openmano

openvim API

North API documentation

document version: 0.5

27/02/2015

Technical description of the north interface of openvim VIM (Virtualised Infrastructure Manager. It contains json examples of commands/responses of the REST interface.

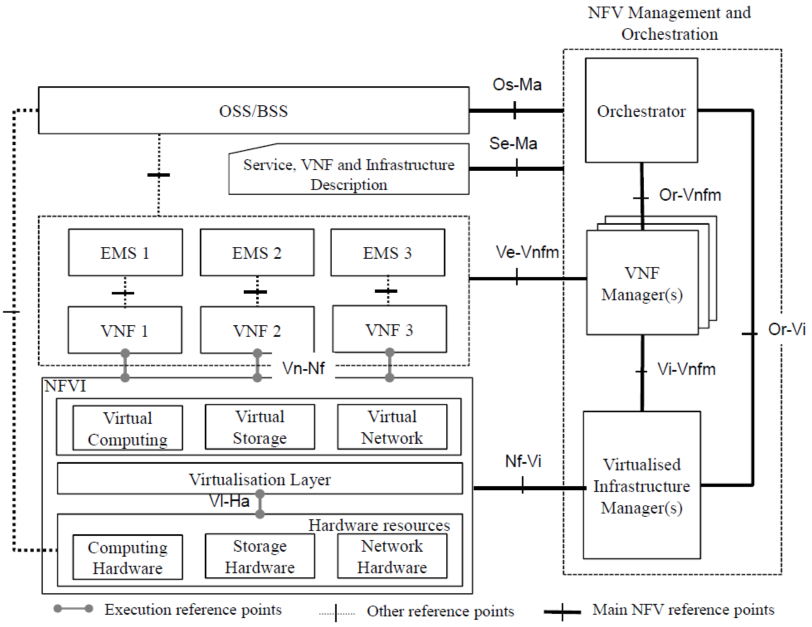
Content

CHANGE CONTROL

doc version date author (openvim version)	Comments
0.1 2014-07-11 Alfonso Tierno Pablo Montes	First version of document
0.2 2014-07-25 Alfonso Tierno	GENERAL: Spelling corrections DETAILS section: More general HTTP protocol and format guidelines GET /openvim/{tenant_id}/servers/{server_id} Section: Highlight all Network section to blue because not in openstack POST /openvim/{tenant_id}/servers/{server_id} Section: Highlight all Network section to blue because not in openstack New "Network Primitives" sections: The way of managing networks is different from previous version and closer to openstack. Now the API manages "networks" and "ports" with a similar meaning of openstack, and similar http primitives
0.3 2014-08-14 Alfonso Tierno	GENERAL: 'id' at the json examples are filled with a valid uuid format 3.1.3 POST /openvim/tenants: Some clarification 3.1.4 POST /openvim/tenants/{tenant_id}: Remove '*id*', that cannot be changed 3.2.3 POST /openvim/{tenant_id}/flavors: Inside 'extended': Removed 'bridge-ifaces' parameters from 'extended' 'numas','mermory' and 'cpus' are not optional, but mandatory 3.3.3 POST /openvim/{tenant_id}/images: 'name' is mandatory and removed a image link in the json example 3.4.2 GET /openvim/{tenant_id}/servers/{server_id}: Inside 'extended': New informative 'source' parameter at 'extended' Some changes in the interface information, as the 'iface_id', 'net_id' Some changes at 'network' entry 3.4.3 POST /openvim/{tenant_id}/servers: Some changes at 'network' entry changed to 'pauseD' one possible value of 'start' parameter 3.4.6 GET /openvim/{tenant_id}/servers/{server_id}/os-virtual-interfaces SECTION REMOVED, not needed, this information can be get otherwise 4.1.1 GET /openvim/networks: parameter 'type' changes and more clarification 4.1.3 POST /openvim/networks: changed an error in the json example 4.2.1 GET /openvim/ports: New parameters as 'device_owner', 'bandwidth' (instead of bandwidth_in and bandwidth_out), 'binding:vlan', 'binding:switch_port'
0.4 2014-11-06 Alfonso Tierno	3.2.2 GET /openvim/{tenant_id}/flavors/{flavor_id} Removed 'bridge-interfaces' from 'extended' Added a new field, 'imageRef' at 'devices' inside 'extended' 3.3 Images: Added metadata information 3.4 Servers: Added 'status' field when listing, new 'status' values and 'last_error' field 3.3.1 GET /openvim/{tenant_id}/images Remove 'link' fields and added the 'path' field 3.4.1 GET /openvim/{tenant_id}/servers Added the field 'hostId' in the response 3.4.4 POST /openvim/{tenant_id}/servers/{server_id}/action Added new actions, 'shutoff' and 'terminate' 4.1.1 GET /openvim/networks Misspelling with 'ptp' (point to point)fixed 5 ADMINISTRATIVE PRIMITIVES: New section
0.5 2015-02-26 Alfonso Tierno (0.1.90-r355)	GENERAL Change to the openvim naming 5.1.2 GET /openvim/hosts/{host_id} added the 'admin_state_up' and 'memory' to host 5.1.3 PUT /openvim/ports/{port_id} New section for editing a host

1 INTRODUCTION

This document describes the north interface of openvim VIM (Virtualised Infrastructure Manager).
According to the ETSI document "Network Functions Virtualisation (NFV); Architectural Framework"[1]; this document describes the "Or-Vi" interface.



In general, this interface is similar to the Openstack v2, so that this document focuses on the differences. These differences are motivated because while Openstack is focused in a cloud computing environment, VIM targets the deployment of NFV that allows:

- Getting physical resources without oversubscription
- Deterministic resources allocation per NUMA node.
- Create networks among physical and SR-IOV ports assigned to virtual machines in pass-through.

The new fields added to Openstack API are labelled in **blue**. Those items of Openstack ignored by current VIM are labelled in ~~crossed out red~~.

