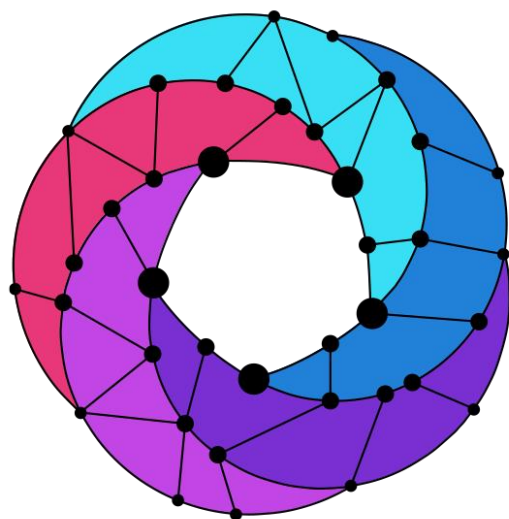




# Open Source MANO

VNF PACKAGE CREATION & ONBOARDING  
RIFT.io

# GETTING STARTED WITH OSM R1

- INSTALL OSM R1
- CREATE VNF & NS PACKAGES
- ONBOARD VNF & NS PACKAGES
- INSTANTIATE NETWORK SERVICES

- Used for onboarding NS and VNFs
- Descriptors(Mandatory) and Additional artifacts (eg icons, charms, scripts)
- Provides a simple and one click on-boarding experience.
- Defined directory structure for each package type.
- Mandatory checksums.txt file
- The packages are distributed in the tar.gz format.

