# Re-virtualization for Next Generation Global Connectivity Services

## Distributed Edge/NFV Use Case

Version 1.0 – November 2024


### Abstract

OpenNebula can run Virtual Network Functions (VNFs) and enable Container Network Functions (CNFs) as a single Front-end cloud, managing any number from tens to hundreds of geo-distributed clusters on minimal hardware infrastructure, using local storage and Data Plane Development Kit (DPDK) or Single Root I/O Virtualization (SR-IOV) technologies to achieve high-performance throughput. Additionally, OpenNebula's implementation of Enhanced Platform Awareness (EPA) helps to improve VM packet forwarding performance (throughput, latency, jitter) by exposing low-level CPU and NIC acceleration components to the VNF. OpenNebula also provides the GPU support, bare-metal automation and multi-cluster features needed to address distributed telco-cloud use cases.

OpenNebula is enhancing its Artificial Intelligence techniques and Zero-Touch resource management methods for the efficient deployment and operation of distributed cloud-edge-continuums. OpenNebula is helping its users to combine centralized clouds with edge resources and to choose the right combination of geographically distributed cloud-edge locations to efficiently execute their workloads, meet their enterprise needs, and avoid vendor lock-in.

This white paper presents a detailed test plan validated with multiple users, including Tier-1 telco carriers. It provides a release report highlighting the particular features and functionalities that position OpenNebula as the answer for highly distributed NFV deployments, (re-)virtualization of distributed cloud-edge infrastructure, and overall progress towards a cloud technology-agnostic mobile network with next generation features for automated and AI-driven deployment and operations.

## Contents

# Glossary

| | |
|---|---|
| **ACL** | Access Control List |
| **AD** | Active Directory |
| **API** | Application Programming Interface |
| **AWS** | Amazon Web Services |
| **BSS** | Business Support System |
| **CAGR** | Compound Annual Growth Rate |
| **CLI** | Command-Line Interface |
| **CNF** | Container Network Functions |
| **CPU** | Central Processing Unit |
| **DPDK** | Data Plane Development Kit |
| **EMS** | Element Management System |
| **EPA** | Enhanced Platform Awareness |
| **ETSI** | European Telecommunications Standards Institute |
| **EU** | European Union |
| **GPU** | Graphics Processing Unit |
| **GUI** | Graphical User Interface |
| **HA** | High Availability |
| **HW** | Hardware |
| **ID** | Identifier |
| **IO** | Input/Output |
| **IP** | Internet Protocol |
| **ISG** | Industry Specification Group |
| **IT** | Information Technology |
| **KVM** | Kernel-based Virtual Machine |
| **LDAP** | Lightweight Directory Access Protocol |
| **LFN** | Long Fat Networks |
| **MANO** | Management and Network Orchestration |
| **MPLS** | Multi-Protocol Label Switching |
| **MVNO** | Mobile Virtual Network Operator |
| **NAS** | Network Area Storage |
| **NFH** | Network Function Hub |
| **NFS** | Network File System |
| **NFV** | Network Function Virtualization |
| **NFVI** | Network Function Virtualization Infrastructure |
| **NFVO** | Network Function Virtualization Orchestrator |
| **NIC** | Network Interface Card |
| **NMS** | Network Management System |
| **NUMA** | Non-Uniform Memory Access |
| **OS** | Operating System |
| **OSS** | Operations Support System |
| **OTT** | Over-The-Top media service |
| **OVS** | Open V-Switch |
| **PCI** | Peripheral Component Interconnect |
| **PoC** | Proof of Concept |
| **PT** | PassThrough |
| **RAN** | Radio Access Network |
| **REST** | REpresentational State Transfer |
| **RTT** | Round-Trip Time |
| **SDN** | Software-Defined Networking |
| **SR-IOV** | Single Root IO Virtualization |
| **SSH** | Secure SHell |
| **SW** | SoftWare |
| **TAM** | Technical Account Management |
| **TCO** | Total Cost of Ownership |
| **UUID** | Universally Unique Identifier |
| **VDC** | Virtual Data Center |
| **VF** | Virtual Function |
| **VIM** | Virtual Infrastructure Manager |
| **VLAN** | Virtual Local Area Network |
| **VM** | Virtual Machine |
| **VNF** | Virtual Network Function |
| **VNFM** | Virtual Network Functions Manager |
| **VNO** | Virtual Network Operator |
| **WAN** | Wide Area Network |
| **WIM** | WAN Infrastructure Manager |

# PART A. Introduction

## 1. Virtualizing Infrastructure for the Most Complex Telecom Use Cases

Mobile Networks have benefited from virtualization in the past years, evolving from a group of vertically-integrated service-specific network functions that rely on dedicated physical resources, to fully virtualized environments in which network and service functions are performed by software decoupled from the underlying hardware. This evolution was facilitated by the application, in the domain of mobile networks, of virtualization methods that were already in use in the IT world, such as SDN (Software- Defined Networking) and NFV (Network Function Virtualization).

Network Function Virtualization offers the possibility of decoupling network functionality from proprietary hardware, and deploying it on any type of hardware in a cloud-based architecture. ETSI ISG NFV has defined the NFV-MANO framework, founded on three main building blocks: NFVI, VNFs and MANO.[1]

The Network Functions Virtualization Infrastructure (NFVI) represents the foundational layer of the NFV-MANO framework. It is composed of compute, storage, management and network resources, usually as part of one service provider. By using hypervisors (such as VMware ESXi or KVM) it transforms the physical resources into the virtual compute, storage, and network infrastructure used for deploying, scaling and running the VNFs. NFVI resources can be geographically-distributed across multiple sites and zones, ensuring high availability and fulfilling the requirements and workloads of a wide range of use cases.

