Open Source MANO

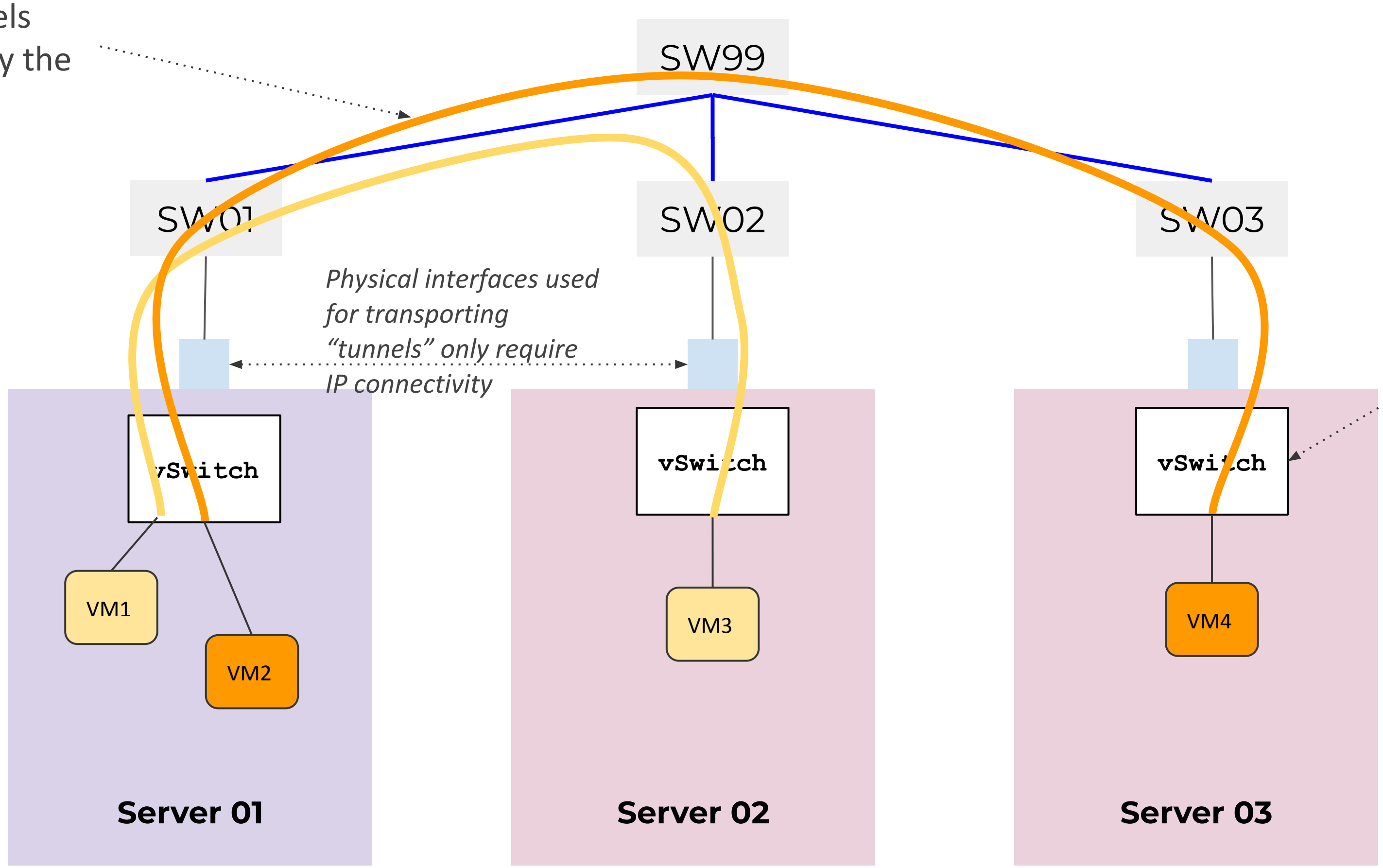
# OSM#9 Hackfest

## Underlay Automation with SDN Assist

Gianpietro Lavado (Whitestack)