# VNF PACKAGE STRUCTURE

<vnfname>\_vnfd.tar.gz

checksums.txt

README

<vnfname>\_vnfd.yaml

charms

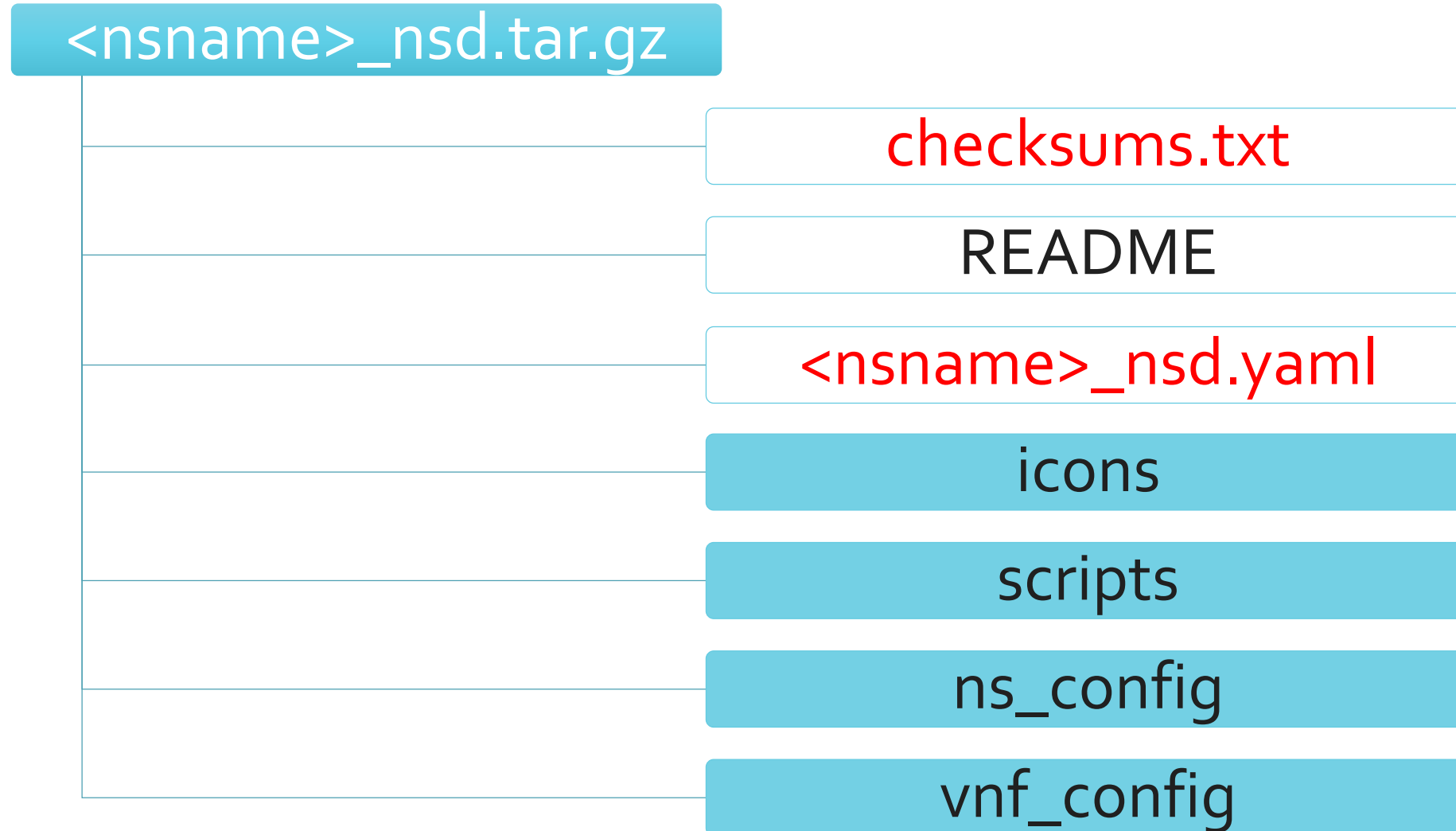
cloud\_init

icons

images

scripts

# NS PACKAGE STRUCTURE

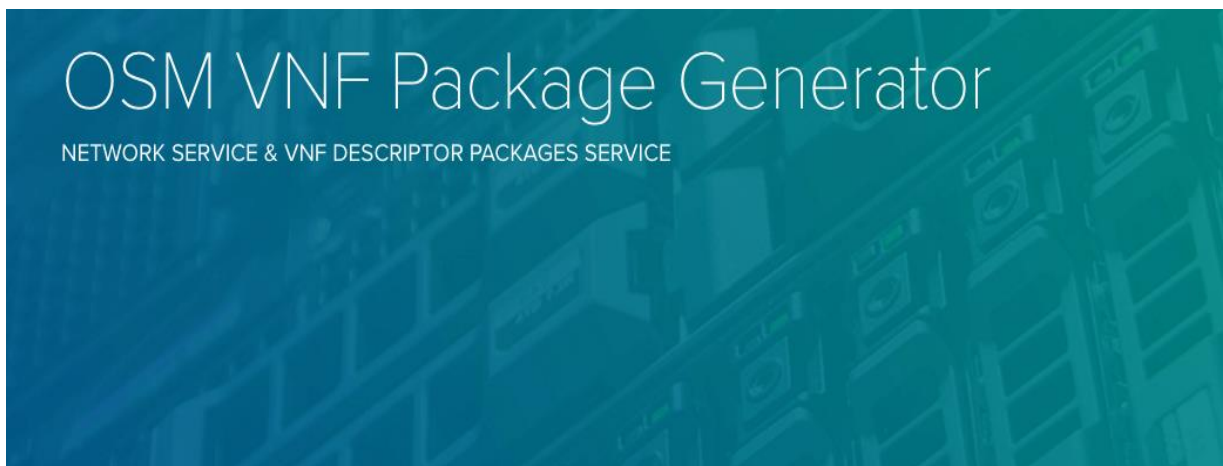


# CREATING PACKAGES

- Use a simple web based form
- Use package creation utility
- Use composer

# WEB BASED PACKAGE GENERATOR

- <https://riftio.com/osm-vnf-package-generator/>



Creating a VNF Descriptor package for your virtual network function/service is essential for deploying and operating your solution in your customer's NFV and NFV MANO environment. RIFT.io has the most VNF on-boarding experience in the industry. Using a standardized, Common Information Model (CIM) RIFT.io will get you started, with no modification to your VNF, and provide a path for evolving your solution to a distributed, cloud-scale architecture.

Take your first steps today. Provide some information about your VNF below and RIFT.io will automatically create a Network Service and VNF Descriptor Package that empowers you to quickly validate and deploy your VNFs and network services in multiple cloud environments.