[1] http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

2 DETAILS

The Northbound interface is based on REST and it allows performing actions over the following entities:

- Tenant: Intended to create groups of virtual machines. In this VIM version no security mechanisms are implemented.
- Image: Software package containing the code of the virtual machine.
- Flavor: Description of resources needed to run an instance of a virtual machine.
- Server: Instance of a Virtual Machine (VM).
- Network: A net interconnects Virtual Machine interfaces among them. The IP configuration for each VM interface is not managed by the network and must be configured by the user.
- Ports: Each one of the VM interfaces.
- Hosts: Each compute node

HTTP protocol details

- The HTTP HEADER "X-Auth-Token" is ignored in this version, though it will be available in future. Current version does not support security and authentication
- Server supports JSON (by default), and YAML. Use HTTP HEADER "Content-Type: application/FORMAT" for specifying the input format and HTTP HEADER "Accept: application/FORMAT" for the wanted output format. In this version it does not support the URL suffix .yaml or .json as for example openstack neutron does.
- Server supports URL Query String filters. For example:
  HTTP GET /whatever?name1=value1&name2=value2"
  Will filter by "name1=value1 AND name2=value2"
- In a near future version it will support pagination using *limit*, *market*, *page_reverse* and *field* filtering in the same way as openstack neutron.
- Possible responses of HTTP Commands are:
  200 Ok
  400 Bad Request
  404 Not Found
  405 Method Not Allowed
  409 Conflict
  503 Service Unavailable
  500 Internal Server Error

3 SERVER PRIMITIVES

3.1 Tenants

3.1.1 GET /openvim/tenants

Get a list of all tenants

Params: none

Response Content-type: application/json

```
{
  "tenants": [
      {
           "id": "365d1f18-1959-11e4-9b59-52540030594e",
           "name": "ACME Corp",
           "description": "A description ...",
           "enabled": true
      },
      {
           "id": "8f0ad836-195d-11e4-836d-52540030594e",
           "name": "Iron Works",
           "description": "A description ...",
           "enabled": true
      }
  ]~~,~~
~~  "tenants_links": []~~
}
```

3.1.2 GET /openvim/tenants/{tenant_id}

Get the full description of the tenant identified by tenant_id

Params: none

Response Params:

- ***id***: uuid for the tenant
- ***name***: tenant name
- ***description***: (optional)tenant description
- ***enabled***: tenant enable or disable

Response Content-type: application/json

```
{
  "tenant": {
      "id": " 8f0ad836-195d-11e4-836d-52540030594e",
      "name": "ACME corp",
      "description": "A description ...",
      "enabled": true
  }
}
```

3.1.3 POST /openvim/tenants

Create new tenant.

Params: (Extra parameters are ignored)

- ***id***: (optional) proposed uuid for the tenant
- ***name***: tenant name provided by the client
- ***description***: (optional) tenant description provided by the client
- ***enabled***: (optional), enable or disable the tenant

Content-type: application/json

```
{
  "tenant": {
      "id": "8f0ad836-195d-11e4-836d-52540030594e",
      "name": "ACME corp",
      "description": "A description ...",
      "enabled": true
  }
}
```

Response: If not error same as GET /openvim/tenants/{tenant_id}

3.1.4 POST /openvim/tenants/{tenant_id}

Update tenant identified by tenant_id

Params: same as POST /openvim/tenants with the exception of "id" that cannot be changed. Extra parameters are not allowed

Content-type: application/json

```
{
  "tenant": {
      "name": "ACME corp",
      "description": "A description ...",
      "enabled": true
  }
}
```

Response: If not error same as GET /openvim/tenants/{tenant_id}

3.2 Flavors

3.2.1 GET /openvim/{tenant_id}/flavors

Get a list of all flavors defined for a tenant.

Response Content-type: application/json

```
{
  "flavors": [
      {
           "id": "1c609662-1bed-11e4-8d9f-52540030594e",
           "links": [
                {
                     "href": "http://openstack.example.com/v2/openstack/flavors/1c609662-
1bed-11e4-8d9f-52540030594e ",
                     "rel": "self"
                },
                {
                     "href": "http://openstack.example.com/openstack/flavors/1c609662-1bed-
11e4-8d9f-52540030594e ",
                     "rel": "bookmark"
                }
           ],
           "name": "m1.tiny"
      },
      {
           "id": "23afdb8e-22d3-11e4-94c0-52540030594e",
           "links": [
                {
                     "href": "http://openstack.example.com/v2/openstack/flavors/23afdb8e-
22d3-11e4-94c0-52540030594e ",
                     "rel": "self"
                },
                {
                     "href": "http://openstack.example.com/openstack/flavors/23afdb8e-22d3-
11e4-94c0-52540030594e ",
                     "rel": "bookmark"
                }
           ],
           "name": "m1.small"
      }
  ]
}
```

3.2.2 GET /openvim/{tenant_id}/flavors/{flavor_id}

Obtain a flavor full description

Params:

- ***id***: uuid or flavor
- ***ram***, ***vcpus***: (optional) used of not huge pages memory (MB), and not isolated cpus.
- ***extended***: (optional) NFV resources specification.
  - ***processor-ranking***: A ranking of minimum processor performance
  - ***devices***: (optional) List of extra devices as usb, disks, etc.
    - ***type***: type of device as "usb", "disk"
    - ***vpci***: (optional) desired virtual pci bus allocation
    - ***imageRef***: (mandatory if type is disk). Image uuid used for this disk
  - ***numas***: (optional) List of requirements set regarding memory, dataplane interfaces and cpus. Every set must be allocated in the same numa
    - ***memory***: (optional) needed memory in GB.
    - There are three possibilities for specifying cpus (optional):
      - ***cores***: Use physical cores (no hyperthreading).
      - ***paired-threads***: Use full physical cores with hyperthreading.
      - ***threads***: Use hyperthreading with no the restriction about the physical core.
    - For each of the above options you can optionally specify the vcpu identifier at the virtual machine. These identifier s must not overlap, must not contain gaps, and must start by zero:
      - ***cores →cores-id***: (Optional). Example: [0,1,2,3]
      - ***paired-threads →paired-threads-id***: (Optional). Example: [[0,1], [2,3], [4,5]]
      - ***threads →threads-id***: (Optional). Example: [0,1,2,3]
    - ***interfaces***: (Optional): list of data plane interfaces
      - ***name***: (optional) Interface proposed name.
      - ***dedicated***: yes/no. If dedicated full physical port is assigned in pass-through, otherwise SR-IOV port is assigned in pass-through.
      - ***bandwidth***: maximum needed bandwidth.
      - ***vpci***: (optional). PCI bus at virtual machine.