# Using Virtual Interfaces (VIRTIO)

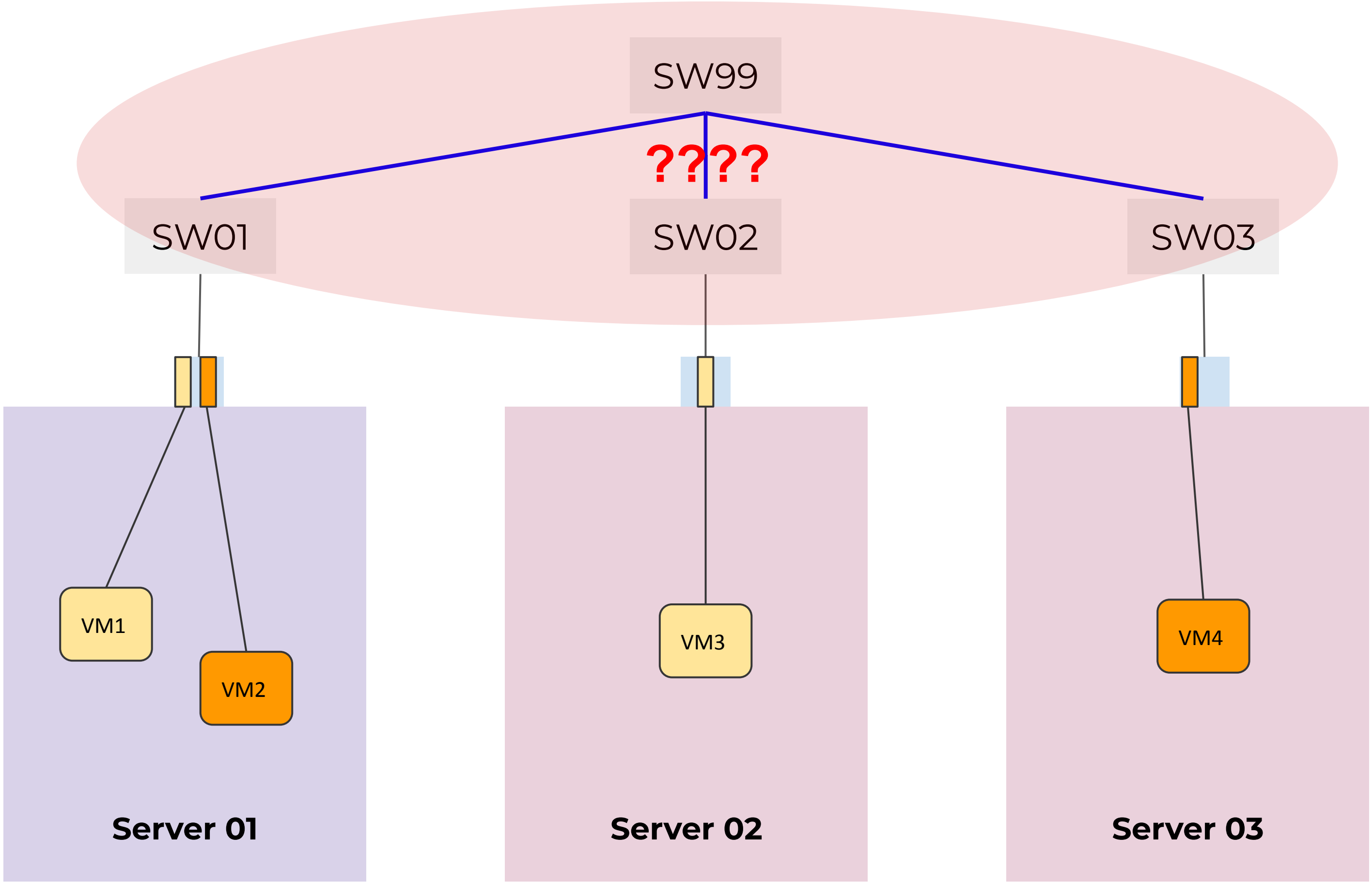
VXLAN Tunnels (controlled by the VIM)