VNF Name \*

0 of 60 max characters

VNF Vendor \*

0 of 60 max characters

NSD Descriptor Package

☒ Yes, auto-generate a corresponding NSD Descriptor Package for this VNF.

VM FLAVOR RELATED PARAMETERS

VCPUs Count \*

1

Please enter a value between 1 and 128.

Memory - MB \*

512

Please enter a value between 1 and 65535.

Storage - GB \*

10

Please enter a value between 1 and 65535.

VDU PARAMETERS

We assume you are using a single VDU.

Image Name \*

Be sure to make sure your bootable VM image is preloaded in the cloud where you plan to instantiate it (e.g. OpenStack or AWS)

0 of 60 max characters

Number Of Additional External Interfaces \*

We're automatically adding in a OM-MGMT interface option to your descriptor. Here you can add additional interfaces.

0

Please enter a value between 0 and 100.

Submit

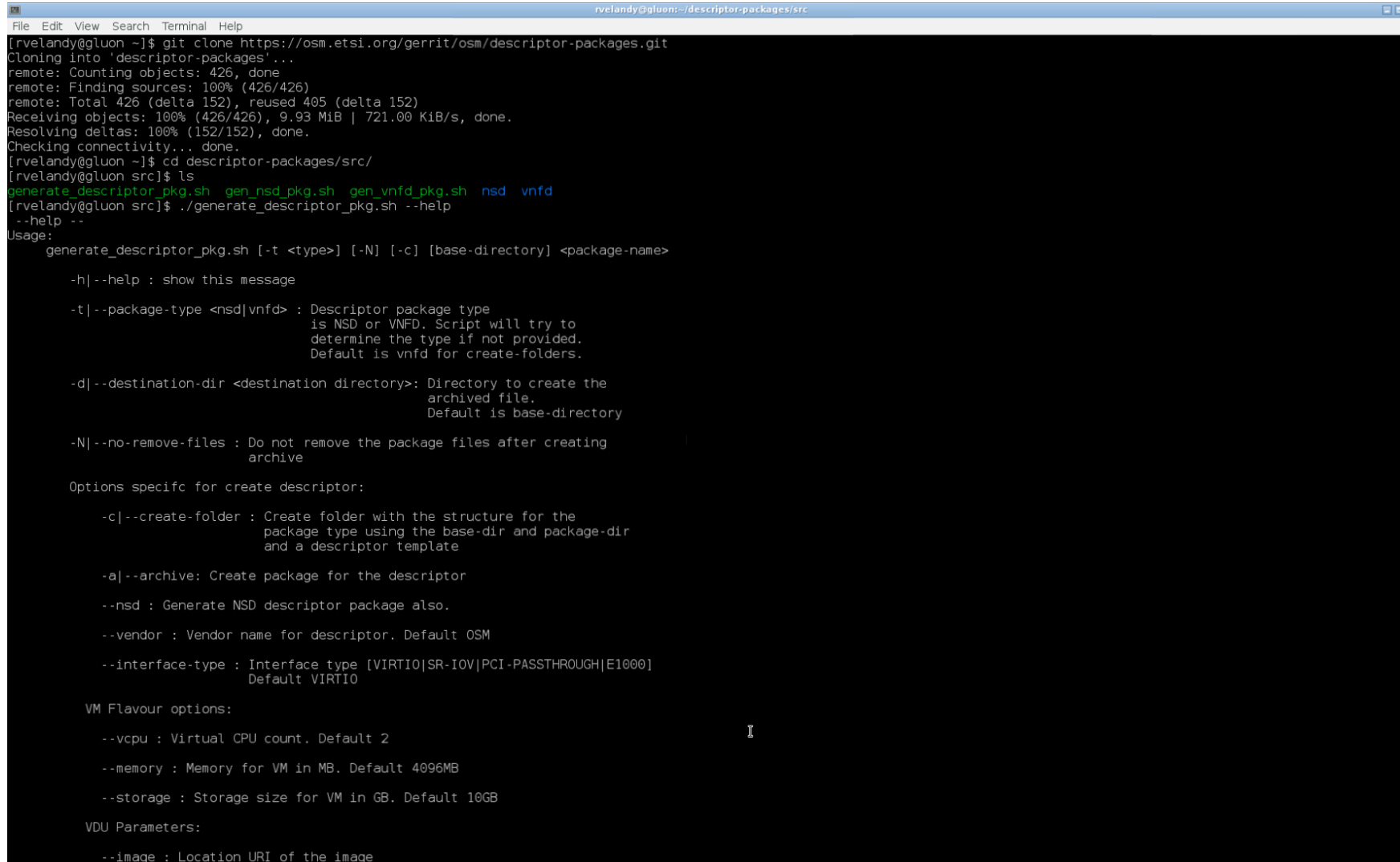
- Pros
  - Extremely easy to use and helps get started with OSM.
- Cons
  - Only supports simple VNFs as of now.
  - No support for additional artifacts within the package