Response Content-type: application/json

```
{
  "flavor":
  {
     "disk": 1,
     "id": "23afdb8e-22d3-11e4-94c0-52540030594e",
     "links": [
        {
            "href":  "http://openstack.example.com/v2/openstack/flavors/23afdb8e-22d3-
11e4-94c0-52540030594e",
            "rel": "self"
        },
        {
            "href": "http://openstack.example.com/openstack/flavors/23afdb8e-22d3-11e4-
94c0-52540030594e1",
            "rel": "bookmark"
        }
    ],
    "name": "m1.tiny",
    "description": "description",
    "ram": 512,
    "vcpus": 1,
    "extended":
    {
        "processor-ranking": 205,
        "devices": [
            {
                "type": "usb",
                "vpci": "0000:00:1a.0"
            },
            {
                "type": "usb",
                "vpci": "0000:00:1b.0"
            }
        ],
        "numas": [
            {
                "memory": 16,
                "cores": 8,
                "cores-id": [0, 1, 2, 3, 4, 5, 6, 7],
                "interfaces": [
                    {
                        "name": "xe0",
                        "dedicated": "yes",
                        "bandwidth": "10 Gbps",
                        "vpci": "0000:00:05.0"
                    },
                    {
                        "name": "xe1",
                        "dedicated": "yes",
                        "bandwidth": "10 Gbps",
                        "vpci": "0000:00:06.0"
                    }
                ]
            },
            {
                "memory": 24,
                "paired-threads": 3,
                "paired-threads-id": [[10, 11], [12, 13], [14, 15]],
                "interfaces":  [
                    {
                        "name": "xe3",
                        "dedicated": "no",
                        "bandwidth": "5 Gbps",
                        "vpci": "0000:00:08.0"
                    }
                ]
            }
        ]
    }
  }
}
```

3.2.3 POST /openvim/{tenant_id}/flavors

Create a flavour

Params: (Extra parameters are ignored)

- ***id***:(optional): proposed uuid
- ***ram***, ***vcpus***: (optional) needed non-hupages memory (MB) and cpus. Ignored if memory and/or cores is supplied at *extended*.
- ***extended***: (optional) NFV resources specification.
  - ***processor-ranking***: (optional) A ranking of minimum processor performance
  - ***devices***: (optional) List of extra devices as usb, disks, etc.
    - ***type***: type of device as "usb", "disk"
    - ***vpci***: (optional) desired virtual pci bus allocation when instance is deployed.
    - ***imageRef***: (mandatory if type is disk). Image uuid used for this disk
  - ***numas***: List of requirements set regarding memory, dataplane interfaces and cpus. Every set must be allocated in the same numa
    - ***memory***:   needed memory in GB
    - There are three possibilities for specifying cpus:
      - ***cores***: Use physical cores (no hyperthreading).
      - ***paired-threads***: Use full physical cores with hyperthreading.
      - ***threads***: Use hyperthreading with no the restriction about the physical core.
    - For each of the above options you can optionally specify the vcpu identifier at the virtual machine. These identifier s must not overlap, must not contain gaps, and must start by zero:
      - ***cores*** → ***cores-id***: (Optional). Example: [0,1,2,3]
      - ***paired-threads*** → ***paired-threads-id***: (Optional). Example: [[0,1], [2,3], [4,5]]
      - ***threads*** → ***threads-id***: (Optional). Example: [0,1,2,3]
    - ***interfaces***: (Optional): list of data plane interfaces
      - ***name***: (optional) Interface proposed name.
      - ***dedicated***: yes/no. If dedicated full physical port is assigned in pass-through, otherwise SR-IOV port is assigned in pass-through.
      - ***bandwidth***: maximum needed bandwidth. No decimals allowed, units can be Gbps, Mbps(by default)
      - ***vpci***: (optional). PCI bus at virtual machine.

Content-type: application/json

```
{
  "flavor":
  {
     "disk": 1,
     "id": "23afdb8e-22d3-11e4-94c0-52540030594e ",
     "name": "m1.tiny",
     "description": "description",
     "ram": 512,
     "vcpus": 1,
     "extended":
     {
         "processor-ranking": 205,
        "devices": [
            {
                "type": "usb",
                "vpci": "0000:00:1a.0"
            },
            {
                "type": "usb",
                "vpci": "0000:00:1b.0"
            }
        ],
        "numas": [
            {
                "memory": 16,
                "cores": 8,
                "cores-id": [0, 1, 2, 3, 4, 5, 6, 7],
                "interfaces": [
                    {
                        "name": "xe0",
                        "dedicated": "yes",
                        "bandwidth": "10 Gbps",
                        "vpci": "0000:00:05.0"
                    },
                    {
                        "name": "xe1",
                        "dedicated": "yes",
                        "bandwidth": "10 Gbps",
                        "vpci": "0000:00:06.0"
                    }
                ]
            },
            {
                "memory": 24,
                "paired-threads": 3,
                "paired-threads-id": [[10, 11], [12, 13], [14, 15]],
                "interfaces": [
                    {
                        "name": "xe3",
                        "dedicated": "no",
                        "bandwidth": "5 Gbps",
                        "vpci": "0000:00:08.0"
                    }
                ]
            }
        ]
    }
  }
}
```

Response: If no error same as GET /openvim/{tenant_id}/flavors/{flavor_id}

3.2.4 DELETE /openvim/{tenant_id}/flavors/{flavor_id}

Remove a flavor

3.3 Images

3.3.1 GET /openvim/{tenant_id}/images

Get a list of all available images for a tenant

Response content-type: application/json

```
{
  "images": [
      {
          "id": "70a599e0-31e7-49b7-b260-868f441e862b",
~~          "links": [~~
~~               {~~
~~                    "href": "http://openstack.example.com/openstack/images/70a599e0-~~
~~31e7-49b7-b260-868f441e862b",~~
~~                    "rel": "bookmark"~~
~~               }~~
~~          ],~~
          "path": "/opt/image1.raw",
          "name": "fakeimage7"
      },
      {
          "id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
~~          "links": [~~
~~               {~~
~~                    "href": "http://openstack.example.com/openstack/images/155d900f-~~
~~4e14-4e4c-a73d-069cbf4541e6",~~
~~                    "rel": "bookmark"~~
~~               }~~
~~          ],~~
          "path": "/opt/image2.qcow2",
          "name": "fakeimage123456"
      }
  ]
}
```

3.3.2 GET /openvim/{tenant_id}/images/{image_id}

Get an image full description

Response Params:

- ***id***: uuid of the image
- ***path***: path where iso/qcow2 image is present.
- ***name***: image name
- ***description***:(optional): user description

Response content-type: application/json

```
{
  "image": {
      "created": "2011-01-01T01:02:03Z",
      "id": "70a599e0-31e7-49b7-b260-868f441e862b",
      "links": [
          {
               "href": "http://openstack.example.com/openstack/images/70a599e0-31e7-
49b7-b260-868f441e862b",
               "rel": "bookmark"
          }
      ],
      "metadata": {
          "architecture": "x86_64",
          "use_incremental": "no",
          "vpci": "0000:07:00.0",
          "os_distro": "ubuntu",
          "os_type": "linux",
          "os_version": "14.04",
      },
      ~~        "minDisk": 0,~~
      ~~        "minRam": 0,~~
      "name": "fakeimage7",
      "progress": 100,
      "status": "ACTIVE",
      "updated": "2011-01-01T01:02:03Z"
  }
}
```