*Physical interfaces used for transporting "tunnels" only require IP connectivity*

vSwitches like OVS have limited performance (~1Gbps), unless OVS-DPDK or similar techniques are used (~10Gbps)

# Using Physical Interfaces (SR-IOV/PASSTHROUGH)



SR-IOV and Passthrough features expose the instance directly to the physical NIC, so **who takes care of the end-to-end connectivity?**

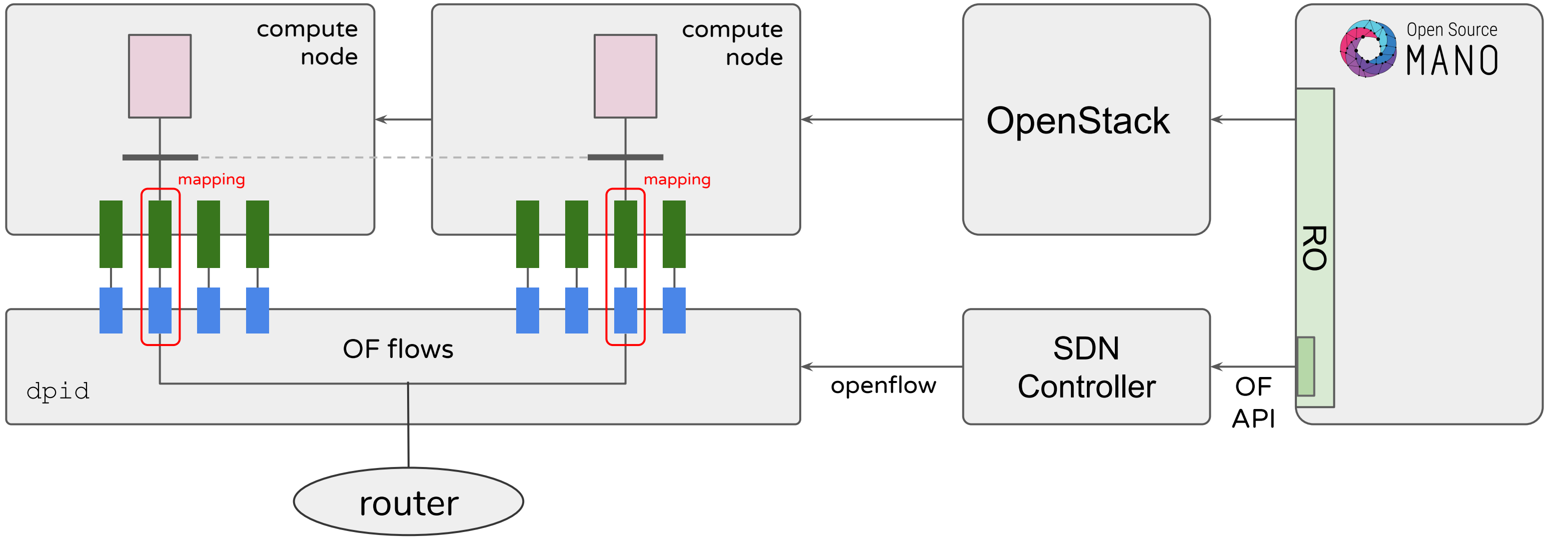
# SDN Assist

1. OSM orquestrates SR-IOV or Passthrough

   → Proper assignment of I/O physical interfaces to the VM (PFs or VFs = Physical or Virtual Functions)