- Utility is part of R1
- Web based form uses this utility
- Option to generate NS package
- Use cases:
  1. VNF/NS package including descriptors from scratch
  2. VNF/NS package after modifications to an existing VNF/NS package
    - Validates the package structure
    - Re-creates the checksums.txt file

- TO INSTALL
  - git clone <https://osm.etsi.org/gerrit/osm/descriptor-packages.git>
- TO USE
  - # cd descriptor-packages/src
  - # ./generate\_descriptor\_pkg.sh -help

- `# cd descriptor-packages/src`
- `# ./generate_descriptor_pkg.sh -help`

# PACKAGE CREATION UTILITY INSTALL – SCREENSHOT



```
File Edit View Search Terminal Help
rvelandy@gluon: ~/descriptor-packages/src

[rvelandy@gluon ~]$ git clone https://osm.etsi.org/gerrit/osm/descriptor-packages.git
Cloning into 'descriptor-packages'...
remote: Counting objects: 426, done
remote: Finding sources: 100% (426/426)
remote: Total 426 (delta 152), reused 405 (delta 152)
Receiving objects: 100% (426/426), 9.93 MiB | 721.00 KiB/s, done.
Resolving deltas: 100% (152/152), done.
Checking connectivity... done.
[rvelandy@gluon ~]$ cd descriptor-packages/src/
[rvelandy@gluon src]$ ls
generate_descriptor_pkg.sh  gen_nsd_pkg.sh  gen_vnfd_pkg.sh  nsd  vnfd
[rvelandy@gluon src]$ ./generate_descriptor_pkg.sh --help
--help --
Usage:
generate_descriptor_pkg.sh [-t <type>] [-N] [-c] [base-directory] <package-name>

-h|--help : show this message

-t|--package-type <nsd|vnfd> : Descriptor package type
                             is NSD or VNFD. Script will try to
                             determine the type if not provided.
                             Default is vnfd for create-folders.

-d|--destination-dir <destination directory>: Directory to create the
                                             archived file.
                                             Default is base-directory

-N|--no-remove-files : Do not remove the package files after creating
                       archive

Options specific for create descriptor:

-c|--create-folder : Create folder with the structure for the
                    package type using the base-dir and package-dir
                    and a descriptor template

-a|--archive: Create package for the descriptor

--nsd : Generate NSD descriptor package also.

--vendor : Vendor name for descriptor. Default OSM

--interface-type : Interface type [VIRTIO|SR-IOV|PCI-PASSTHROUGH|E1000]
                  Default VIRTIO

VM Flavour options:

--vcpu : Virtual CPU count. Default 2

--memory : Memory for VM in MB. Default 4096MB

--storage : Storage size for VM in GB. Default 10GB

VDU Parameters:

--image : Location URI of the image
```

- Create a VNF and NS package from scratch

```
# ./generate_descriptor_pkg.sh -c -a --nsd --vcpu <vcpu_cnt> --memory  
<memory_mb> --storage <storage_gb> --image <image-name> <base-dir>  
<vnf-name>
```

-c|--create-folder : Create folder with the structure for the package type

-a|--archive: Create package for the descriptor

--nsd: Generate NSD descriptor package as well for the VNF.

