



Department of Computer Science  
Computer Networks Research Group

## Technology Review



### Management of ServiCes Across MultipLE clouds

#### Authors:

ARKAJIT DHAR  
ASHWIN PRASAD SHIVARPATNA VENKATESH  
BHARGAVI MOHAN  
DEEKSHA MYSORE RAMESH  
HARSHITHA PANDITHANAHALLI SOMASHEKARAIH  
SANKET KUMAR GUPTA  
SUHEEL SHRIRANGAPURA NAZEERSAB  
VIVEK JAGANATH

#### Supervisors:

Prof. Dr. Holger Karl | Sevil Dräxler | Hadi Razzaghi Kouchaksaraei

Paderborn, January 3, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Open Source MANO</b>	<b>4</b>
2.1	Configuration requirements needed to run Open Source MANO (OSM) Release FOUR is a single server or VM: . . . . .	4
2.2	Open Source Mano Installation . . . . .	4
2.2.1	Steps for Installation: . . . . .	4
2.2.2	verifying installation from the OSM GUI: . . . . .	4
2.3	VIM Installation . . . . .	5
2.3.1	Steps to install openstack using devstack are as follows: . . . . .	5
2.4	Configure openstack for OSM . . . . .	6
2.5	Deploying Network Service . . . . .	9
<b>3</b>	<b>OpenBaton</b>	<b>11</b>
<b>4</b>	<b>SONATA/PISHAHANG</b>	<b>12</b>
4.1	Configuration requirements to run Pishahang on a single server or VM . . . . .	12
4.2	OpenStack Installation (Ocata) . . . . .	12
4.3	Pishahang installation . . . . .	13
4.4	Service Descriptor Packaging and uploading . . . . .	14
4.5	Linking VIM to sonata . . . . .	17
4.6	Onboarding Descriptors . . . . .	20
4.7	Network Service Instantiation . . . . .	21
<b>5</b>	<b>Conclusion</b>	<b>22</b>
<b>A</b>	<b>Appendix</b>	<b>23</b>

## List of Figures

2.1	OSM GUI . . . . .	5
2.2	Open Stack Dashboard . . . . .	6
2.3	Creating a Network in Openstack . . . . .	7
2.4	creating a valid tenant/user in openstack . . . . .	7
2.5	creating a valid tenant/user in openstack . . . . .	7
2.6	Uploading VM image to VIM in openstack . . . . .	8
2.7	Adding VIMs to OSM . . . . .	9
2.8	Adding VIMs to OSM . . . . .	9
2.9	On-boarding of VNFD in OSM . . . . .	9
2.10	On-boarding of NS in OSM . . . . .	10
2.11	Initiating of NS in OSM . . . . .	10
4.1	Sonata Dashboard . . . . .	14
4.2	Add WIM . . . . .	15
4.3	Add VIM . . . . .	15
4.4	Select Router . . . . .	16
4.5	Select IDs . . . . .	16
4.6	VIM Details . . . . .	17
4.7	Create user . . . . .	18
4.8	Edit project quotas . . . . .	19
4.9	Create router . . . . .	20
4.10	REST call for NSD . . . . .	20
4.11	REST call for VNFD . . . . .	21

## List of Tables



## Introduction

The aim of technology review was to understand the working of all the tools and technologies relevant for the project.

To achieve this, tasks was assigned to each of the team to review few set of tools as below.

Technology Review		
Team	Members	Reviewed
1	Arkajit Dhar	Open Source Mano
	Suheel Nazeersab	
	Vivek Jaganath	
2	Sanket Kumar	Open Baton
	Harshitha Somashekaraiah	
3	Ashwin Prasad	Sonata
	Bhargavi Mohan	
	Deeksha Ramesh	

MANO frameworks such as Open Source MANO, Sonata and Open Baton has been reviewed in this phase. We have also tried and worked on virtual infrastructure managers such as Kubernetes and Open Stack.

The MANO frameworks are up and running in virtual machines and we have established connections between MANO and VIM's.

The detailed explanation on installation and steps is given below in the document.