3.3.3 POST /openvim/{tenant_id}/images

Create image

Params: (Extra parameters are ignored)

- ***id***:(optional): proposed uuid
- ***path***: path where iso/qcow2 image is present.
- ***name***:(Mandatory): user name
- ***description***:(optional): user description

Content-type: application/json

```
{
  "image": {
        "id": "70a599e0-31e7-49b7-b260-868f441e862b",
        "path": "/local/path/where/isoqcow2/is/present",
        "metadata": {
            "architecture": "x86_64",
            "use_incremental": "no",
            "vpci": "0000:07:00.0",
            "os_distro": "ubuntu",
            "os_type": "linux",
            "os_version": "14.04",
        },
~~        "minDisk": 0,~~
~~        "minRam": 0,~~
        "name": "fakeimage7",
        "description":  "user description"
  }
}
```

3.3.4 DELETE /openvim/{tenant_id}/images/{image_id}

Remove an image

3.4 Servers

3.4.1 GET /openvim/{tenant_id}/servers

List all instances of a tenant, name and links

Response content-type: application/json

```
{
    "servers": [
        {
            "hostId": "33c249e2-3e82-11e4-8b32-52540030594e",
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
            "links": [
                {
                    "href": "http://openstack.example.com/openstack/servers/616fb98f-
46ca-475e-917e-2563e5a8cd19",
                    "rel": "bookmark"
                }
            ],
            "name": "example",
            "status": "ACTIVE"
        }
    ]
}
```

3.4.2 GET /openvim/{tenant_id}/servers/{server_id}

Get a server details

Response params:

- ***status***: Can be
  - "***ACTIVE***" when server is running,
  - "***PAUSED***", or
  - "***INACTIVE***" when server has been created with ("start": "no") attribute
  - "***CREATING***", server is been created. "**progress**" shows a percentage
  - **"ERROR"**, some errer has happened on the machine creation. Field "**last_error**" describe what has happened
  - **"DELETING"**: machine is on process of removing
- **last_error**: (Optional). Contain a descriptive message of the last error happened.
- **progress**: (Optional). A percentage when machine is CREATING
- ***extended***: (Optional) Same than in flavors (see POST /openvim/flavors).
  It contain a "source" entry that details what element of host machine servers the virtual machine. For example pci NICs, core ids, numa socket.
  For every interface inside *numas* entry there are these new fields:
  - ***iface_id***: Each interface contain a proposed uuid
  - ***net_id***: (optional) This is the uuid of the network where this interface is attached. It is optional because in case it is not present or Null, the interface is not attached to any network. It can be attached using (PUT /mimtid/ports/iface_id).
  - ***source***: (optional) The source host pci. This is only informative for debugging purposes
- ***network***: Contain the list of bridge interfaces and the netwoks where this interfaces are connected.
  - ***net_id*** network identity where this interface is attached. The bridged interfaces are always attached to a network, so that this parameter is not optional
  - ***iface_id*** uuid of the bridge interface.
  - ***name***: (optional) Interface user proposed name.
  - ***bandwidth***: (optional) informative expected needed bandwidth (no decimals allowed, units can be Mbps by default or Gbps).
  - ***vpci***: (optional) virtual pci bus allocation when instance is deployed. Only appears if user forces it at creation
- ***hostID***: host uuid where server is hosted

Content-type: application/json.

```
{
  "server": {
      "description": "description",
      "hostId": "65201c14a29663e06d0748e561207d998b343e1d164bfa0aafa9c45d",
      ~~"accessIPv4": "",~~
      ~~"accessIPv6": "",~~
      ~~"addresses": {~~
          ~~"private": [~~
              ~~{~~
                  ~~"addr": "192.168.0.3",~~
                  ~~"version": 4~~
              ~~}~~
          ~~]~~
      ~~},~~
      "networks": [
          {
              "net_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
              "iface_id": "server interface uuid A"
              "name": "mngt0",
              "bandwidth": "100 Mbps",
              "vpci": "0000:00:10.0"
          }
      ],
      "created": "2012-08-20T21:11:09Z",
      "flavor": {
          "id": "1",
          "links": [
              {
                  "href": "http://openstack.example.com/openstack/flavors/1",
                  "rel": "bookmark"
              }
          ]
      },
      "hostId": "65201c14a29663e06d0748e561207d998b343e1d164bfa0aafa9c45d",
      "id": "893c7791-f1df-4c3d-8383-3caae9656c62",
      "image": {
          "id": "70a599e0-31e7-49b7-b260-868f441e862b",
          "links": [
              {
                  "href": "http://openstack.example.com/openstack/images/70a599e0-
31e7-49b7-b260-868f441e862b",
                  "rel": "bookmark"
              }
          ]
      },
      "links": [
          {
              "href": "http://openstack.example.com/openstack/servers/893c7791-f1df-
4c3d-8383-3caae9656c62",
              "rel": "bookmark"
          }
      ],
      ~~"metadata": {~~
          ~~"My Server Name": "Apache1"~~
      ~~},~~
      "name": "new-server-test",
      "progress": 0,
      "status": "ACTIVE",
      "tenant_id": "openstack",
      "updated": "2012-08-20T21:11:09Z",
      ~~"user_id": "fake",~~
      "extended":
```

```
      {
          "processor-ranking": 205,
          "devices":
          [
              {
                  "type": "usb",
                  "vpci": "0000:00:1a.0"
              },
              {
                  "type": "usb",
                  "vpci": "0000:00:1b.0"
              }
          ],
          "numas": [
          {
                  "memory": 16,
                  "cores": 8,
                  "cores-id": [0, 1, 2, 3, 4, 5, 6, 7],
                  "interfaces": [
                      {
                          "net_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
                          "name": "xe0",
                          "iface_id": "47eee0d6-2214-11e4-bfae-52540030594e"
                          "dedicated": "yes",
                          "bandwidth": "10 Gbps",
                          "vpci": "0000:00:05.0"
                          "source": "0000:10:00.0"
                      },
                      {
                          "net_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
                          "name": "xe1",
                          "iface_id": "4a70ea52-2214-11e4-bfae-52540030594e"
                          "dedicated": "yes",
                          "bandwidth": "10 Gbps",
                          "vpci": "0000:00:06.0"
                          "source": "0000:11:00.0"
                      }
              ]
              },
              {
                  "memory": 24,
                  "paired-threads": 3,
                  "paired-threads-id": [[10, 11], [12, 13], [14, 15]],
                  "interfaces":
                  [
                      {
                          "net_id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
                          "name": "xe3",
                          "iface_id": " 4ce3af4a-2214-11e4-bfae-52540030594e"
                          "dedicated": "no",
                          "bandwidth": "5 Gbps",
                          "vpci": "0000:00:08.0"
                          "source": "0000:12:00.0"
                      }
                  ]
              }
          ]
      }
  }
}
```

