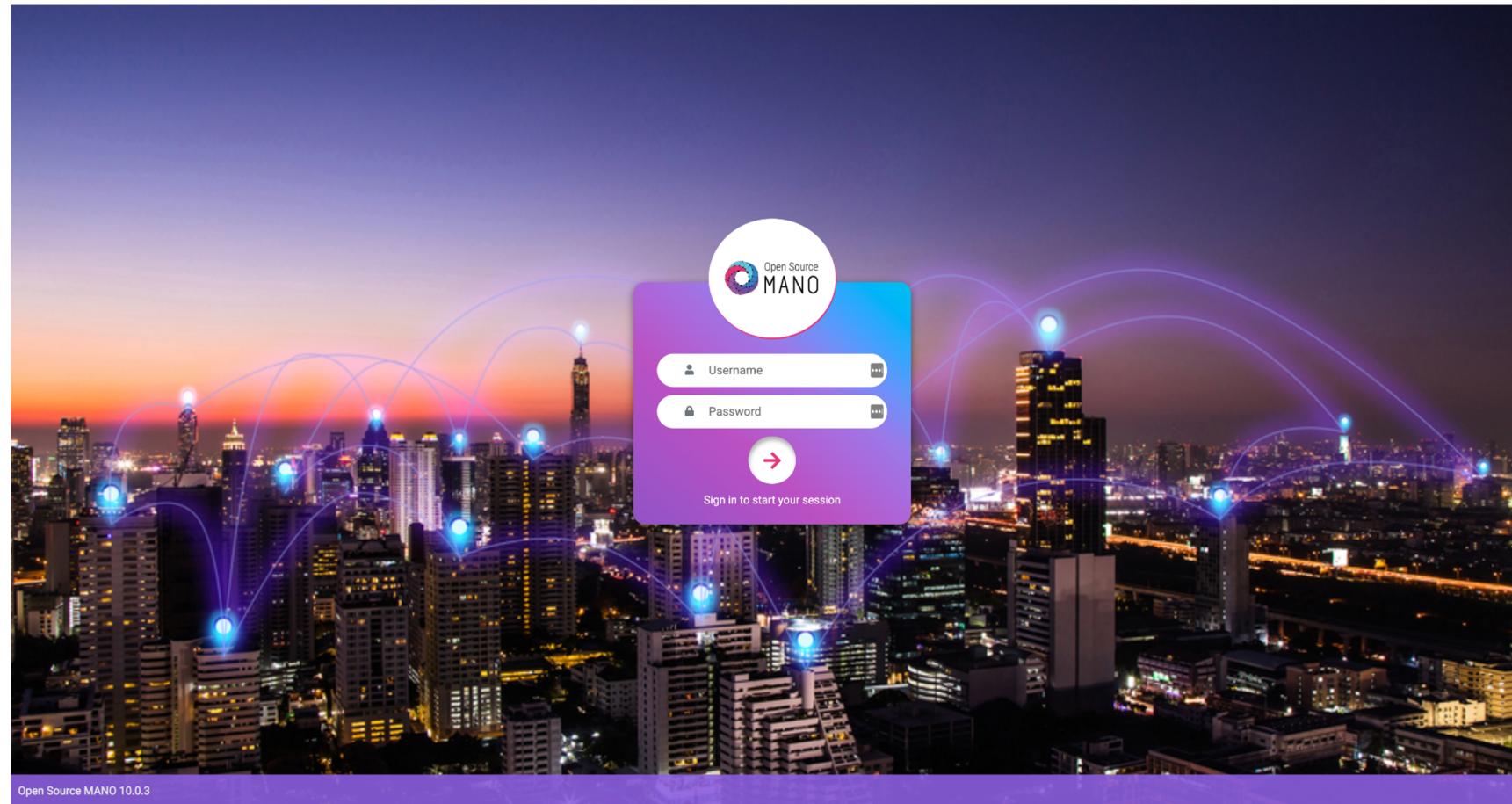
```
$ ./install_osm.sh
```

```
($ ./install_osm.sh --k8s_monitor) - install OSM with an add-on to monitor the Kubernetes cluster
```