<base-dir>: Directory where the package is generated. If unspecified  
package is created in the current directory

# VNF/NS PACKAGE CREATION UTILITY – USE CASE 1 – EXAMPLE



- `./generate_descriptor_pkg.sh -ac --nsd --vcpu 4 --memory 2048 --storage 20 --image cirros-0.3.4.img /tmp/cirros cirros`

# VNF/NS PACKAGE CREATION UTILITY – USE CASE 1 – SCREENSHOT

```
rvelandy@gluon:~/descriptor-packages/src
File Edit View Search Terminal Help
^[[A^[[A[rvelandy@gluon src]$ ./generate_descriptor_pkg.sh -ac --nsd --vcpu 4 --memory 2048 --storage 20 --image cirros-0.3.4.img /tmp/cirros /tmp/cirros
-a -c --nsd --vcpu '4' --memory '2048' --storage '20' --image 'cirros-0.3.4.img' -- '/tmp/cirros' '/tmp/cirros'
/tmp/cirros/cirros_vnfd.tar.gz
/tmp/cirros/cirros_nsd.tar.gz
[rvelandy@gluon src]$ ls /tmp/cirros/
cirros_nsd  cirros_nsd.tar.gz  cirros_vnfd  cirros_vnfd.tar.gz
[rvelandy@gluon src]$ ls /tmp/cirros/cirros_vnfd
charms  cirros_vnfd.yaml  cloud_init  icons  images  README  scripts
[rvelandy@gluon src]$ ls /tmp/cirros/cirros_nsd
cirros_nsd.yaml  icons  ns_config  README  scripts  vnf_config
[rvelandy@gluon src]$ cat /tmp/cirros/cirros_vnfd/cirros_vnfd.yaml
vnfd:vnfd-catalog:
  vnfd:
    - id: cirros_vnfd
      name: cirros_vnfd
      short-name: cirros_vnfd
      description: Generated by OSM package generator
      vendor: OSM
      version: '1.0'

      # Place the logo as png in icons directory and provide the name here
      # logo: <update, optional>

      # Management interface
      mgmt-interface:
        vdu-id: cirros_vnfd-VM

      # Atleast one VDU need to be specified
      vdu:
      # Additional VDUs can be created by copying the
      # VDU descriptor below
      - id: cirros_vnfd-VM
        name: cirros_vnfd-VM
        description: cirros_vnfd-VM
        count: 1

      # Flavour of the VM to be instantiated for the VDU
      vm-flavor:
        vcpu-count: 4
        memory-mb: 2048
        storage-gb: 20

      # Image including the full path
      image: 'cirros-0.3.4.img'
```

- Create VND and NS package after Modification of an existing package.
  - Start with an existing package directory
  - Make necessary modifications to the descriptor
  - Copy artifacts (charm, icon etc ...) to the appropriate directory
  - Run one of the following commands to create the package with the new checksums.txt file.
    - `./generate_descriptor_pkg.sh -t vnfd <vnfd_directory>` for VNFDs
    - `./generate_descriptor_pkg.sh -t nsd <nsd_directory>` for NSDs