# 2

## Open Source MANO

### 2.1 Configuration requirements needed to run Open Source MANO (OSM) Release FOUR is a single server or VM:

- MINIMUM: 2 CPUs, 4 GB RAM, 20GB disk and a single interface with Internet access
- RECOMMENDED: 2 CPUs, 8 GB RAM, 40GB disk and a single interface with Internet access
- Ubuntu16.04 (64-bit variant required) as base image (<http://releases.ubuntu.com/16.04/>)

### 2.2 Open Source Mano Installation

#### 2.2.1 Steps for Installation:

- Downloading latest version of OSM

---

```
wget https://osm-download.etsi.org/ftp/osm-4.0-four/install_osm.sh
```

---

- Installing OSM

---

```
chmod +x install_osm.sh  
$ ./install_osm.sh 2>&1 | tee osm_install_log.txt
```

---

#### 2.2.2 verifying installation from the OSM GUI:

- Accessing GUI:

---

Access <http://1.2.3.4>, replacing 1.2.3.4 with the IP address of your host.  
Login using Userid : admin , password : admin

---

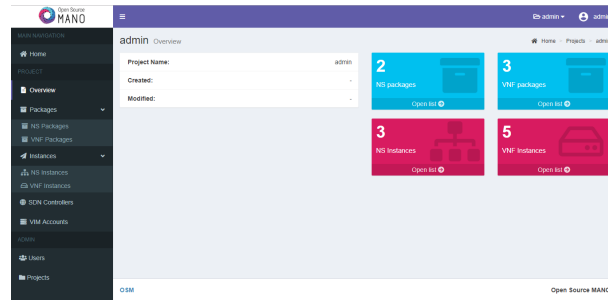


Figure 2.1: OSM GUI

- Verify 10 docker containers were created:

---

```
docker stack ps osm |grep -i running
docker service ls
```

---

## 2.3 VIM Installation

### 2.3.1 Steps to install openstack using devstack are as follows:

- Create a user “stack”

---

```
sudo useradd -s /bin/bash -d /opt/stack -m stack
echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
sudo su -stack
```

---

- Clone the devstack repository

---

```
git clone https://git.openstack.org/openstack-dev/devstack
cd devstack
```

---

- Create and configure the local.conf file

---

```
[[local|localrc]]
ADMIN_PASSWORD=password
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
```

---

- Execute the command

---

```
./stack.sh
```

---

- After installation check and verify from openstack horizon GUI:

---

Access <http://1.2.3.4>, replacing 1.2.3.4 with the IP address of your host.  
Login using Userid : admin , password : admin

---

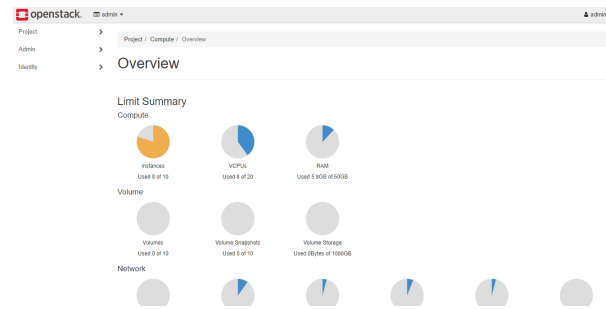


Figure 2.2: Open Stack Dashboard

## 2.4 Configure openstack for OSM

- Guarantee that Openstack API endpoints are reachable from OSM (particularly from RO container):

---

Login to openstack api access from the horizon gui.  
 Click on DOWNLOAD OPENSTACK RC FILE (api version 3).  
 Copy the OS\_AUTH\_URL variable value.  
 Paste in the browser or do a curl from the VM where OSM is installed to check its reachability.

---

- Create a management network, with DHCP enabled, reachable from OSM (particularly from VCA container)

---

Login to openstack horizon gui.  
 Go to admin-> create network.  
 Give the project name as your project ( default:admin)  
 Give a network name -> mgmt.  
 Give a network subnet name and network address (10.208.1.0/24).  
 Keep the Network Address source as 'ENTER NETWORK ADDRESS MANUALLY'.  
 Keep Gateway IP blank.  
 In Allocation Pools, give the IPs: start=10.208.0.2,end=10.208.0.254.  
 Leave DNS Name servers and Host Routes blank and click create.

---

- creating a valid tenant/user

---

Login to openstack horizon gui.  
 Go to identity-> create user.  
 Give the project name as your project ( default:admin)  
 Give a user name -> tenant.  
 Give the role also as admin and click create.

---



Create Network

Network

Subnet

Subnet Details

Name

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Project

Select a project

Provider Network Type

Local

☒ Enable Admin State

☐ Shared

☐ External Network

☒ Create Subnet

Availability Zone Hints

nova

Cancel

Back

Next

Figure 2.3: Creating a Network in Openstack

Project

Admin

Identity

Users

Users

Filter

Create User

Enable Users

Groups	Displaying 10 items	User Name	Description	Email	User ID	Enabled	Domain Name	Actions
<input type="checkbox"/>	nova	-			1ccdc9f8c4d41028954a295a294f	Yes	Default	Edit
<input type="checkbox"/>	all_demo	-	all_demo@example.com		1d221e1ccae4298184a220148d8b7	Yes	Default	Edit
<input type="checkbox"/>	cloud	-			31644c1d8c24d78a7963055a7230	Yes	Default	Edit

Figure 2.4: creating a valid tenant/user in openstack

Domain ID

default

Domain Name

Default

User Name

Description

Email

Password

Confirm Password

Primary Project

Select a project

Role

member

☒ Enabled

Description:

Create a new user and set related properties including the Primary Project and Role.

