

Open Source MANO

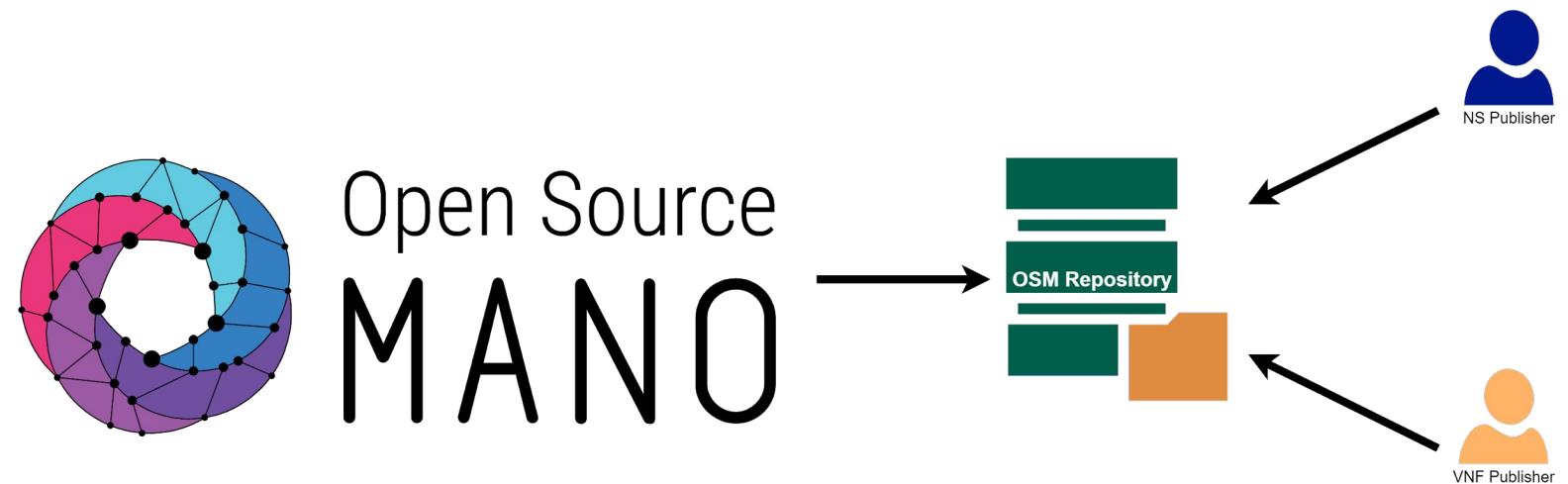
Managing VNF Packages with new
OSM Repositories

Felipe Vicens (ATOS)

Description

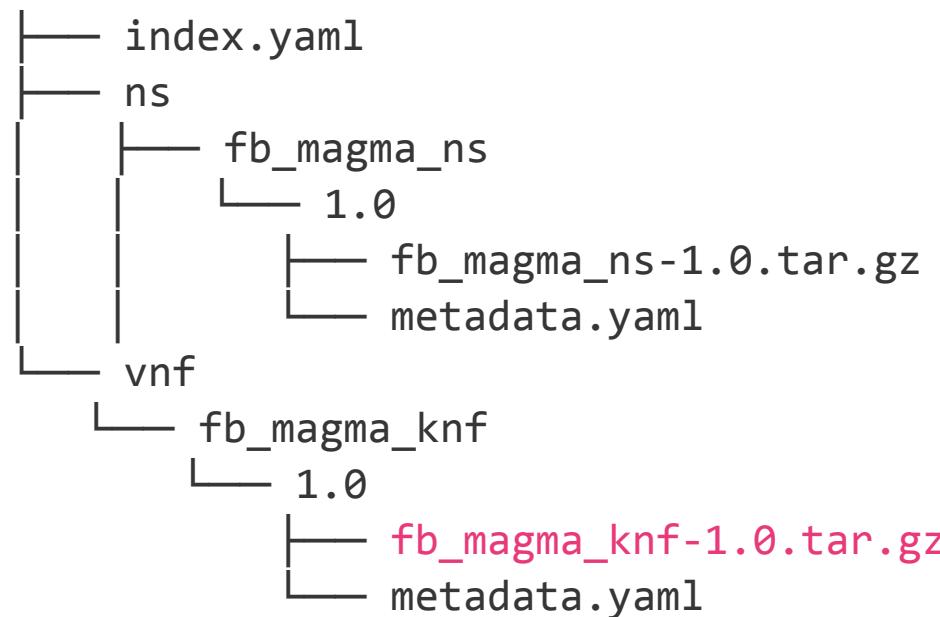
OSM Repository is a set of NSs & VNFs related artifacts and metadata, organized with a predefined structure, aimed to expose them to OSM users, by using **HTTP** or **HTTPS**.