Access OSM

After the installation, you can access OSM by <http://1.2.3.4> (replace by your host IP):



By default, user and password are both 'admin':

The screenshot shows the OSM 10.0.3 dashboard. At the top, a banner reads "Here is the new version 10.0.3 of OSM!". The top right corner shows "OSM Version 10.0.3", "Projects (admin)", and "User (admin)". The left sidebar has sections for PROJECT (Packages, Instances, SDN Controller, VIM Accounts, K8s, OSM Repositories, WIM Accounts) and ADMIN (Projects, Users, Roles). The main dashboard area shows the following statistics:

Category	Count
NS Packages	5
VNF Packages	3
VIM Accounts	4
NS Instances	1
VNF Instances	1
SDN Controller	0

The "Failed Instances" section indicates "No Instances Available". The "All Projects" section shows 1 project with the user "admin" selected.



Adding a VIM account

- We'll use **OpenStack**, but there are other options, like Microsoft Azure, VMware vCloud Director, OpenVIM, etc.
- To add the VIM through OSM:

```
$ osm vim-create --name openstack-site --user admin --password userpwd \
--auth_url http://10.10.10.11:5000/v2.0 --tenant admin --account_type openstack
```

- <https://osm.etsi.org/docs/user-guide/04-vim-setup.html#openstack> to more advanced options.



Add VIM through GUI

Here is the new version 10.0.3 of OSM!

OSM Version 10.0.3 Projects (admin) User (admin)

The screenshot shows the 'New VIM Account' form. The left sidebar has a 'PROJECT' section with 'Dashboard', 'Packages', 'Instances', 'SDN Controller', 'VIM Accounts' (selected), 'K8s', 'OSM Repositories', and 'WIM Accounts'. The 'ADMIN' section includes 'Projects', 'Users', and 'Roles'. The top navigation bar shows 'Dashboard > Projects > admin > VIM Accounts > New VIM'. The main form has fields for 'Name*', 'Type*', 'VIM Project/Tenant Name*', 'Description', 'VIM URL*', 'Schema Type', 'VIM Username*', 'VIM Password*', 'VIM Location' (with a map input), and 'Upload Config' (with 'Choose File' and 'Browse' buttons). A note says 'Please upload file with .yaml or .yml format'. At the bottom are 'Back to VimAccounts' and 'Create' buttons.

New VIM Account

Mandatory fields are marked with an asterisk (*)

Name*

Type*

VIM URL*

VIM Username*

VIM Location

Upload Config

VIM Project/Tenant Name*

Description

Schema Type

VIM Password*

Name Latitude Longitude

Type the Data location name, Latitude & Longitude to show in map view

Please upload file with .yaml or .yml format

Choose File Browse

Back to VimAccounts Create

Creating the packages - VNF

Directory to save files

Package type
&
Package name

```
ubuntu@osm10-0-3-1:~$ osm package-create --base-directory ~/tutorial vnf tutorial
Creating the VNF structure: /home/ubuntu/tutorial/tutorial
Creating folder:      tutorial_vnf
Creating folder:      /home/ubuntu/tutorial/tutorial_vnf/charms
Creating folder:      /home/ubuntu/tutorial/tutorial_vnf/cloud_init
Creating folder:      /home/ubuntu/tutorial/tutorial_vnf/images
Creating folder:      /home/ubuntu/tutorial/tutorial_vnf/icons
Creating folder:      /home/ubuntu/tutorial/tutorial_vnf/scripts
Creating file:        /home/ubuntu/tutorial/tutorial_vnf/tutorial_vnfd.yaml
Creating file:        /home/ubuntu/tutorial/tutorial_vnf/cloud_init/cloud-config.txt
Creating file:        /home/ubuntu/tutorial/tutorial_vnf/README.md
Created
```



Virtual Deployment Unit (VDU)

- Basic part of a VNF
- VM that hosts the network function
- Its properties are described in the VNF descriptor



Changing the VNF descriptor

~/tutorial/tutorial_vnf/tutorial_vnfd.yaml

Basic info

Deployment flavour

Internal and external connection points

Image to be used (needs to exist in VIM)