Cancel

Create User

Figure 2.5: creating a valid tenant/user in openstack

## 2.4 CONFIGURE OPENSTACK FOR OSM

- Uploading VM image(s) to the VIM(s)

---

Download the image from the following link:

(\hyperlink{name}{http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86\_64-disk.img})

Login to openstack horizon gui.

Go to admin -> Compute -> Images and click on create image.

Give the image name 'cirros034'

Upload the downloaded image file in step 1.

Choose the image format as QCOW2 : QEMU Emulator

Click on create image.

---

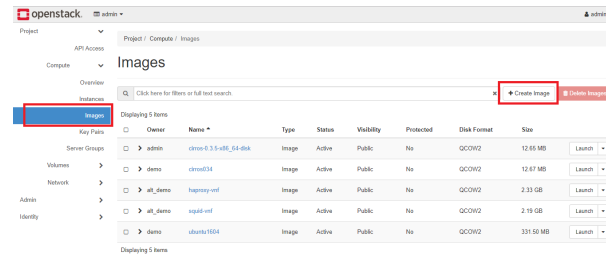


Figure 2.6: Uploading VM image to VIM in openstack

- Adding VIMs to OSM

---

Login to OSM and click on VIM Accounts.

Click on new VIM.

Give a name to your VIM instance and choose openstack from the type

dropdown. Give the VIM URL as the OS\\_AUTH\\_URL variable value in openstack's rc file.

Enter the VIM userid and password as the login userid and password for openstack horizon gui.

Give the tenant name as admin/tenant.

Click on create.

---

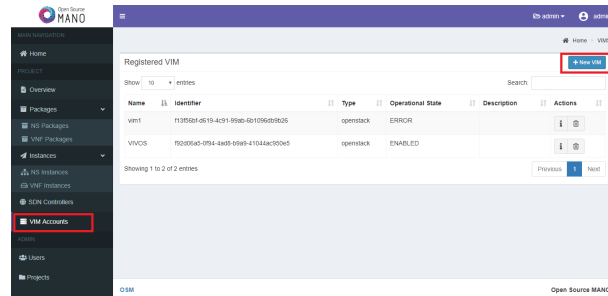


Figure 2.7: Adding VIMs to OSM

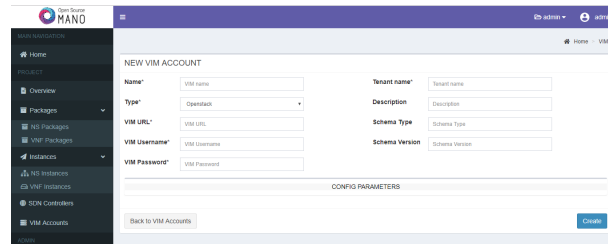


Figure 2.8: Adding VIMs to OSM

## 2.5 Deploying Network Service

First download the required VNF and NS packages from this URL: ([https://osm-download.etsi.org/ftp/osm-3.0-three/examples/cirros\\_2vnf\\_ns/](https://osm-download.etsi.org/ftp/osm-3.0-three/examples/cirros_2vnf_ns/))

- On-boarding a VNFD

---

From the UI , Go to Projects --> Admin --> VNF Packages (Open List)  
 Click on the Onboard VNFD button  
 Drag and drop the VNF package file `cirros_vnf.tar.gz` in the importing area.

---

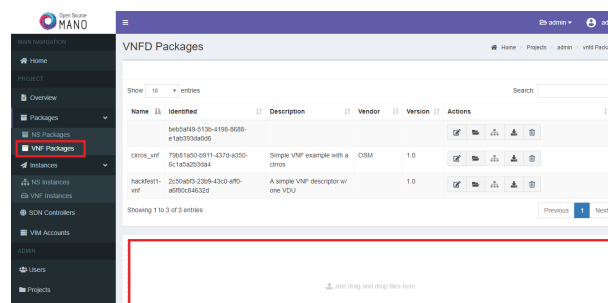


Figure 2.9: On-boarding of VNFD in OSM

- Onboarding a NS

---

From the UI, Go to Projects --> Admin --> NS Packages (Open List)  
 Click on the Onboard NSD button  
 Drag and drop the NS package file `cirros_2vnf_ns.tar.gz` in the importing area.