# VNF/NS PACKAGE CREATION UTILITY – USE CASE 2

## – SCREENSHOT

```
rvelandy@gluon:~/descriptor-packages/src
File Edit View Search Terminal Help
[rvelandy@gluon ~]$ cp descriptor-packages/src/vnfd/cirros_vnfd/icons/cirros-64.png /tmp/cirros/cirros_vnfd/icons/
[rvelandy@gluon ~]$ vi /tmp/cirros/cirros_vnfd/cirros_vnfd.yaml
[rvelandy@gluon ~]$ cat /tmp/cirros/cirros_vnfd/cirros_vnfd.yaml |grep logo
# Place the Logo as png in icons directory and provide the name here
Logo: cirros-64.png
[rvelandy@gluon ~]$ ./generate_descriptor_pkg.sh -t vnfd /tmp/cirros/cirros_vnfd
bash: ./generate_descriptor_pkg.sh: No such file or directory
[rvelandy@gluon ~]$ pwd
/net/homes/home1/rvelandy
[rvelandy@gluon ~]$ cd descriptor-packages/
[rvelandy@gluon descriptor-packages]$ ./generate_descriptor_pkg.sh -t vnfd /tmp/cirros/cirros_vnfd
bash: ./generate_descriptor_pkg.sh: No such file or directory
[rvelandy@gluon descriptor-packages]$ cd src/
[rvelandy@gluon src]$ ./generate_descriptor_pkg.sh -t vnfd /tmp/cirros/cirros_vnfd
-t 'vnfd' -- '/tmp/cirros/cirros_vnfd'
cirros_vnfd/
cirros_vnfd/checksums.txt
cirros_vnfd/README
cirros_vnfd/cirros_vnfd.yaml
cirros_vnfd/cloud_init/
cirros_vnfd/charms/
cirros_vnfd/icons/
cirros_vnfd/icons/cirros-64.png
cirros_vnfd/scripts/
cirros_vnfd/images/
[rvelandy@gluon src]$ date
Fri Sep 30 16:22:37 EDT 2016
[rvelandy@gluon src]$ ls -ltr /tmp/cirros/cirros_vnfd.tar.gz
-rw-rw-r-- 1 rvelandy rvelandy 5635 Sep 30 16:22 /tmp/cirros/cirros_vnfd.tar.gz
[rvelandy@gluon src]$ tar tvf /tmp/cirros/cirros_vnfd.tar.gz
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 16:22 cirros_vnfd/
-rw-rw-r-- rvelandy/rvelandy 105 2016-09-30 16:22 cirros_vnfd/checksums.txt
-rw-rw-r-- rvelandy/rvelandy 95 2016-09-30 15:59 cirros_vnfd/README
-rw-rw-r-- rvelandy/rvelandy 2698 2016-09-30 16:20 cirros_vnfd/cirros_vnfd.yaml
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 15:59 cirros_vnfd/cloud_init/
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 15:59 cirros_vnfd/charms/
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 16:18 cirros_vnfd/icons/
-rw-rw-r-- rvelandy/rvelandy 4134 2016-09-30 16:20 cirros_vnfd/icons/cirros-64.png
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 15:59 cirros_vnfd/scripts/
drwxrwxr-x rvelandy/rvelandy 0 2016-09-30 15:59 cirros_vnfd/images/
[rvelandy@gluon src]$
```

- Pros
  - Easy to use for simple VNFs
  - Ability to make updates to the descriptor in an existing package
  - Support for additional artifacts
  - Easy to get a starting template for complex VNFs
- Cons
  - More involved (For packages with artifacts or for complex VNFs)
    - Requires hand editing of the initial descriptor template using regular text editors

ACCOUNTS
CONFIG
LAUNCHPAD
**CATALOG**
ABOUT
DEBUG
LOGGING
LOGOUT

## COMPOSER

### DESCRIPTOR CATALOGS

NSD
**VNFD**

**myvnfd**  
myvnfd  
A simple VNF descriptor w/ one VDU  
v1.0

### MYVNFD

ONBOARD
CANCEL
YAML VIEWER
AUTO LAYOUT
ADD IVLD
ADD VDU
DELETE

**myvnfd**

**myvdu1**

**myvdu2**

MYVNFD / MYVDU2

myvdu2

NAME

Description

DESCRIPTION

VM FLAVOR

Vcpu Count

VCPU COUNT

Memory Mb

MEMORY MB

Storage Gb

STORAGE GB

Image

IMAGE

EXTERNAL INTERFACE 0 ITEMS

+

ADD

[MORE...](#)

CATALOG PACKAGE MANAGER

- Pros
  - Easy to use graphical Interface for generating descriptor only packages
  - Easier to generate descriptors for complex VNFs and NSes
- Cons
  - Does not yet support artifacts management
    - Requires to use the package utility to add additional artifacts to a package

- Get S0 code
  - # git clone <https://osm.etsi.org/gerrit/osm/S0.git>
  - # cd S0
  - # ./rwlaunchpad/plugins/rwlaunchpadtasklet/scripts/onboard\_pkg --help
- Alternatively, if you have the OSM R1 installed, the same utility will be available under /usr/rift/usr/bin/onboard\_pkg

- Usage
  - Onboarding
    - # onboard\_pkg -s <launchpad-ip> -u <package-name>  
-u can be repeated for multiple packages
- Uses rest interface for onboarding
- Example command to onboard vnfd and nsd in one shot
  - ./rwlaunchpad/plugins/rwlaunchpadtasklet/scripts/onboard\_pkg -s 10.64.1.175 -u cirros\_vnfd.tar.gz -u cirros\_nsd.tar.gz