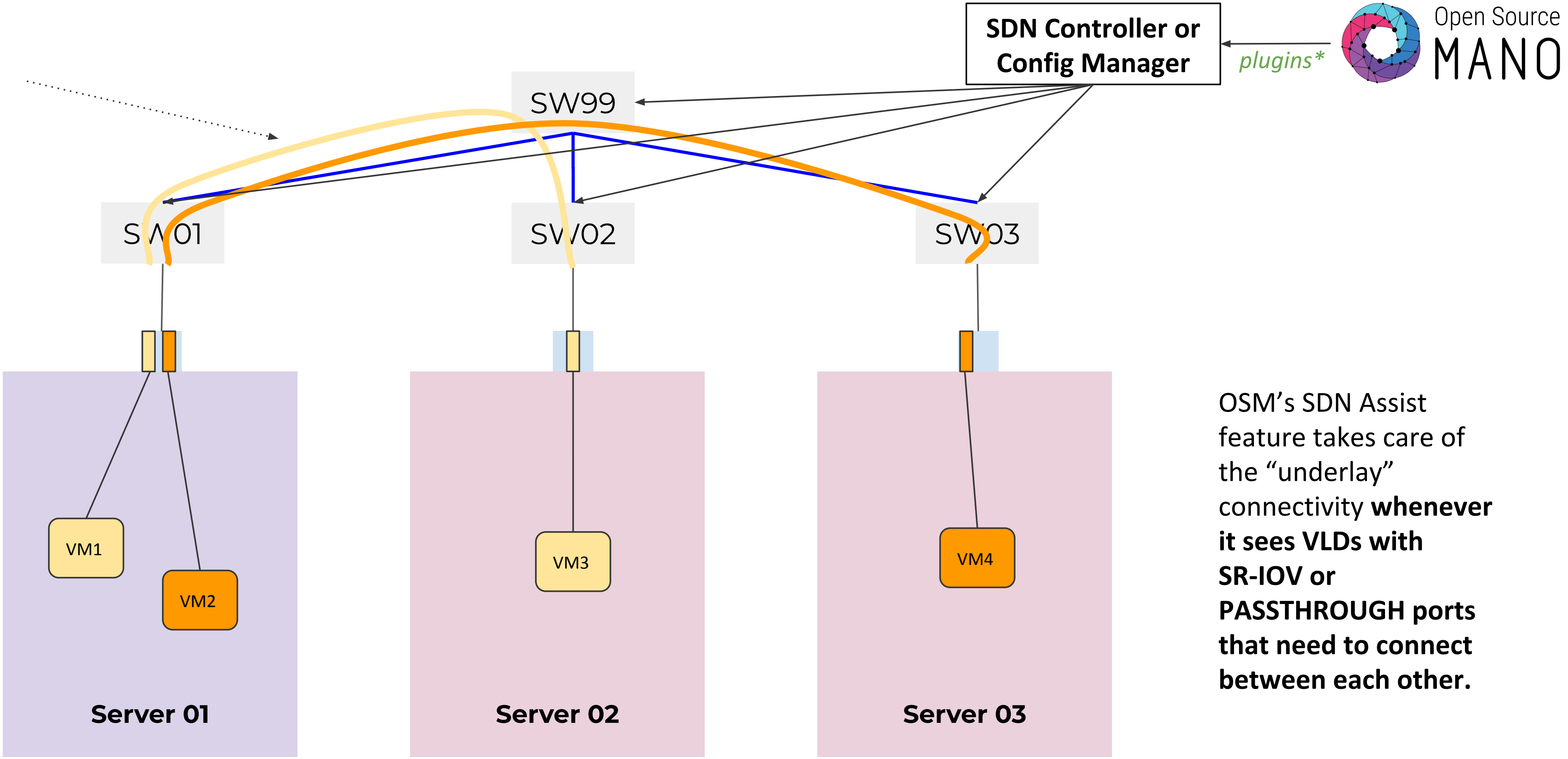
1. OSM SDN Assist gives the ability to create L2 connections between VFs