---

## 2.5 DEPLOYING NETWORK SERVICE

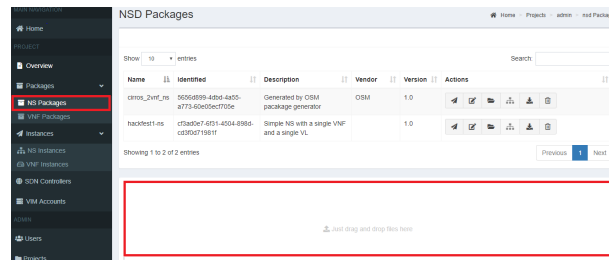


Figure 2.10: On-boarding of NS in OSM

- Instantiating the NS

---

From the UI, Go to Projects --> Admin --> NS Packages (Open List)  
Next the NS descriptor to be instantiated, click on Launch  
Fill the form, adding at least a name and selecting the VIM

---

New Instance

Name \*

Ns name

Description \*

Description

Nsd Id \*

Select NSD

Vim Account Id \*

Select VIM

SSH Key

Paste your key here...

Config

Yaml config

Cancel

Create

Figure 2.11: Initiating of NS in OSM

3

OpenBaton

# SONATA/PISHAHANG

## 4.1 Configuration requirements to run Pishahang on a single server or VM

- Operating System: Ubuntu 16.04 as base image (<http://releases.ubuntu.com/16.04/>)
- Minimum Requirements: 4GB RAM, 40GB hard disk and a non-root user account

## 4.2 OpenStack Installation (Ocata)

We set up an OpenStack environment using DevStack, which is installed via a configuration file named local.conf. The installation guide can also be found at <https://docs.openstack.org/devstack/latest/>

- Other references <sup>1</sup> <sup>2</sup>

### Steps of installation:

- Create a user “stack”

---

```
sudo useradd -s /bin/bash -d /opt/stack -m stack
echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
sudo su - stack
```

---

- Clone the devstack repository

---

```
git clone https://git.openstack.org/openstack-dev/devstack -b stable/ocata
cd devstack
```

---

---

<sup>1</sup>Refer DevStack heat documentation to enable heat service

<sup>2</sup>Refer DevStack networking-sfc documentation for service chaining

- Create and configure the local.conf file

---

```
[[local|localrc]]
ADMIN\_PASSWORD=password
DATABASE\_PASSWORD=$ADMIN\_PASSWORD
RABBIT\_PASSWORD=$ADMIN\_PASSWORD
SERVICE\_PASSWORD=$ADMIN\_PASSWORD
```

---

- Execute the command

---

```
./stack.sh
```

---

- After installation check and verify from openstack horizon GUI

Access <http://1.2.3.4>, replace 1.2.3.4 with the IP address of your host Login using user id: admin, password: admin

## 4.3 Pishahang installation

The Below steps of installation are performed from the non-root user account

- Installing packages

---

```
sudo apt-get install -y software-properties-common
sudo apt-add-repository -y ppa:ansible/ansible
sudo apt-get update
sudo apt-get install -y git ansible
```

---

- Clone repository

---

```
git clone https://github.com/CN-UPB/Pishahang.git cd Pishahang/son-install
echo sonata | tee ~/.ssh/.vault_pass
```

---

- Start Installation, replace "<your\_ip4\_address>" with the IP address where SONATA should be available.

---

```
ansible-playbook utils/deploy/sp.yml -e "target=localhost \
public_ip=<your_ip4_address>" -v
```

---

- Verify Installation

Open your browser and navigate to [http://public\\_ip](http://public_ip). Login using the username sonata and password 1234. If the installation was successful, you should now see the dashboard of the service platform

- Installation of son-cli The SONATA CLI toolset can also be installed via the Python setup script

---

```
git clone https://github.com/sonata-nfv/son-cli.git
cd son-cli
python3 setup.py install
```

---

- Test if its working by invoking

---

```
son-workspace -h
son-package -h
son-publish -h
son-push -h
son-monitor -h
```

---

Reference Link - <https://github.com/sonata-nfv/son-cli#all-dists-using-setuptools>

## 4.4 Service Descriptor Packaging and uploading

We also need the son-cli to be installed and son-examples repository to be cloned in the environment

- Add WIM
  - Open your browser and navigate to [http://public\\_ip](http://public_ip)
  - Open the "WIM/VIM Settings" tab