- It exposes:
 - The VNF Artifacts
 - The NS Artifacts
 - The Repository index

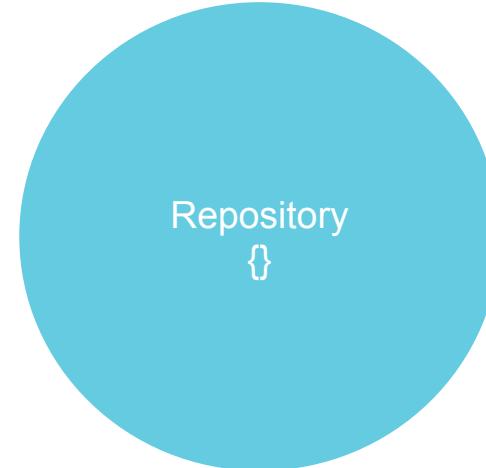


Server Side commands

- Repository server side:
 - One line repository creation and indexing
 - `osm repo-index --origin packages --destination repository`



Graphical Representation

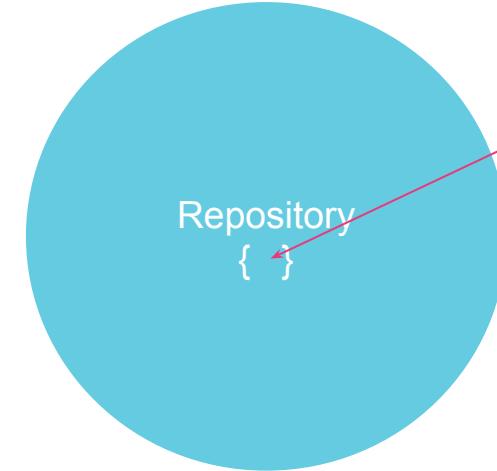


Repository Structure

- Repositories can be local (exposed in the local filesystem) or remote, both exposed by HTTP
- In both cases, the repository will have a structure like this:
 - A index.yaml file which is in charge of indexing the packages
 - A folders structure (trying to be self-explanatory)
 - One artifact's version and its metadata per folder

<https://osm-download.etsi.org/ftp/vnf-catalog/>
repository/
 └── index.yaml
 └── ns/
 └── <ns_name>/
 └── <ns_version>/
 ├── metadata.yaml
 └── <ns_name>-<ns_version>.tar.gz
 └── vnf/
 └── <vnf_name>/
 └── <vnf_version>/
 ├── metadata.yaml
 └── <vnf_name>-<vnf_version>.tar.gz