3.4.3 POST /openvim/{tenant_id}/servers

Create a new server or VM instance

Params: (Extra parameters are ignored)

- ***description***: (optional) user description
- ***start***: (defaults to "yes"). Possible values are:
  "yes": reserve resources and start the server.
  "no": just reserve resources.
  "paused": reserve resources and start the server in pause mode
- ***extended***: (Optional) Same than in flavors (see POST /openvim/flavors). When creating a server, in case this field is defined both in the flavor and here, the one defined here prevails.
  - ***uuid***: (optional) Each interface can contain a proposed uuid
- ***hostID***: (Optional) forces to deploy, if possible, in a concrete host
- ***network***: (Optional). For every entry a bridge interface is created and attached to this netwok. The network must exist and be of type bridge_man or bridge_data. (see POST vTID/netwoks). The content is:
  - ***uuid***: network of type bridge_XXX where to connect this interface
  - ***name***: (optional) Interface user proposed name.
  - ***bandwidth***: (optional) expected needed bandwidth, informative.
  - ***vpci***: (optional) desired virtual pci bus allocation when instance is deployed.

Content-type: application/json

```
{
  "server": {
       "name": "server-test-1",
       "description": "description",
       "start": "yes",
       "hostId": "65201c14a29663e06d0748e561207d998b343e1d164bfa0aafa9c45d",
       "imageRef": "b5660a6e-4b46-4be3-9707-6b47221b454f",
       "flavorRef": "2",
~~       "max_count": 1,~~
~~       "min_count": 1,~~
       "networks": [
            {
                 "uuid": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
                 "vpci": "0000:10:00.0"
                 "name": "mngt0"
                 "bandwidth": "1 Mbps"
            }
       ],
       ~~"security_groups": [~~
            ~~{~~
                 ~~"name": "default"~~
            ~~},~~
            ~~{~~
                 ~~"name": "another-secgroup-name"~~
            ~~}~~
       ~~],~~
       "extended": {
            "processor-ranking": 205,
            "devices": [
                 {
                      "type": "usb",
                      "vpci": "0000:00:1a.0"
                 },
                 {
                      "type": "usb",
                      "vpci": "0000:00:1b.0"
                 }
            ],
            "numas": [
                 {
                      "memory": 16,
                      "cores": 8,
                      "cores-id": [0, 1, 2, 3, 4, 5, 6, 7],
                      "interfaces": [
                          {
                               "name": "xe0",
                               "dedicated": "yes",
                               "bandwidth": "10 Gbps",
                               "vpci": "0000:00:05.0"
                          },
                          {
                               "name": "xe1",
                               "dedicated": "yes",
                               "bandwidth": "10 Gbps",
                               "vpci": "0000:00:06.0"
                          }
                      ]
                 },
                 {
                      "memory": 24,
                      "paired-threads": 3,
                      "paired-threads-id": [[10, 11], [12, 13], [14, 15]],
                      "interfaces":       [
                          {
                               "name": "xe3",
                               "dedicated": "no",
                               "bandwidth": "5 Gbps",
                               "vpci": "0000:00:08.0"
                          }
                      ]
                 }
            ]
       }
  }
}
```

3.4.4 POST /openvim/{tenant_id}/servers/{server_id}/action

Take an action over a server

Content-type: application/json:
No extra parameters are allowed. For reboot, the "type" is ignored
To reboot:

```
{
  "reboot": {
       "type": "SOFT"
  }
}
```

To start a server created with ("start":"no") attribute:

```
{
  "start": null
}
```

To pause a running server:

```
{
  "pause": null
}
```

To resume a paused server:

```
{
  "resume": null
}
```

To shutoff a server:

```
{
  "shutoff": null
}
```

To delete a server, this is the same as DELETE /openvim/{tenant_id}/servers/{server_id}:

```
{
  "terminate": null
}
```

3.4.5 DELETE /openvim/{tenant_id}/servers/{server_id}

Remove a server

4 NETWORK PRIMITIVES

Openvim considers a network as a layer 2 network. There are not subnetworks, nor IP management.
Thre are two type of networks:

- control plane: provided automatically by compute node bridges. It is assumed that the connection among compute node bridges are already set up by a switch or whatever.
- data plane networks: these networks are managed by openvim using an openflow controller. Normally it uses 10Gbps NIC ports connected to an openflow capability switch. SRIOV and passthrough ports are allocated to VMs

The concept of port is used for a VM interface or for an external connection on a dataplane network. The control plane VM interfaces must be attached to a network at the VM creation (same as openstack). However the data plane VM interface can be attached to a dataplane net both at VM creation or in a later step.

4.1 Nets

4.1.1 GET /openvim/networks

Get a list of all networks. It can be filtered by "tenant_id",

Params: none

Response Params:

- ***type***: Can be:
  - "**ptp**" (point-to-point) used by a layer 2 connection of two data plane interfaces
  - "**data**" used for connecting several (two or more) data plane interfaces among them

  - "**bridge_data**" used for connecting several bridged interfaces of instances among them. The bridged interfaces will use the physical resources configured for the "data" nets.
  - "**bridge_man**" (by default) used for connecting several bridged interfaces of instances among them. The bridged interfaces will use the physical resources configured for the "management" nets.
  - NOTE: the difference between bridge_data or bridge management is because they can be deployed over different physical interfaces and physical nets. openvim try to allocate the bridge_data on those e more bandwidth The allocated physical resources for data or management are a openvim installation configuration. The rationale is having separated resources for management and data communication. Anyway there is not any bandwidth guarantee in the bridge_data as far as several instances will share the same physical resources of the host
- The "***subnets***" list field is not present

Response content-type: application/json

```
{
    "networks": [
        {
            "status": "ACTIVE",
            ~~"subnets": [],~~
            "name": "provider-network",
            "provider:physical": null,
            "admin_state_up": true,
            "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
            ~~"provider:network_type": "local",~~
            "type": "data",
            ~~"router:external": true,~~
            "shared": true,
            "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
            ~~"provider:segmentation_id": null~~
        },
        {
            "status": "ACTIVE",
            ~~"subnets": [ ],~~
            "name": "private",
            "provider:physical": "bridge:virvbMan1,
            "admin_state_up": true,
            "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
            ~~"provider:network_type": "local",~~
            "type": "bridge_data",
            ~~"router:external": false,~~
            "shared": true,
            "id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
            ~~"provider:segmentation_id": null~~
        }
    ]
}
```

4.1.2 GET /openvim/networks/{network_id}

Get details of a network

Response Params:

- ***ports***: List of ports (interfaces), if any, that this net has already attached.