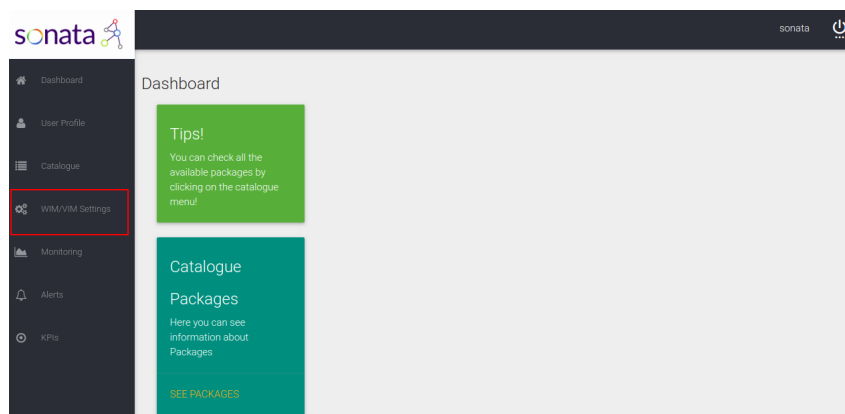


Figure 4.1: Sonata Dashboard

- click on add a WIM
- Select "Mock" WIM vendor
- Enter any WIM name(e.g. Sonata Test), WIM address(e.g. local host), username(e.g., Sonata) and password(e.g. 1234)
- Confirm by clicking "SAVE"



**New Wim**

Wim Name  
SonataTest

Wim vendor  
MOCK

Wim address  
localhost

Username  
Sonata

Password  
....

CANCEL SAVE

Figure 4.2: Add WIM

- Adding OpenStack VIM
  - Click on add a VIM
  - Enter the VIM name(e.g. DevStack ) , select the WIM just created, enter the country(e.g. germany) and city(Paderborn)
  - Select "Heat" VIM vendor

**New Vim**

**General Configuration**

VIM Name  
DevStavk

Select WIM  
e1265246-51e7-491e-9134

Country  
Germany

City  
Paderborn

**Compute Configuration**

Vim Vendor  
Heat

**Networking Configuration**

Network configuration VIM Type  
Select vim vendor

CANCEL SAVE

Figure 4.3: Add VIM

- Tenant ID: DevStack project id (e.g. sonatademo), Tenant External Network ID: DevStack ID of the public network and Tenant External Router ID: DevStack ID of the router created under sonatademo user i.e. sonata-router as shown below