Graphical Representation



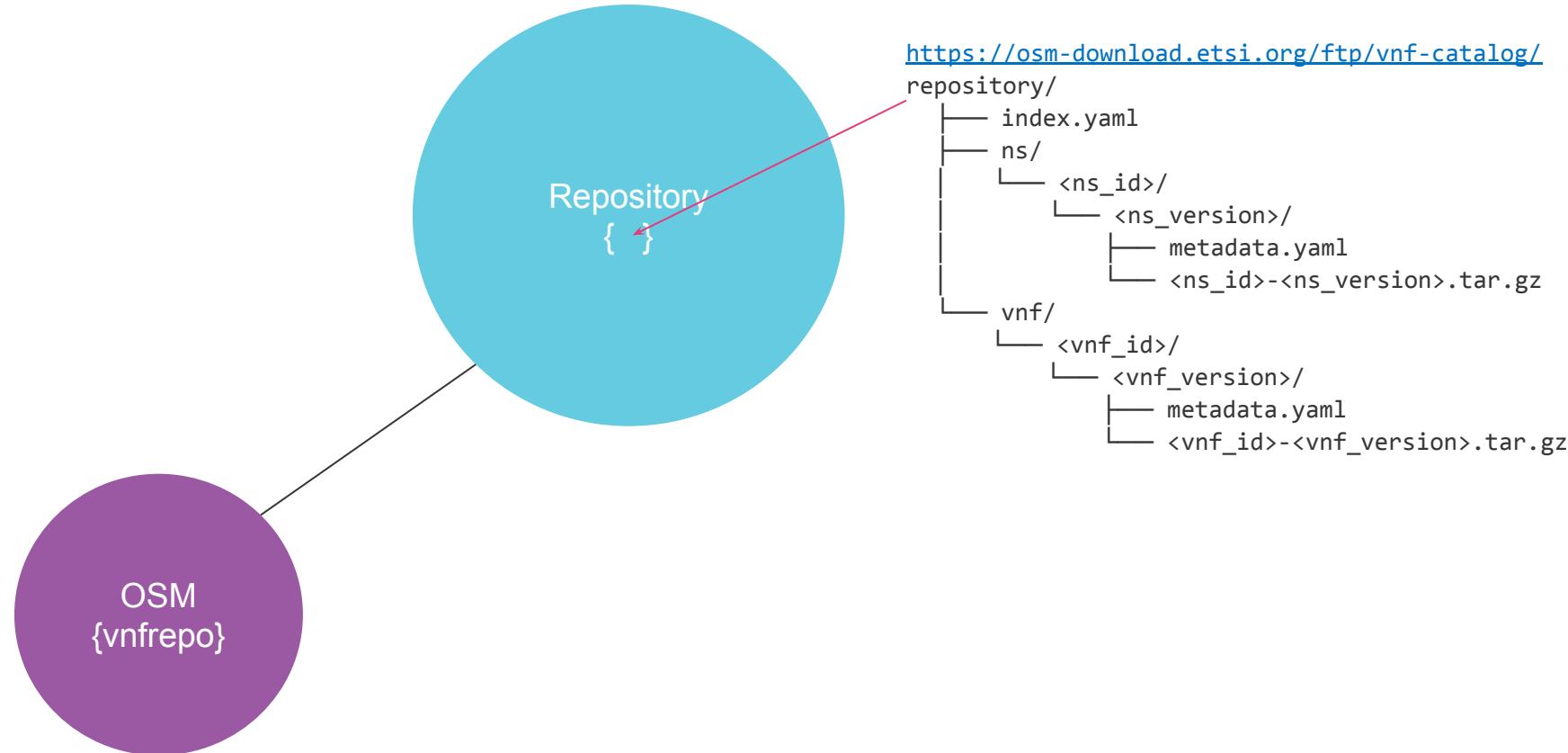
<https://osm-download.etsi.org/ftp/vnf-catalog/>

```
repository/
  └── index.yaml
  └── ns/
      └── <ns_id>/
          └── <ns_version>/
              ├── metadata.yaml
              └── <ns_id>-<ns_version>.tar.gz
  └── vnf/
      └── <vnf_id>/
          └── <vnf_version>/
              ├── metadata.yaml
              └── <vnf_id>-<vnf_version>.tar.gz
```