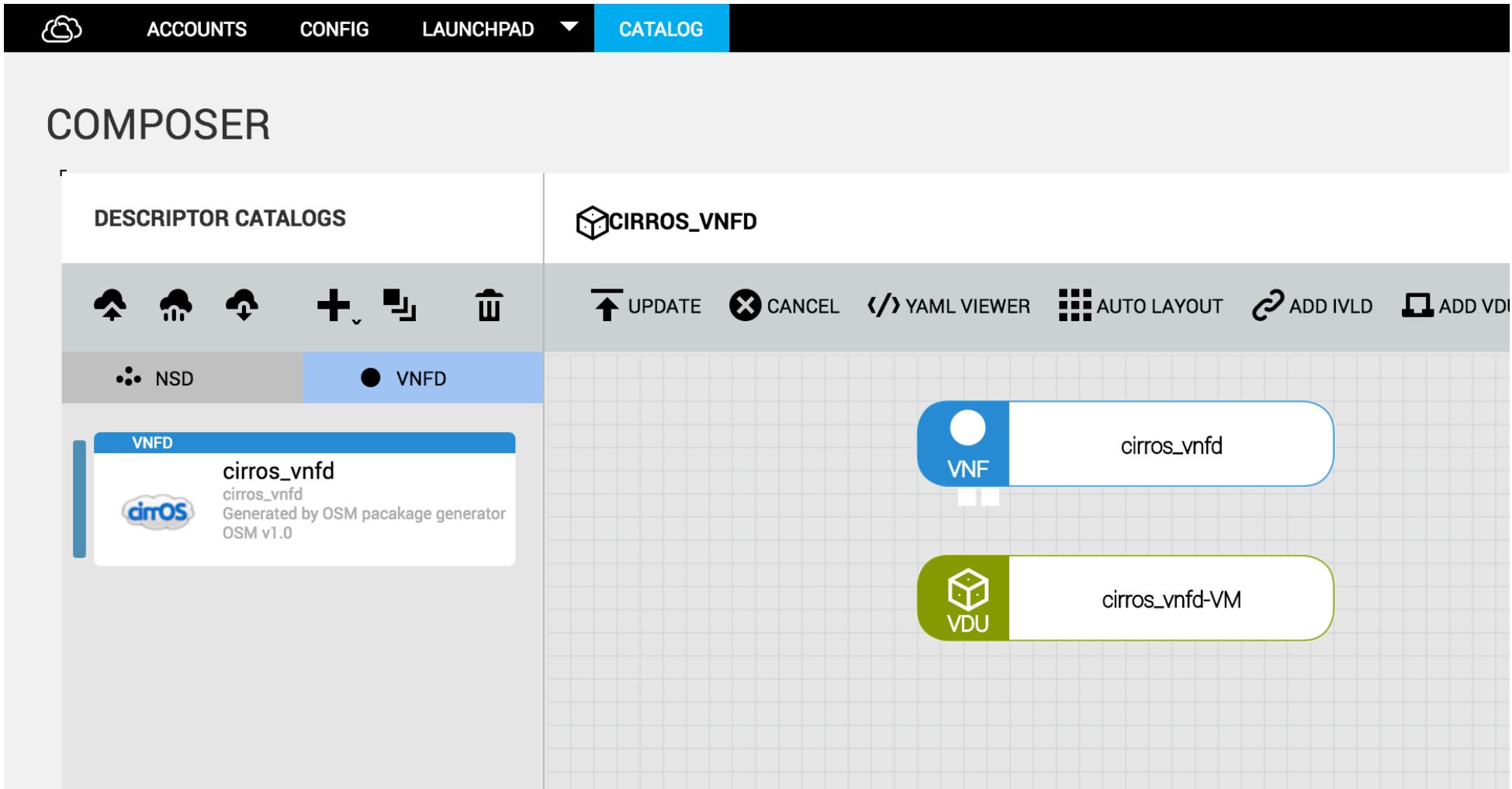
# PACKAGE ONBOARDING AND INSTANTIATION UTILITY – ONBOARDING – SCREENSHOT

```
rvelandy@gluon:tmp/S0
File Edit View Search Terminal Help
[rvelandy@gluon tmp]$ git clone https://osm.etsi.org/gerrit/osm/S0.git
Cloning into 'S0'...
remote: Counting objects: 1169, done
remote: Finding sources: 100% (1169/1169)
remote: Total 1169 (delta 443), reused 1140 (delta 443)
Receiving objects: 100% (1169/1169), 978.36 KiB | 888.00 KiB/s, done.
Resolving deltas: 100% (443/443), done.
Checking connectivity... done.
[rvelandy@gluon tmp]$ cd S0
[rvelandy@gluon S0]$ ls
build-dependencies BUILD.sh CMakeLists.txt common confd_client designs examples foss.txt Makefile manifest models README rwcsl rwcsl rwcsl rwcsl rwcsl rwcsl
[rvelandy@gluon S0]$ ./rwlanchpad/plugins/rwlanchpadtasklet/scripts/onboard_pkg --help
usage: onboard_pkg [-h] [-s S0_IP] [-u UPLOAD_PKG] [-i INSTANTIATE]
                  [-d NSD_ID] [-D DATACENTER] [-c VIM_ACCOUNT]
                  [-p UPLOAD_PORT] [-P RESTCONF_PORT]
                  [--restconf-user RESTCONF_USER]
                  [--restconf-password RESTCONF_PASSWORD] [-v]

Upload and instantiate NS

optional arguments:
  -h, --help            show this help message and exit
  -s S0_IP, --so-ip S0_IP
                        S0 Launchpad IP
  -u UPLOAD_PKG, --upload-pkg UPLOAD_PKG
                        Descriptor packages to upload. If multiple descriptors
                        are provided, they are uploaded in the same sequence.
  -i INSTANTIATE, --instantiate INSTANTIATE
                        Instantiate a network service with the name
  -d NSD_ID, --nsd-id NSD_ID
                        Network descriptor ID to instantiate
  -D DATACENTER, --datacenter DATACENTER