Specifications for VDU

VDU (Virtual Deployment Unit)

```
vnfd:
  id: tutorial_vnf
  description: A basic VNF descriptor with one VDU
  product-name: tutorial_vnf
  version: "1.0"
  df:
    - id: default-df
      instantiation-level:
        - id: default-instantiation-level
          vdu-level:
            - number-of-instances: "1"
            | vdu-id: tutorial
          vdu-profile:
            - id: tutorial
              min-number-of-instances: "1"
  ext-cpd:
    - id: vnf-cp0-ext
      int-cpd:
        cpd: vdu-eth0-int
        vdu-id: tutorial
  mgmt-cp: vnf-cp0-ext
  sw-image-desc:
    - id: "Ubuntu Server 20.04"
      image: "Ubuntu Server 20.04"
      name: "Ubuntu Server 20.04"
```

```
virtual-compute-desc:
  - id: tutorial-compute
    virtual-cpu:
      num-virtual-cpu: "1"
    virtual-memory:
      size: "1.0"
  virtual-storage-desc:
    - id: tutorial-storage
      size-of-storage: "10"
  vdu:
    - cloud-init-file: cloud-init.cfg
      id: tutorial
      int-cpd:
        - id: vdu-eth0-int
          virtual-network-interface-requirement:
            - name: vdu-eth0
              virtual-interface:
                type: PARAVIRT
      name: tutorial
      sw-image-desc: "Ubuntu Server 20.04"
      virtual-compute-desc: tutorial-compute
      virtual-storage-desc: tutorial-storage
```

Cloud-init file

- ~/tutorial/tutorial_vnf/cloud_init/cloud-init.cfg
- Basic initial configurations

```
1 #cloud-config
2 password: tutorial
3 chpasswd: { expire: False }
4 ssh_pwauth: True
5 package_update: true
6 packages:
7   - nmap
```



Validating and Uploading VNF package

VNF directory

```
ubuntu@osm10-0-3-1:~/tutorial$ osm nfpkg-create ~/tutorial/tutorial_vnf
Validating package /home/ubuntu/tutorial/tutorial_vnf
Validation OK
List of charms in the descriptor: []
Adding File: tutorial_vnf
Package created: /home/ubuntu/tutorial/tutorial_vnf.tar.gz
Uploading package /home/ubuntu/tutorial/tutorial_vnf.tar.gz
fddd3a9-6a6a-45a6-b75f-66647e43861a
```



Creating the packages - NS

Directory to save files

Package type
&
Package name

```
ubuntu@osm10-0-3-1:~/tutorial$ osm package-create --base-directory ~/tutorial ns tutorial
Creating the NS structure: /home/ubuntu/tutorial/tutorial
Creating folder:      /home/ubuntu/tutorial/tutorial_ns
Creating folder:      /home/ubuntu/tutorial/tutorial_ns/icons
Creating folder:      /home/ubuntu/tutorial/tutorial_ns/charms
Creating file:        /home/ubuntu/tutorial/tutorial_ns/tutorial_nsd.yaml
Creating file:        /home/ubuntu/tutorial/tutorial_ns/README.md
Created
```



Changing the NS descriptor

~/tutorial/tutorial_ns/tutorial_nsd.yaml

Basic info

```
1 nsd:  
2   nsd:  
3     - id: tutorial_ns  
4       name: tutorial_ns  
5       version: "1.0"  
6       description: Simple NS with one VNF and a single Virtual Link  
7       virtual-link-desc:  
8         - id: mgmtnet_2  
9           mgmt-network: true  
10          vim-network-name: proj_net  
11          vnfd-id:  
12            - tutorial_vnf  
13            df:  
14              - id: default-df  
15              vnf-profile:  
16                - id: "1"  
17                  virtual-link-connectivity:  
18                    - constituent-cpd-id:  
19                      - constituent-base-element-id: "1"  
20                        constituent-cpd-id: vnf-cp0-ext  
21                        virtual-link-profile-id: mgmtnet_2  
22            vnfd-id: tutorial_vnf
```

Virtual Link
(Connects with VIM Network)

Id of the VNF created before

Deployment flavour - establish connections
between connection points and VIM network