   - Interconnecting VMs
   - Attaching external traffic sources



# SDN Assist

VXLAN, OpenFlow, VLAN, etc. (depends on the SDN Controller or Manager)



OSM's SDN Assist feature takes care of the "underlay" connectivity **whenever it sees VLDs with SR-IOV or PASSTHROUGH ports that need to connect between each other.**

* Supported as of REL7.0.1 → ONOS, Arista, Contrail, Open Daylight and Floodlight

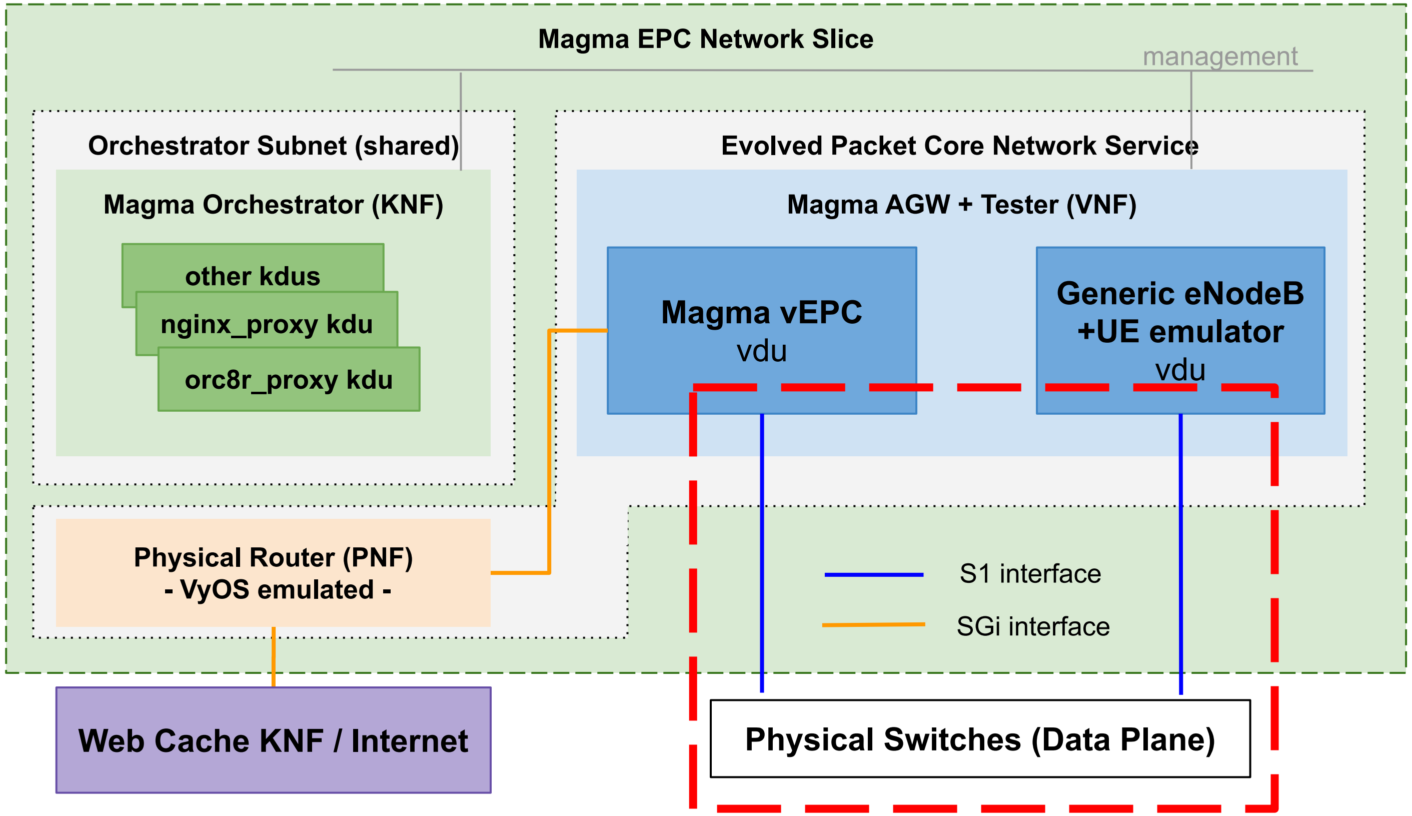
# SDN Assist requirements

- The VIM must have been created with a reference to the physnet(s) to use for SR-IOV (typically something like: --config '{dataplane_physical_net: <name>, microversion: 2.32}')
- An compatible SDN Controller of Config Manager must be installed and it must be reachable from OSM.
- For certain plugins (at least the OpenFlow based), an "SDN Port Mapping" file must be prepared to include all the possible PCI ports that can be selected by the VIM, per port.
- The VIM user must have admin privileges or rights to get the PCI information.
- Finally, the VIM and SDNC must be configured properly in OSM.
- More information here: https://osm.etsi.org/docs/user-guide/04-vim-setup.html#using-sdn-assist