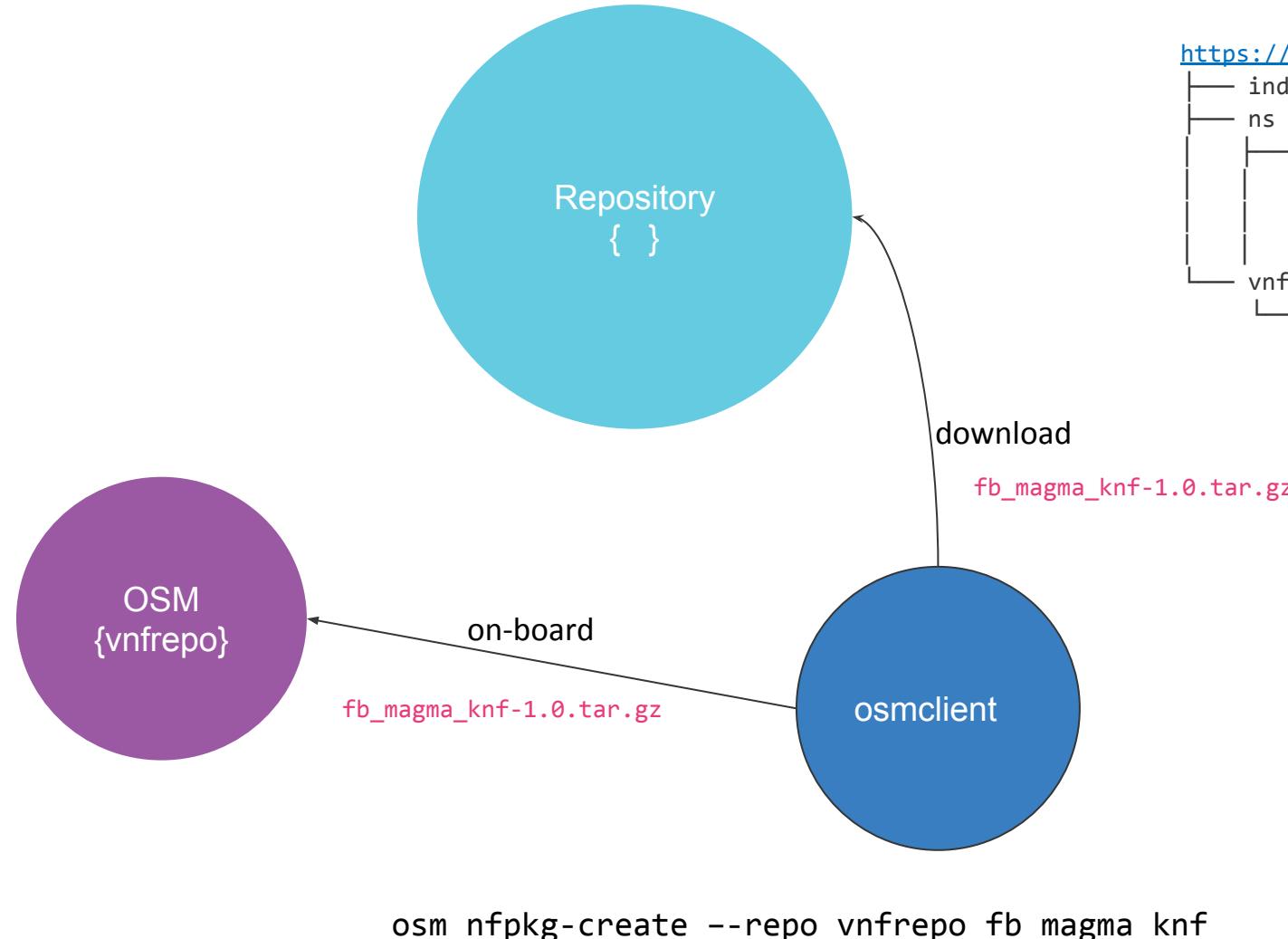
Client Side commands

- Add an OSM Repository
 - osm repo-add --description "my vnf repository" vnfrepo <https://osm-download.etsi.org/ftp/vnf-catalog/>
- List repositories
 - osm repo-list
- Delete a repository
 - osm repo-delete vnfrepo