Validating and Uploading NS package

NS directory

```
ubuntu@osm10-0-3-1:~/tutorial/tutorial_ns$ osm nspkg-create ~/tutorial/tutorial_ns
Validating package /home/ubuntu/tutorial/tutorial_ns
Validation OK
List of charms in the descriptor: []
Adding File: tutorial_ns
Package created: /home/ubuntu/tutorial/tutorial_ns.tar.gz
b7999a5b-6f81-4081-bfce-955aa7470f43
```



NS Instantiation

```
ubuntu@osm10-0-3-1:~/tutorial/tutorial_ns$ osm ns-create --ns_name tutorial_ns --nsd_name tutorial_ns --vim_account Tron  
cecfffc11-dc74-41be-b547-05d15e32bbe3
```

Here is the new version 10.0.3 of OSM!

OSM Version 10.0.3 Projects (admin) User (admin)

New NS

Entries 10

NS Instances

init running / configured failed scaling

Name	Identifier	Nsd name	Operational Status	Config Status	Detailed Status	Actions
Tutorial	b4864a7b-28cf-4795-b3a7-c6d0b275aa2a	hackfest_basic-ns	✗	✓	Operation: INSTANTIATING.d331f984-d433-4df3-896c-93103d493bed, Stage 2/5: deployment of KDUs, VMs and execution environments. Detail: Deploying at VIM: Error at create net: More than one network found with this criteria: {'mgmt': True, 'name': 'mgmtnet'}. Error at create vdu: Cannot CREATE vdu because depends on failed CREATE net id=nsrs:b4864a7b-28cf-4795-b3a7-c6d0b275aa2a:vld.mgmtnet): More than one network found with this criteria: {'mgmt': True, 'name': 'mgmtnet'}	✗ !h v d Action
prometheus_test	e66df936-f0fd-4e0d-a178-ca25d5b6890c	ci_cd_service_test_ns	✓	✓	Done	v l t d Action
prometheus_test_2	8a17bffe-1b2a-4934-8d95-19c23f14c067	ci_cd_service_test_ns	✓	✓	Done	v l t d Action
tutorial_ns	cecfffc11-dc74-41be-b547-05d15e32bbe3	tutorial_ns	✓	✓	Done	v l t d Action

Open Source MANO

Dashboard > Projects > admin > NS Instances

PROJECT

- Packages
- Instances
- NS Instances
- VNF Instances
- PDU Instances
- NetSlice Instances
- Operational Dashboard

SDN Controller

VIM Accounts

K8s

OSM Repositories

WIM Accounts

ADMIN

Projects

Users





OpenStack

- Check in your VIM if the instance is running and active

Screenshot of the OpenStack Horizon web interface showing the Instances page.

The top navigation bar shows the project "atnog • 5gasp_base" and user "pedro.bas".

The left sidebar has dropdown menus for Project, API Access, Compute (selected), and Overview.

The main content area is titled "Instances" and displays a table of 12 items.

Actions	Instance ID	Launch Instance	Delete Instances	More Actions						
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
tutorial_ns-1-tutorial-0	Ubuntu Server 20.04	10.0.12.228	ci_cd_service_test_VM-flv	-	Active	nova	None	Running	0 minutes	Create Snapshot



SSH into our machine

```
• 100% → ssh ubuntu@10.0.12.228
The authenticity of host '10.0.12.228 (10.0.12.228)' can't be established.
ECDSA key fingerprint is SHA256:ON/02BA00aCv0G1IPUZ+dHwWnA00DcK7cUr+eGYQ2A4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.12.228' (ECDSA) to the list of known hosts.
ubuntu@10.0.12.228's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-86-generic x86_64)
ubuntu@tutorial-ns-1-tutorial-0:~$
```

Check the packages installed with the cloud-init file

Using user `ubuntu` and the password defined in the `cloud-init` file

```
ubuntu@tutorial-ns-1-tutorial-0:~$ nmap
Nmap 7.01 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
    Can pass hostnames, IP addresses, networks, etc.
    Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
    -iL <inputfilename>: Input from list of hosts/networks
    -iR <num hosts>: Choose random targets
    --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
    --excludefile <exclude_file>: Exclude list from file
```





Thanks for watching

You can find more content at

community.5gasp.eu