# Back to our scenario!

In our example, we can configure the S1 data interface, currently using VIRTIO drivers (OVS/VxLAN) to use SR-IOV instead. We can also set the descriptor to request CPU Pinning, memory Huge Pages, and stick the VDUs to a single NUMA node.

Today in OSM, all these optimizations are applied automatically when selecting SR-IOV in one of the interfaces, in order to match the packet processing capabilities that the direct connection to the NIC will allow for.



# Preparing the environment

1. Create the SDN Controller, replacing XX with your tenant number-

```
osm sdnc-create --name onosXX --type onos_vpls --url http://172.21.248.57:8181 --user karaf --password karaf
```

2. Download the recommended SDN Port Mapping file

```
wget http://osm-download.etsi.org/ftp/osm-7.0-seven/OSM9-hackfest/files/magma_sdn_port_mapping.yaml
```

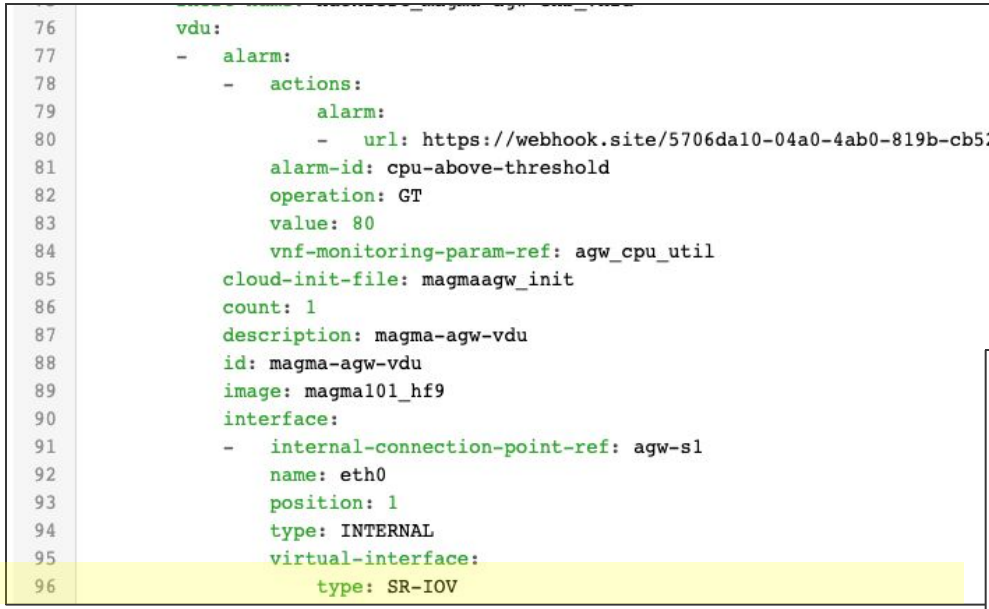
3. Update your VIM to know about the SDNC and the mapping file

```
osm vim-update etsi-openstack-XX --sdn_controller onosXX --sdn_port_mapping magma_sdn_port_mapping.yaml
```