- ***type***: same as GET /openvim/networks

Response content-type: application/json

```
{
    "network": {
        "status": "ACTIVE",
        ~~"subnets": [],~~
        "name": "provider-network",
        "provider:physical": null,
        "admin_state_up": true,
        "tenant_id": "4fd44f30292945e481c7b8a0c8908869",
        ~~"provider:network_type": "local",~~
        "type": "data",
        "ports": [
            "port_id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
            "port_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
        ],
        "router:external": true,
        "shared": true,
        "id": "d32019d3-bc6e-4319-9c1d-6722fc136a22",
        ~~"provider:segmentation_id": null~~
    },
}
```

4.1.3 POST /openvim/networks

Create a new network

Params:

- ***name***: name provided by user
- ***id***: (optional). Proposed identifier of this network
- ***type***: (default to bridge_man). same as GET /openvim/networks
- ***shared***: (default to false if tenant_id is provided). Shared network
- ***tenant_id***: (optional) tenant that this network is created for
- ***admin_state_up***: (default to true) Administrative status

Content-type: application/json

```
{
    "network":{
         "name": "name provided by user",
         "id": "4e8e5957-649f-477b-9e5b-f1f75b21c045",
         "type": "point-to-point",
         "tenant_id": "9bacb3c5d39d41a79512987f338cf177 ",
         "shared": false,
         "admin_state_up": true
    }
}
```

Response: Same as GET /openvim/networks/{network_id}

4.1.4 PUT /openvim/networks/{network_id}

Modify a network. All the params of POST /openvim/networks that can be changed are:

- ***name***:
- ***type***: Only if network is free

- ***shared***: In case it is turn false, other tenants already using it are not modified, that is, the change applies to the new connections
- ***tenant_id***:
- ***admin_state_up***:

Params:
Same as POST /openvim/networks;

Response: Same as GET /openvim/networks/{network_id}

4.1.5 DELETE /openvim/networks/{network_id}

Delete a network. Must be empty, that is, without any port attached

4.2 Ports

4.2.1 GET /openvim/ports

Get the list of ports or instance interfaces. Can be filterded by tenant_id or device_id

Response params:

- ***name***: friendly user name
- ***id***: identification
- ***tenant_id***: (optional) tenant that uses this port
- ***network_id***: (optional) network where this port is attached. Can be NULL
- ***device_id***: server that this port belongs to.
- ***device_owner***: can be *instance:bridge* for the management or data bridge interfaces, *instance:data* for the extended data interfaces or *external* for an external connection outside of the VIM controller.
- ***bandwidth***: (Optional) Expected bandwidth in this port. Only informative
- ***binding***: (Optional) details of virtual interfaces binding
  - ***binding:switch_port***: (Optional, Mandatory for external device_owner) Informative physical connection
  - ***binding:vlan***: (Optional,) Informative physical connection

Response content-type: application/json

```
{
    "ports": [
        {
            "status": "ACTIVE",
            ~~"binding:host_id": "devstack",~~
            "name": "",
            ~~"allowed_address_pairs": [],~~
            "admin_state_up": true,
            "network_id": "70c1db1f-b701-45bd-96e0-a313ee3430b3",
            "tenant_id": "",
            ~~"extra_dhcp_opts": [],~~
            ~~"binding:vif_details": {~~
                ~~"port_filter": true,~~
                ~~"ovs_hybrid_plug": true~~
            ~~},~~
            ~~"binding:vif_type": "ovs",~~
            "device_owner": "network:router_gateway",
            "mac_address": "fa:16:3e:58:42:ed",
            "binding:vlan": 345
            "binding:switch_port: "Te0/10"
            "bandwidth: 10000 Mbps"
            ~~"binding:profile": {},~~
            ~~"binding:vnic_type": "normal",~~
            ~~"fixed_ips": [~~
                ~~{~~
                    ~~"subnet_id": "008ba151-0b8c-4a67-98b5-0d2b87666062",~~
                    ~~"ip_address": "172.24.4.2"~~
                ~~}~~
            ~~],~~
            "id": "d80b1a3b-4fc1-49f3-952e-1e2ab7081d8b",
            ~~"security_groups": [],~~
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
        },
        {
            "status": "ACTIVE",
            ~~"binding:host_id": "devstack",~~
            "name": "",
            ~~"allowed_address_pairs": [],~~
            "admin_state_up": true,
            "network_id": "f27aa545-cbdd-4907-b0c6-c9e8b039dcc2",
            "tenant_id": "d397de8a63f341818f198abb0966f6f3",
            ~~"extra_dhcp_opts": [],~~
            ~~"binding:vif_details": {~~
                ~~"port_filter": true,~~
                ~~"ovs_hybrid_plug": true~~
            ~~},~~
            ~~"binding:vif_type": "ovs",~~
            ~~"device_owner": "network:router_interface",~~
            "mac_address": "fa:16:3e:bb:3c:e4",
            ~~"binding:profile": {},~~
            ~~"binding:vnic_type": "normal",~~
            ~~"fixed_ips": [~~
                ~~{~~
                    ~~"subnet_id": "288bf4a1-51ba-43b6-9d0a-520e9005db17",~~
                    ~~"ip_address": "10.0.0.1"~~
                ~~}~~
            ~~],~~
            "id": "f71a6703-d6de-4be1-a91a-a570ede1d159",
            ~~"security_groups": [],~~
            "device_id": "9ae135f4-b6e0-4dad-9e91-3c223e385824"
        }
    ]
}
```

4.2.2 PUT /openvim/ports/{port_id}

Update parameters of a Port. It is used also for attaching/detaching this port to a concrete network

Request params:

- ***name***: (Optional) for changing the name
- ***network_id***: (Optional) Can be:
    - ***id*** of network where this port want to be attached. Port is de-attached from its previous network if any.

- ***Null:*** Port is de-attaching from its previous network and left unattached

Response: Same as GET /openvim/porss/{port_id}

4.2.3 Creation and deletion of ports

~~POST /openvim/ports/{port_id} DELETE /openvim/ports/{port_id}~~

Ports are created automatically when instances are launched for every interface specify at the "extended" field. Also, they are deleted when instances are deleted.