## 4.4 SERVICE DESCRIPTOR PACKAGING AND UPLOADING

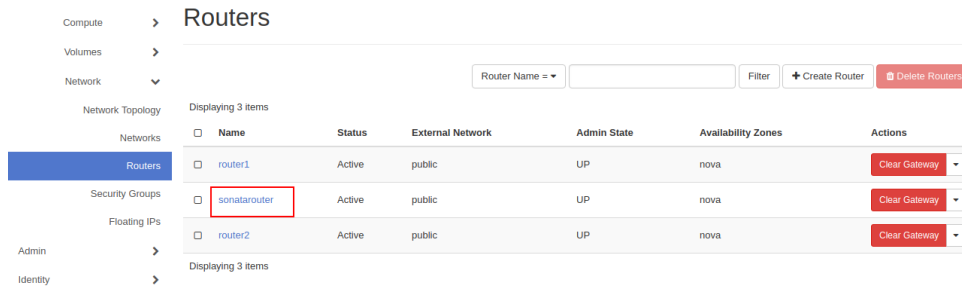


Figure 4.4: Select Router

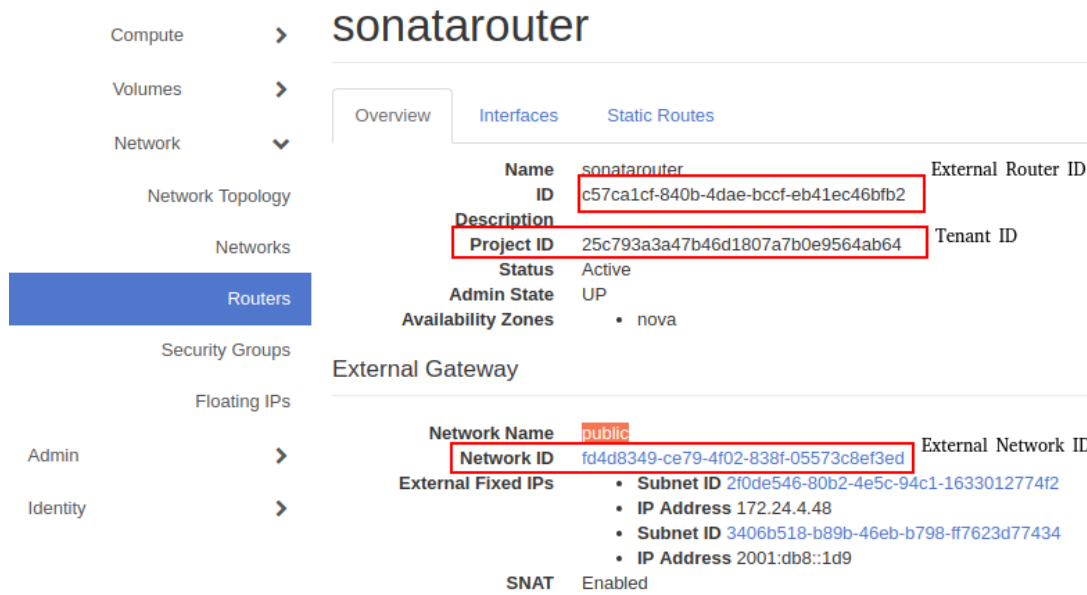


Figure 4.5: Select IDs

- VIM Address: DevStack (131.234.29.34)
- Vim Vendor: “OVS”, Username: sonatademo, Password: password of the user sonatademo (e.g. sonata), Domain: Default
- Click on "Save"

Compute Configuration		Networking Configuration	
Vim Vendor		Network configuration VIM Type	
Heat		Select vim vendor	
Tenant ID	Vim address	OVS	
e3cf6987c0ff43c48	131.234.29.112	Vim address	
Tenant External Network ID	Tenant External Router ID	131.234.29.112	
fd4d8349-ce79-4f02	c57ca1cf-840b-4da1	Username	
sonatademo	*****	Password	
Domain		sonatademo	
Default		*****	
		CANCEL SAVE	

Figure 4.6: VIM Details

#### – On-boarding Service Package

```
git clone https://github.com/sonata-nfv/son-examples.git
son-workspace --init
son-validate --project son-examples/service-projects/sonata-demo
son-package --project son-examples/service-projects/sonata-demo -n \
    service_package
son-access config --platform_id ServicePlatform --new --url \
    http://131.234.29.102 --default
son-access auth -u sonata -p 1234
son-access push --upload service_package.son
```

Reference video - <https://www.youtube.com/watch?v=RsXUIt4rzF0>

## 4.5 Linking VIM to sonata

Login to the DevStack dashboard: <http://131.234.29.34/dashboard>. There are two users created during installation admin and demo. Password for both users is sonata

- Create New User and Project

- Login as admin user in domain Default and create new user (e.g. sonatademo)
- In the menu, go to Identity->User (Create User)
- Give the admin role to the new user

## Create User

**Domain ID**

default

**Domain Name**

Default

**User Name** \*

sonatademo

**Description**

**Email**

**Password** \*

\*\*\*\*\*

**Confirm Password** \*

\*\*\*\*\*

**Primary Project**

sonatademo

**Role**

admin

☒ Enabled

**Description:**

Create a new user and set related properties including the Primary Project and Role.

Cancel

Create User

Figure 4.7: Create user

- Add a new project with the below details
  - Project name/tenant name: sonatademo
  - Allocate maximum number of resources for that project under Quotas tab

**Edit Quotas** ✕

**Compute \*** **Volume \*** **Network \***

**Instances \*** 100|

**VCPUs \*** 100

**RAM (MB) \*** 51200

**Metadata Items \*** 128

**Key Pairs \*** 100

**Server Groups \*** 10

**Server Group Members \*** 10

**Injected Files \*** 50

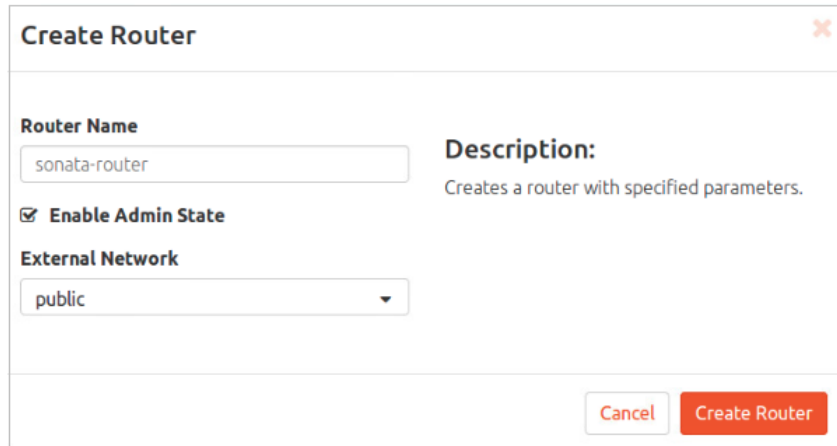
**Injected File Content (Bytes) \*** 52100