Graphical Representation



Graphical Representation



<https://osm-download.etsi.org/ftp/vnf-catalog/>

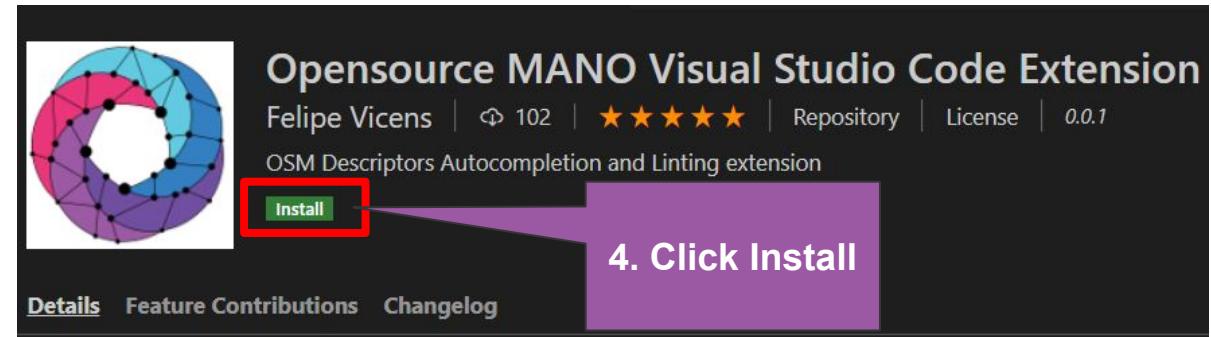
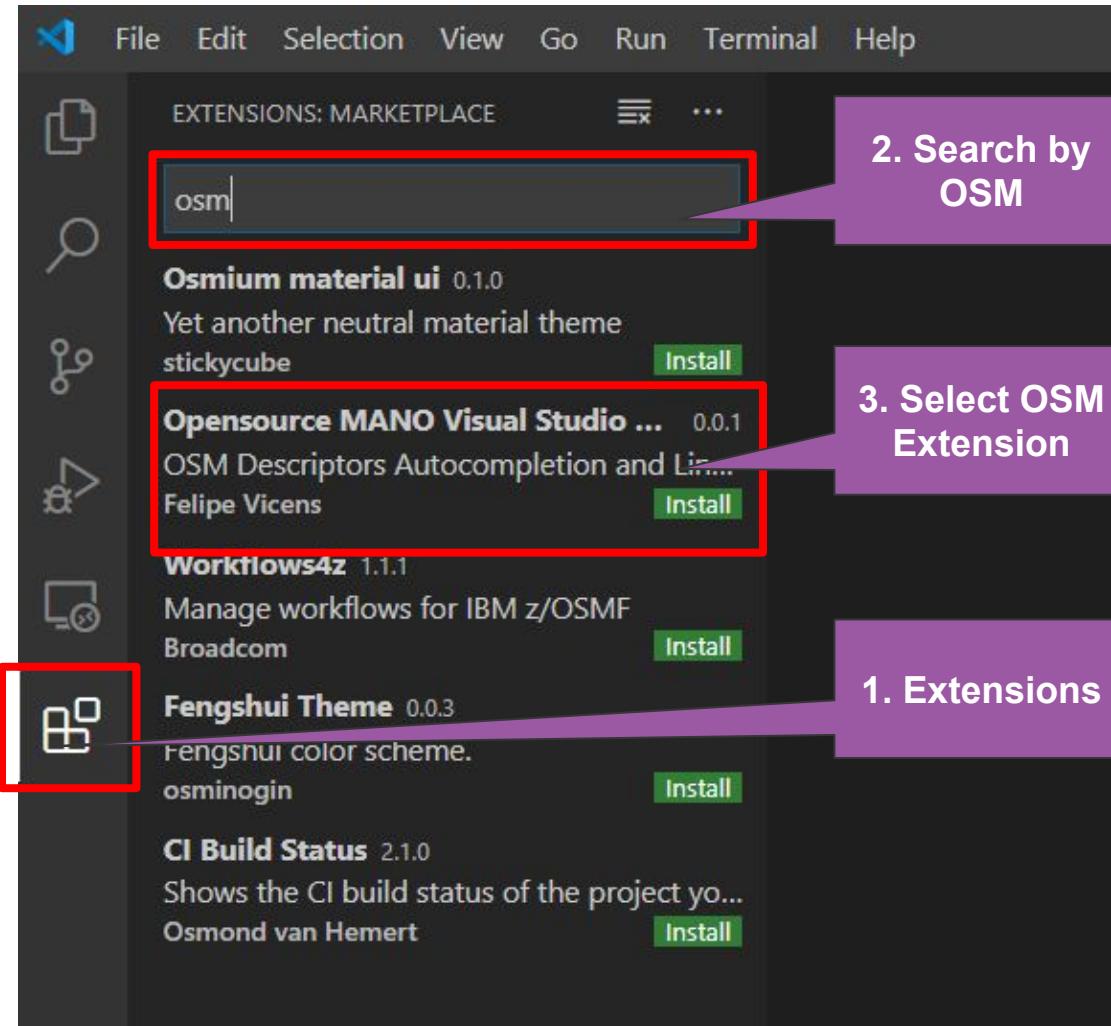
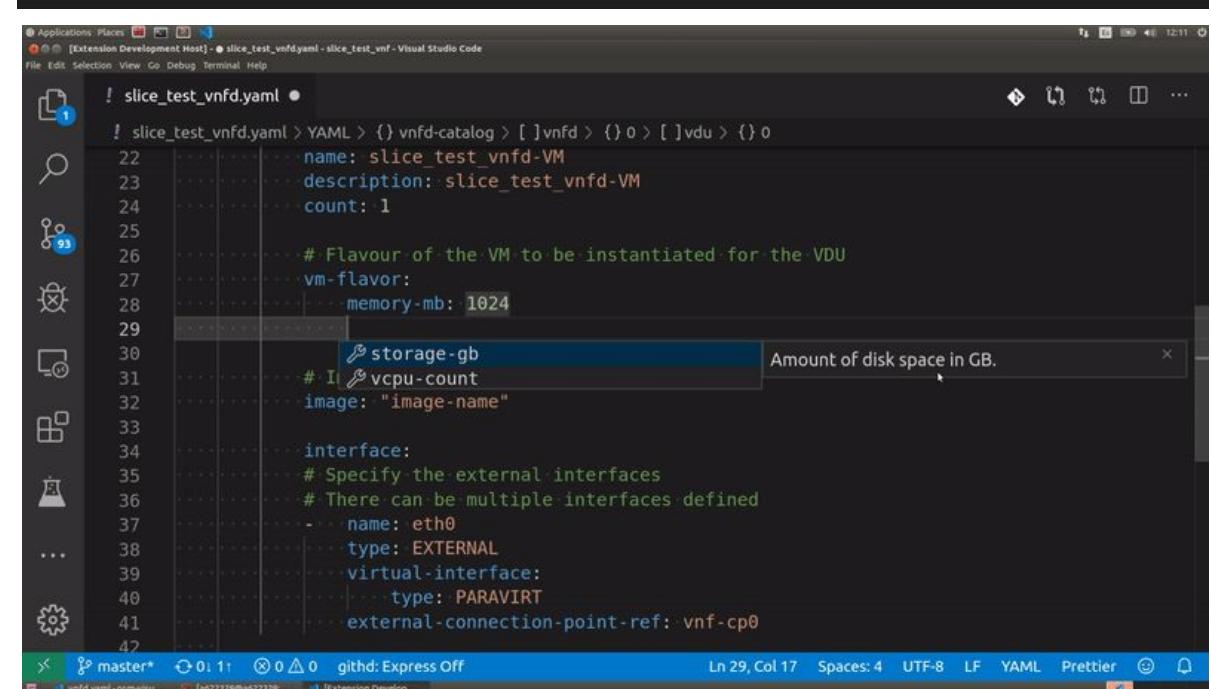
```

index.yaml
└── ns
    └── fb_magma_ns
        └── 1.0
            ├── fb_magma_ns-1.0.tar.gz
            └── metadata.yaml
└── vnf
    └── fb_magma_knf
        └── 1.0
            ├── fb_magma_knf-1.0.tar.gz
            └── metadata.yaml
  