5 ADMINISTRATIVE PRIMITIVES

This section describes other primitives related to administrative users. The "GET" primitives are available for all users, but "POST/DELETE/PUT" primitives are only available for administrative users.
In this version administrative users are distinguish using a different HTTP port, called "admin_port"

5.1 *HOSTs (/openvim/hosts)*

5.1.1 GET /openvim/hosts

Get a list of all available hosts

Response content-type: application/json

```
{
    "hosts": [
        {
            "status": "ok",
            "description": null,
            "id": "aecaf9d8-22bf-11e4-9204-52540030594e",
            ~~"links": [~~
~~                {~~
~~                    "href": "http://localhost:8081/openvim/hosts/aecaf9d8-22bf-11e4-~~
~~9204-52540030594e",~~
~~                    "rel": "bookmark"~~
~~                }~~
~~            ],~~
            "name": "nfv101"
        }
    ]
}
```

5.1.2 GET /openvim/hosts/{host_id}

Get a host full description

Response Params:

- ***id***: uuid of the host
- ***ranking***: processor ranking
- ***ip_name***: ip or name to access the host
- ***name***: host name
- ***description***:(optional): host description
- ***admin_status_up***: host available or not
- ***numas***: description of the hardware,
    - ***admin_status_up***: numa available or not
    - ***memory***: non hugepages memory
    - ***hugepages***, ***cores*** and ***interface***. Each one can contain:
        - ***instance_id***:(Optional) this indicates that this resource is occupied by this virtual machine

- ***status***: (Optional) if present indicates and error status of the resource, or "not_eligible" means that not available for virtualization

Response content-type: application/json

```
{
    "host": {
        "ranking": 170,
        "description": "nfv130",
        "created_at": "2014-11-06T11:43:02",
        "ip_name": "10.95.87.139",
        "name": "nfv130",
        "admin_state_up": true,
        "numas": [
            {
                "admin_state_up": true,
                "interfaces": [
                    {
                        "Mbps": 10000,
                        "pci": "0000:04:00.1",
                        "Mbps_consumed": 0
                    },
                    {
                        "Mbps": 10000,
                        "pci": "0000:03:00.1",
                        "Mbps_consumed": 0
                    },
                    {
                        "Mbps": 10000,
                        "pci": "0000:03:00.0",
                        "Mbps_consumed": 0
                    }
                ],
                "numa_socket": 0,
                "hugepages_consumed": 0,
                "hugepages": 28,
                "memory": 32,
                "cores": [
                    {
                        "core_id": 0,
                        "status": "noteligible",
                        "thread_id": 0
                    },
                    {
                        "core_id": 0,
                        "status": "noteligible",
                        "thread_id": 16
                    },
                    {
                        "core_id": 1,
                        "thread_id": 2
                    },
                    {
                        "core_id": 1,
                        "thread_id": 18
                    },
                    {
                        "core_id": 2,
                        "thread_id": 4
                    },
                    {
                        "core_id": 2,
                        "thread_id": 20
                    },
                    {
                        "core_id": 3,
                        "thread_id": 6
                    },
                    {
                        "core_id": 3,
                        "thread_id": 22
                    },
                    {
                        "core_id": 4,
                        "thread_id": 8
                    },
                    {
                        "core_id": 4,
                        "thread_id": 24
                    },
                    {
                        "core_id": 5,
                        "thread_id": 10
                    },
                    {
                        "core_id": 5,
                        "thread_id": 26
                    },
                    {
                        "core_id": 6,
                        "thread_id": 12
                    },
                    {
                        "core_id": 6,
                        "thread_id": 28
                    },
                    {
                        "core_id": 7,
                        "thread_id": 14
                    },
                    {
                        "core_id": 7,
                        "thread_id": 30
                    }
                ]
            }
        ],
        "id": "aea5a442-65a1-11e4-9d81-52540047525e"
    }
}
```

5.1.3 POST /openvim/hosts

Create a host.

- It can be created automatically, providing only access credential to host (ip or name, user) so that server connect to host an retrieve all the necessary information, or
- It can be created manually providing all the needed information

<u>Automatic creation</u>

Params:

host (Extra parameters are ignored)

- ***id***:(optional): proposed uuid
- ***description***: (optional): description of the host
- ***name***: user name
- ***ip_name***: ip or name to access to the host
- ***user***: user of ssh connection
- ***description***:(optional): user description

Content-type: application/json

```
{
  "host":
  {
       "name":"nfv101",
       "user":"user",
       "ip_name":"10.202.0.101",
       "description":"rel7"
  }
}
```

Manual creation

Params: In addition to the ***host*** entry, a ***host-data*** entry contains the needed information to charge into database instead of getting from the host

host-data: (Extra parameters are not allowed)

- ***ranking***: ranking number for this processor
- ***numas***: numas description list:
  - ***numa_socket***: physical numa socket, starting from 0
  - ***hugepages***: number of 1G huge pages reserved.
  - ***cores***: core list, containing core_id and thread_id
  - ***interfaces***: interface list, with ***mac***, ***Mbps***(speed), ***source_name*** (linux iface name), ***pci***, and possible sriov list
    - ***sriovs***: list of sriovs, with ***mac***, ***source_name*** (sriov number starting from 0), ***pci, vlan***