**Length of Injected File Path \*** 255

Cancel Save

Figure 4.8: Edit project quotas

- Create Private Network
  - Login as new user(e.g. sonatademo)
  - Create a network(e.g. sonata-priv) and add the subnet as well (e.g. sonata-priv-sub)
  - Add the router
  - Use any private network address, for example 192.168.x.0/24. While creating the router select the External Network as public (Error: Reference source not found). Add the sonata-priv-sub as the interface to the router



**Create Router**

**Router Name**  
sonata-router

**Description:**  
Creates a router with specified parameters.

☒ **Enable Admin State**

**External Network**  
public

**Buttons:** Cancel, Create Router

Figure 4.9: Create router

## 4.6 Onboarding Descriptors

NSD can be pushed to the server by using REST API provided by pishahang.

- For CSDs: [http://public\\_ip:4002/catalogues/api/v2/csds](http://public_ip:4002/catalogues/api/v2/csds)
- For COSDs: [http://public\\_ip:4002/catalogues/api/v2/complex-services](http://public_ip:4002/catalogues/api/v2/complex-services)
- Dummy NSD that has been uploaded can be seen in the appendix ??

Postman could be used to make the REST calls

### 1. NSD

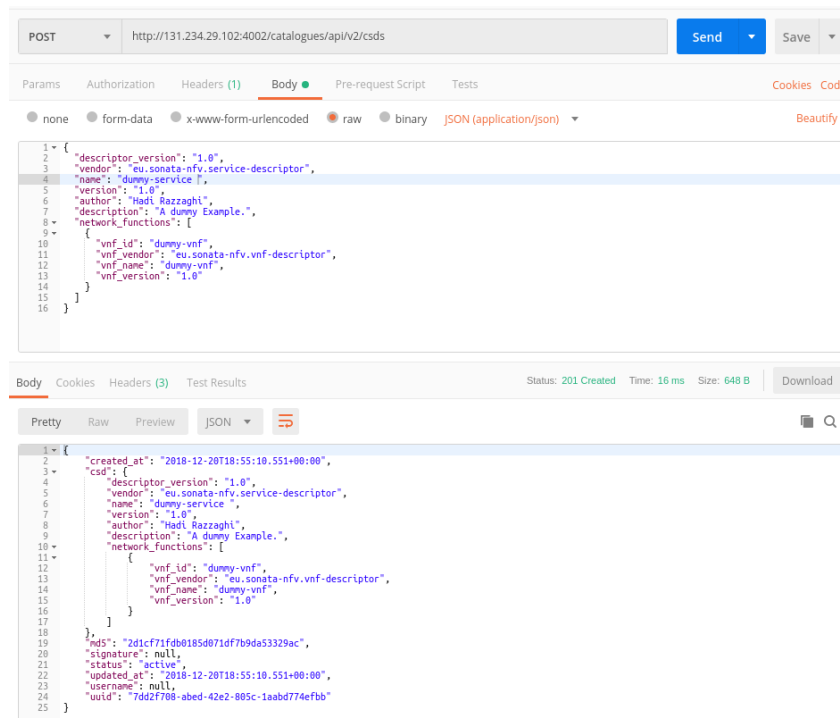


Figure 4.10: REST call for NSD

## CHAPTER 4. SONATA/PISHAHANG

### 2. VNFD

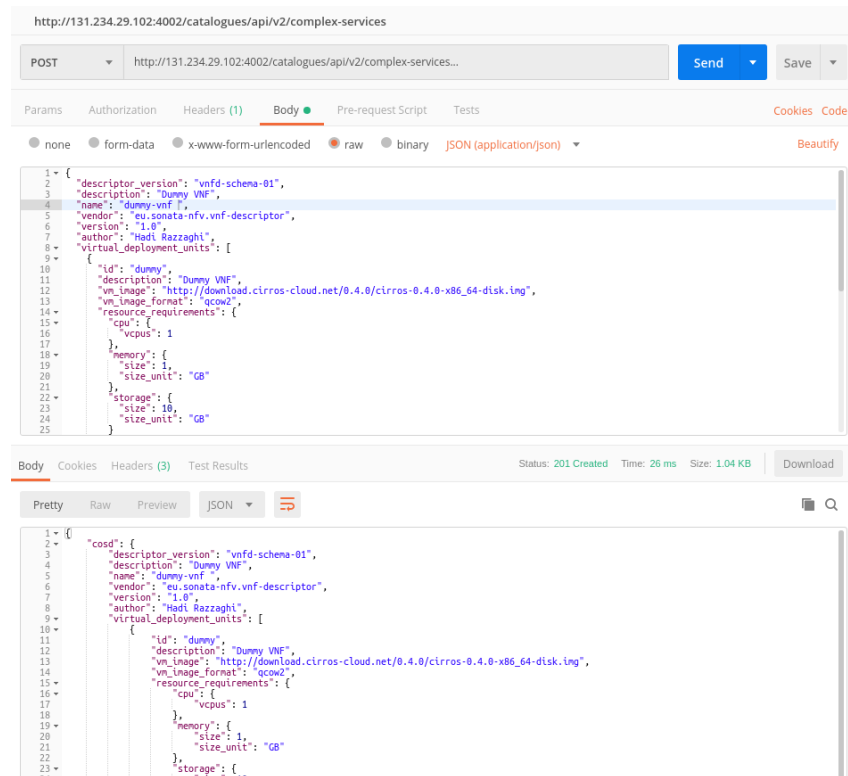


Figure 4.11: REST call for VNFD

## 4.7 Network Service Instantiation

- Open your browser and navigate to `http://public_ip:25001`
- Open the "Available Complex Services" tab
- Click the "Instantiate" button of the service you want to deploy
- Confirm the instantiate modal (ingress and egress can be empty)

## Conclusion

The main goal of this phase was to acquire practical experience on MANO frameworks and its workflow. All the steps required to instantiate a network service has been completed and verified.

Below we define the findings from the technology review phase.

- When we consider an environment with single domain, all of the network resources and services are managed by a single MANO orchestrator. However considering a multi-domain environment, where network services need to be deployed across multiple and different orchestrators, there is a need for a seamless communication between different orchestrators in order to deploy the end-to-end service successfully. Currently a major hindrance in the communication roots from the fact that each MANO framework uses different descriptor formats for describing the network service and virtual network function. One of our goal in this project is to overcome this shortcoming and implement a translator engine to translate the NSD and VNFD and facilitate communication between different orchestrators in a multi-domain environment.