                        OpenMano datacenter to instantiate on
  -c VIM_ACCOUNT, --vim-account VIM_ACCOUNT
                        Cloud/VIM account to instantiate on
  -p UPLOAD_PORT, --upload-port UPLOAD_PORT
                        Upload port number, default 4567
  -P RESTCONF_PORT, --restconf-port RESTCONF_PORT
                        RESTconf port number, default 8888
  --restconf-user RESTCONF_USER
                        RESTconf user name, default admin
  --restconf-password RESTCONF_PASSWORD
                        RESTconf password, default admin
  -v, --verbose          Show more logs
[rvelandy@gluon S0]$ ./rwlanchpad/plugins/rwlanchpadtasklet/scripts/onboard_pkg -s 10.64.1.175 -u /tmp/cirros/cirros_
cirros_nsd/ cirros_nsd.tar.gz cirros_vnfd.tar.gz
[rvelandy@gluon S0]$ ./rwlanchpad/plugins/rwlanchpadtasklet/scripts/onboard_pkg -s 10.64.1.175 -u /tmp/cirros/cirros_vnfd.tar.gz -u /tmp/cirros/cirros_nsd.tar.gz
2016-09-30 16:51:20.589 INFO (onboard-pkg@34218:onboard_pkg:263) - Upload of package /tmp/cirros/cirros_vnfd.tar.gz succeeded
INFO:onboard-pkg:Upload of package /tmp/cirros/cirros_vnfd.tar.gz succeeded
2016-09-30 16:51:21.396 INFO (onboard-pkg@34218:onboard_pkg:263) - Upload of package /tmp/cirros/cirros_nsd.tar.gz succeeded
INFO:onboard-pkg:Upload of package /tmp/cirros/cirros_nsd.tar.gz succeeded
[rvelandy@gluon S0]$
```

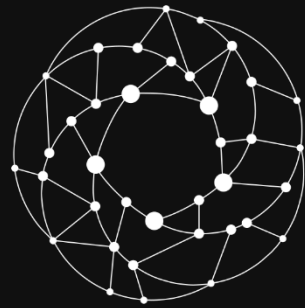
# PACKAGE ONBOARDING AND INSTANTIATION UTILITY – ONBOARDING – UI



- Once onboarded, instantiation can be triggered from the UI or using the utility
- To instantiate from the utility

```
# onboard_pkg -d <NSD_ID> -D <DATACENTER>
```

  - d – NSD ID of the onboarded NS
  - d DATACENTER ID



Open Source  
**MANO**

**THANK YOU**