# Modifying the S1s to use “**SR-IOV**”

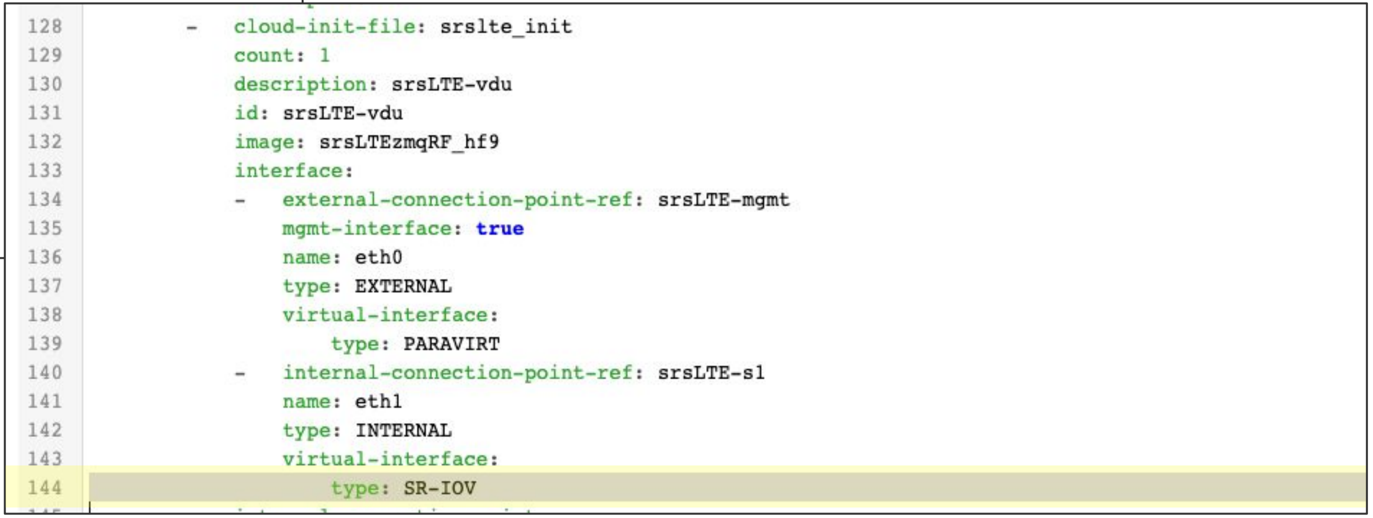
**hackfest_magma-agw-enb_vnfd**

**eth0 interface of MagmaAGW VDU (Line 96)**



```
76          vdu:
77          -   alarm:
78              -   actions:
79                      alarm:
80                      -   url: https://webhook.site/5706da10-04a0-4ab0-819b-cb524f71a367
81                  alarm-id: cpu-above-threshold
82                  operation: GT
83                  value: 80
84                  vnf-monitoring-param-ref: agw_cpu_util
85              cloud-init-file: magmaagw_init
86              count: 1
87              description: magma-agw-vdu
88              id: magma-agw-vdu
89              image: magma101_hf9
90              interface:
91              -   internal-connection-point-ref: agw-s1
92                  name: eth0
93                  position: 1
94                  type: INTERNAL
95                  virtual-interface:
96                      type: SR-IOV
```

**eth1 interface of srsLTE VDU (Line 144)**



```
128         -   cloud-init-file: srslte_init
129             count: 1
130             description: srsLTE-vdu
131             id: srsLTE-vdu
132             image: srsLTEzmqRF_hf9
133             interface:
134             -   external-connection-point-ref: srsLTE-mgmt
135                 mgmt-interface: true
136                 name: eth0
137                 type: EXTERNAL
138                 virtual-interface:
139                     type: PARAVIRT
140             -   internal-connection-point-ref: srsLTE-s1
141                 name: eth1
142                 type: INTERNAL
143                 virtual-interface:
144                     type: SR-IOV
```