```

Client Side commands

- List package inside a repository
 - `osm nfpkg-repo-list`
 - `osm ns pkg-repo-list`
- On-board a package from a repository to OSM
 - `osm nfpkg-create --repo vnfrepo fb_magma_knf`
 - `osm ns pkg-create --repo vnfrepo fb_magma_ns`
- Show packages in a repository
 - `osm nfpkg-repo-show --repo vnfrepo fb_magma_knf`
 - `osm ns pkg-repo-show --repo vnfrepo fb_magma_ns`

Installing VS Code plugin for OSM Descriptors design

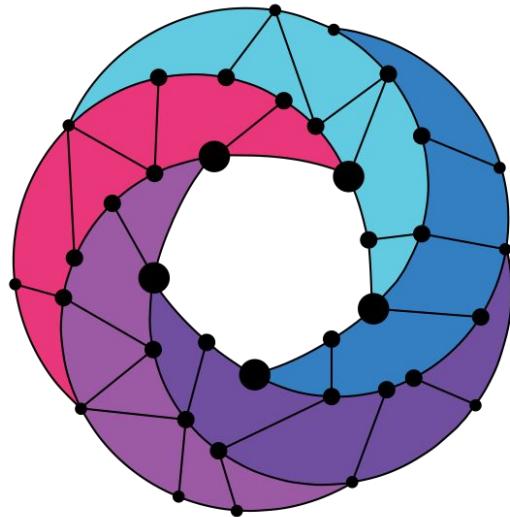



The screenshot shows VS Code displaying a YAML configuration file named 'slice_test_vnfd.yaml'. The file contains several sections, including 'vnfd-catalog', 'vnfd', 'vdu', 'storage-gb', 'vcpu-count', and 'interface'. A code completion dropdown is visible over the 'storage-gb' section, with the text 'Amount of disk space in GB.'. The status bar at the bottom indicates the file is in 'master' branch, has 0 changes, and is using Express Off for git. Other status indicators include Ln 29, Col 17, Spaces: 4, UTF-8, LF, YAML, Prettier, and a bell icon.

```

! slice_test_vnfd.yaml •
! slice_test_vnfd.yaml > YAML > {} vnfd-catalog > [ ]vnfd > {}o > [ ]vdu > {}o
22   name: slice_test_vnfd-VM
23   description: slice_test_vnfd-VM
24   count: 1
25
26   # Flavour of the VM to be instantiated for the VDU
27   vm-flavor:
28     memory-mb: 1024
29
30   storage-gb
31     # I
32     vcpu-count
33     image: "image-name"
34
35   interface:
36     # Specify the external interfaces
37     # There can be multiple interfaces defined
38     - name: eth0
39       type: EXTERNAL
40       virtual-interface:
41         type: PARAVIRT
42         external-connection-point-ref: vnf-cp0

```



Open Source MANO

Find us at:

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