Content-type: application/json

```
{
"host":{
    "ip_name": "10.202.0.101",
    "user": "user",
    "name": "nfv101",
    "passwd": "password"
},
"host-data":
{
    "ranking": 300,
    "numas": [
        {
            "cores": [
                {
                    "core_id": 0,
                    "thread_id": 0
                },
                {
                    "core_id": 0,
                    "thread_id": 24
                },
                {
                    "core_id": 1,
                    "thread_id": 2
                },
                {
                    "core_id": 1,
                    "thread_id": 26
                },
                {
                    "core_id": 2,
                    "thread_id": 4
                },
                {
                    "core_id": 2,
                    "thread_id": 28
                },
                {
                    "core_id": 3,
                    "thread_id": 6
                },
                {
                    "core_id": 3,
                    "thread_id": 30
                },
                {
                    "core_id": 4,
                    "thread_id": 8
                },
                {
                    "core_id": 4,
                    "thread_id": 32
                },
                {
                    "core_id": 5,
                    "thread_id": 10
                },
                {
                    "core_id": 5,
                    "thread_id": 34
                },
                {
                    "core_id": 6,
                    "thread_id": 12
                },
                {
                    "core_id": 6,
                    "thread_id": 36
                },
                {
                    "core_id": 7,
                    "thread_id": 14
                },
                {
                    "core_id": 7,
                    "thread_id": 38
                },
                {
                    "core_id": 8,
                    "thread_id": 16
                },
                {
                    "core_id": 8,
                    "thread_id": 40
                },
                {
                    "core_id": 9,
                    "thread_id": 18
                },
                {
                    "core_id": 9,
                    "thread_id": 42
                },
                {
                    "core_id": 10,
                    "thread_id": 20
                },
                {
                    "core_id": 10,
                    "thread_id": 44
                },
                {
                    "core_id": 11,
                    "thread_id": 22
                },
                {
                    "core_id": 11,
                    "thread_id": 46
                }
            ],
            "interfaces": [
                {
                    "source_name": "p5p2",
                    "Mbps": 10000,
                    "pci": "0000:04:00.1",
                    "sriovs": [
                        {
                            "mac": "02:80:30:4b:ba:6b",
                            "pci": "0000:04:10.1",
                            "source_name": 0,
                            "vlan": 104
                        },
                        {
                            "mac": "c2:8d:99:69:11:ca",
                            "pci": "0000:04:10.3",
                            "source_name": 1,
                            "vlan": 105
                        },
                        {
                            "mac": "be:d5:ad:6e:cd:bc",
                            "pci": "0000:04:10.5",
                            "source_name": 2,
                            "vlan": 106
                        },
                        {
                            "mac": "5a:b7:f6:11:2a:27",
                            "pci": "0000:04:10.7",
                            "source_name": 3,
                            "vlan": 107
                        },
                        {
                            "mac": "02:37:78:8a:ce:a5",
                            "pci": "0000:04:11.1",
            "source_name": 4,
            "vlan": 102
        },
        {
            "mac": "72:c7:6b:5a:89:5a",
            "pci": "0000:04:11.3",
            "source_name": 5,
            "vlan": 103
        },
        {
            "mac": "7e:ec:45:c8:89:8e",
            "pci": "0000:04:11.5",
            "source_name": 6,
            "vlan": 100
        },
        {
            "mac": "06:23:e5:34:a6:02",
            "pci": "0000:04:11.7",
            "source_name": 7,
            "vlan": 101
        }
    ],
    "mac": "a0:36:9f:33:09:6e"
},
{
    "source_name": "p7p1",
    "Mbps": 10000,
    "pci": "0000:06:00.0",
    "sriovs": [
        {
            "mac": "56:a9:21:e3:f8:11",
            "pci": "0000:06:10.0",
            "source_name": 0,
            "vlan": 105
        },
        {
            "mac": "5e:6c:cd:b7:76:b2",
            "pci": "0000:06:10.2",
            "source_name": 1,
            "vlan": 104
        },
        {
            "mac": "36:f4:64:13:c5:e4",
            "pci": "0000:06:10.4",
            "source_name": 2,
            "vlan": 107
        },
        {
            "mac": "aa:c9:40:52:6f:d2",
            "pci": "0000:06:10.6",
            "source_name": 3,
            "vlan": 106
        },
        {
            "mac": "3a:52:27:ce:d1:84",
            "pci": "0000:06:11.0",
            "source_name": 4,
            "vlan": 103
        },
        {
            "mac": "8e:93:78:04:11:74",
            "pci": "0000:06:11.2",
                            "source_name": 5,
                            "vlan": 102
                        },
                        {
                            "mac": "62:5b:4c:4b:db:66",
                            "pci": "0000:06:11.4",
                            "source_name": 6,
                            "vlan": 101
                        },
                        {
                            "mac": "56:c4:28:b5:6a:21",
                            "pci": "0000:06:11.6",
                            "source_name": 7,
                            "vlan": 100
                        }
                    ],
                    "mac": "a0:36:9f:33:0c:d4"
                }
            ],
            "numa_socket": 0,
            "hugepages": 28
        }
    ]
}
}
```

Response:
Same as GET /openvim/hosts/{host_id}

5.1.4 DELETE /openvim/hosts/{host_id}

Remove a host. Must not contain any instance running on this

5.1.5 PUT /openvim/hosts/{host_id}

Edit parameters of a host, as the admin_status_up, name, description, …
Content-type: application/json

```
{
  "host":
  {
      "name":"nfv101",
      "user":"user",
      "ip_name":"10.202.0.101",
      "description":"rel7"
      "admin_state_up": true,
      "numas": [
            {
                "admin_state_up": true,
                "numa_socket": 0
            }
  }
}
```

Response:
Same as GET /openvim/hosts/{host_id}

5.2 Openflow rules (/openvim/networks/{nerwork_id}/openflow)

For debugging purposes it can be reinstalled the open flow rules of a concrete network

It is done with a PUT command over the URL (/openvim/networks/{nerwork_id}/openflow). Input data is ignored. network_id can be "all" for applying to all the networks

5.3 External Ports (/openvim/ports)

These are the interfaces to other external networks. The content is similar to "ports". Every provider network will need to have at least one external port attached to it.
Administrative users can create and delete external ports, and modify more parameters of ports

5.3.1 POST /openvim/ports

Create a new external port.
Params:

- ***name***: name provided by user
- ***id***: (optional). Proposed identifier of this port
- ***network_id***: (optional). If provided the port is attached to this network
- ***tenant_id***: (optional) tenant that this network is created for
- ***mac_address***: (optional): mac address that the external devices will use
- ***admin_state_up***: (default to true) Administrative status
- ***bandwidth***: (optional) Informative bandwidth in Mbps
- ***binding:switch_port***: (Mandatory) Physical port of the switch where this port is allocated
- ***binding:vlan***: (Optional) VLAN used for transmitting/receiving packets from this port. If not provided vlan is not used

Not allowed params:

- ***device_owner***: taked as external
- ***device_id***: taked as Null

Content-type: application/json

```
{
    "port":{
        "admin_state_up": true,
        "binding:switch_port": "Te0/8",
        "tenant_id": "e505d736-195d-11e4-836d-52540030594e",
        "bandwidth": 10000,
        "name": "external 1"
    }
}
```

Response: Same as GET /openvim/ports/{port_id}

5.3.2 PUT /openvim/ports/{port_id}

Modify a port. The params of POST /openvim/ports that can be changed are:

- ***name***: Same as non-administrative.
- ***net_id***: Same as non-administrative.

Only for external ports, this parameters can be changed::

- binding:vlan
- binding:switch_port

- mac_address

Content-type: application/json

```
{
     "port":{
          "binding:switch_port": "Te0/8",
          "tenant_id": "e505d736-195d-11e4-836d-52540030594e",
     }
}
```

Response: Same as GET /openvim/ports/{port_id}

5.3.3 DELETE /openvim/ports/{port_id}

Delete a port. Must be an external port