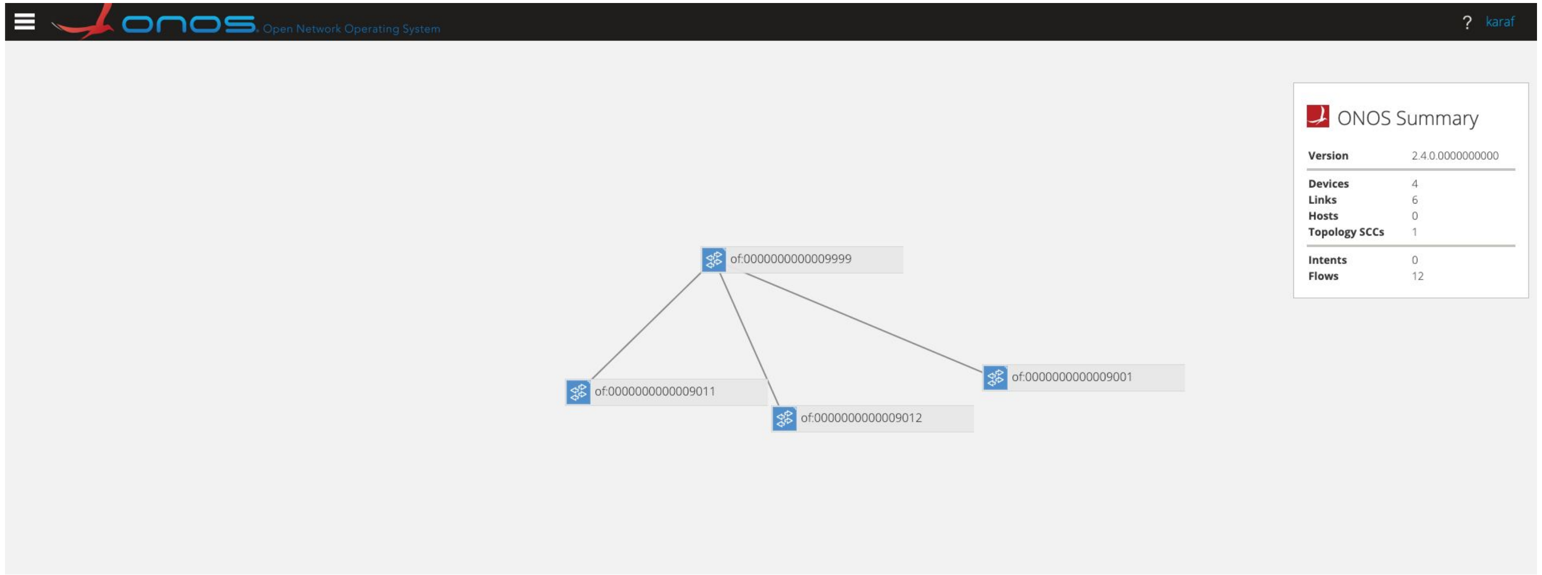
# Launch your NSI!

1. Use the scripts already in place

```
./launch_nsi.sh
```

2. Check the connectivity between VDUs and explore the network assigned in OpenStack

3. Optionally, visit the ONOS Controller UI at http://172.21.248.57:8181/onos/ui/login.html to check topology and flows (which you will see only if assigned different ports)



ONOS Summary

| | |
|---|---|
| Version | 2.4.0.0000000000 |
| Devices | 4 |
| Links | 6 |
| Hosts | 0 |
| Topology SCCs | 1 |
| Intents | 0 |
| Flows | 12 |

Open Source MANO

Find us at:

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