- 
- Currently, there are no means to add a MANO adopter to a main MANO instance which can communicate with other MANO frameworks to instantiate and monitor services running on them. The ability to do inter framework hierarchical orchestration is missing. Adding such an adopter will enable the MANO instances to scale according to the number of service requests. Therefore, implementing a MANO adaptor to tackle this issue is one of the milestones during the course of this project. MANO adopters for SONATA and OSM will be implemented first and OpenBaton would be considered in the next phase.



# A

## Appendix

Listing A.1: NSD

```
1 {
2   "descriptor_version": "1.0",
3   "vendor": "eu.sonata-nfv.service-descriptor",
4   "name": "dummy-service",
5   "version": "1.0",
6   "author": "Hadi Razzaghi",
7   "description": "A dummy Example.",
8   "network_functions": [
9     {
10      "vnf_id": "dummy-vnf",
11      "vnf_vendor": "eu.sonata-nfv.vnf-descriptor",
12      "vnf_name": "dummy-vnf",
13      "vnf_version": "1.0"
14    }
15  ]
16 }
```

Listing A.2: VNFD

```
1 {
2   "descriptor_version": "vnfd-schema-01",
3   "description": "Dummy VNF",
4   "name": "dummy-vnf",
5   "vendor": "eu.sonata-nfv.vnf-descriptor",
6   "version": "1.0",
7   "author": "Hadi Razzaghi",
8   "virtual_deployment_units": [
9     {
10      "id": "dummy",
11      "description": "Dummy VNF",
12    }
13  ]
14 }
```

```

13     "vm_image":
14         "http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img",
15     "vm_image_format": "qcow2",
16     "resource_requirements": {
17         "cpu": {
18             "vcpus": 1
19         },
20         "memory": {
21             "size": 1,
22             "size_unit": "GB"
23         },
24         "storage": {
25             "size": 10,
26             "size_unit": "GB"
27         }
28     },
29     "connection_points": [
30         {
31             "id": "eth0",
32             "interface": "ipv4",
33             "type": "internal"
34         },
35         {
36             "id": "eth1",
37             "interface": "ipv4",
38             "type": "internal"
39         },
40         {
41             "id": "eth2",
42             "interface": "ipv4",
43             "type": "internal"
44         }
45     ],
46     "user_data": {
47         "password": "1234",
48         "chpasswd": {
49             "expire": false
50         },
51         "ssh_pwauth": true
52     }
53 ]
54 }

```