Virtual Infrastructures are key underlying elements for any virtualized network. They allow software applications to run on VMs or containers deployed on general purpose hardware elements. This opens the door to gaining new advantages such as scalability, efficient resource usage, sustainability, competitiveness, innovation, programmability and reduction of the Total Cost of Ownership (TCO) for telecom operators and/or service providers.

The Management and Orchestration (MANO) Layer is responsible for orchestrating and managing the hardware and software resources. MANO is divided into three main components:

- Virtual Infrastructure Manager (VIM).
- Virtual Network Functions Manager (VNFM).
- NFV Orchestrator (NFVO).

The VNFM has become an important part of any telecom network, deployed as a central element for controlling, monitoring and managing the operations and life cycle of the VNF. It also handles the Element Management System (EMS) and/or the Network Management System (NMS). The NFVO is responsible for deploying Network Services and for onboarding the VNFs, managed by one or multiple VNFMs and on the same or multiple NFVIs.

The VIM is responsible for controlling and managing the infrastructure underlying the Virtualized Network Functions (VNFs). Virtual resources need to make an efficient use of physical assets, and here the VIM intervenes providing the below functionalities:

- Software image management.
- Virtualized resources allocation and management in accordance with traffic engineering rules.
- Infrastructure resource fault and performance management.
- NFV acceleration capabilities management.
- Orchestration of usage and provisioning of the virtual infrastructure.

These characteristics are exposed through Northbound APIs/interfaces to VNFM and NFVO, but also via Southbound Interfaces to the NFVI's elements.

[1] https://www.etsi.org/deliver/etsi_gs/NFV/001_099/006/04.04.01_60/gs_NFV006v040401p.pdf ETSI ISG NFV 006 4.4.1, retrieved in October 2024.

VIM ensures that the consumers or end-users get the level of service they expect and the amount of resources needed to meet their requirements at any given moment and for any given use case, without impacting other users' operations.

VMs, NFVI and, more recently, containers transform everything into variables. The amount of bandwidth, memory, CPUs, storage or compute power can be updated simply by modifying configuration parameters. Even network functions, previously provided by pure-hardware devices (e.g. switches, load balancers or routers) can be deployed as pure-software applications, created and destroyed based on the requirements and network evolution of particular use cases.

Virtual Infrastructures reduce the cost of storage by allowing owners to efficiently use physical resources. Combined with the advantage of scalability, they enable dynamic allocation of resources according to the needs of the organization and/or specific use cases. Thanks to their portability, VMs and containers can benefit from the many possible configurations for servers and network devices, which is a great advantage compared to traditional hardware. Another advantage is enhanced security, since an additional security layer is implemented at the VM/container level, on top of the secure infrastructure.

The security barrier resulting from the separation between VMs keeps the system secure from bugs and viruses. Adopting virtual infrastructures improves load balancing, since multiple servers can handle multiple workloads in parallel and specific VMs can be distributed on-demand to offload a very busy system. In terms of backup and recovery, implementing virtualized Infrastructure reduces the number of single points of failure. Leveraging this concept, we can configure a backup/copy of VMs and containers to be easily restored or activated—with minimum to no service impact—when a fault occurs at the level of physical infrastructure.

The VIM Market is influenced by the adoption of cloud-native technologies, 5G and Edge Computing, NFV, and Multi-Cloud Management. Organizations are in search of VIM solutions that are able to seamlessly integrate with container orchestration platforms. Mature 5G rollouts and telecom networks upgrades, as well as the adoption of Edge Computing, require flexible solutions capable of managing a cloud-edge continuum and successfully enabling new types of services. Organizations are eager to benefit from multi-cloud strategies, which require VIMs that can easily manage multiple types of resources across different types of distributed cloud. "*Virtual Infrastructure Manager Market size was valued at USD 40 Billion in 2023 and is expected to reach USD 80 Billion by the end of 2030 with a CAGR of 5.71% during the forecast period 2024-2030.*"[2] Among the growth drivers of the VIM Market, we can highlight the ongoing digital transformation, the aim of continuously reducing costs, the scalability and flexibility demanded by new telecom use cases, and the needs of network modernization, a concept that starts at the infrastructure.

The WAN Infrastructure Manager (WIM) can be seen as a specialized VIM that allows establishing connectivity between different network endpoints at different NFVI-PoPs in the context of a multi-site service Wide Area Network (WAN).[3] To set up network connectivity, the WIM can rely on the network/SDN controllers that manage the network at a lower level using various types of technologies and protocols. WIM abstracts the networking layer in a multi-site cluster deployment, easing provisioning and monitoring operations.

OpenNebula is a robust, flexible and widely used VIM, designed for heterogeneous infrastructure and able to manage any type of cloud. OpenNebula has developed the necessary capabilities to manage the life cycle of the NFVI software and hardware, keeping a live allocation of the physical and virtual resources but also maintaining an inventory of them. OpenNebula, through its rich and robust feature landscape, represents the foundation of the distributed and technology-agnostic cloud-edge continuum, by simplifying service delivery and maintenance, and reducing costs while ensuring the high quality and availability required for critical use cases.

[2] https://www.verifiedmarketreports.com/product/virtual-infrastructure-manager-market, retrieved in October 2024.
[3] https://osm.etsi.org/wikipub/index.php/WIM, retrieved in October 2024.

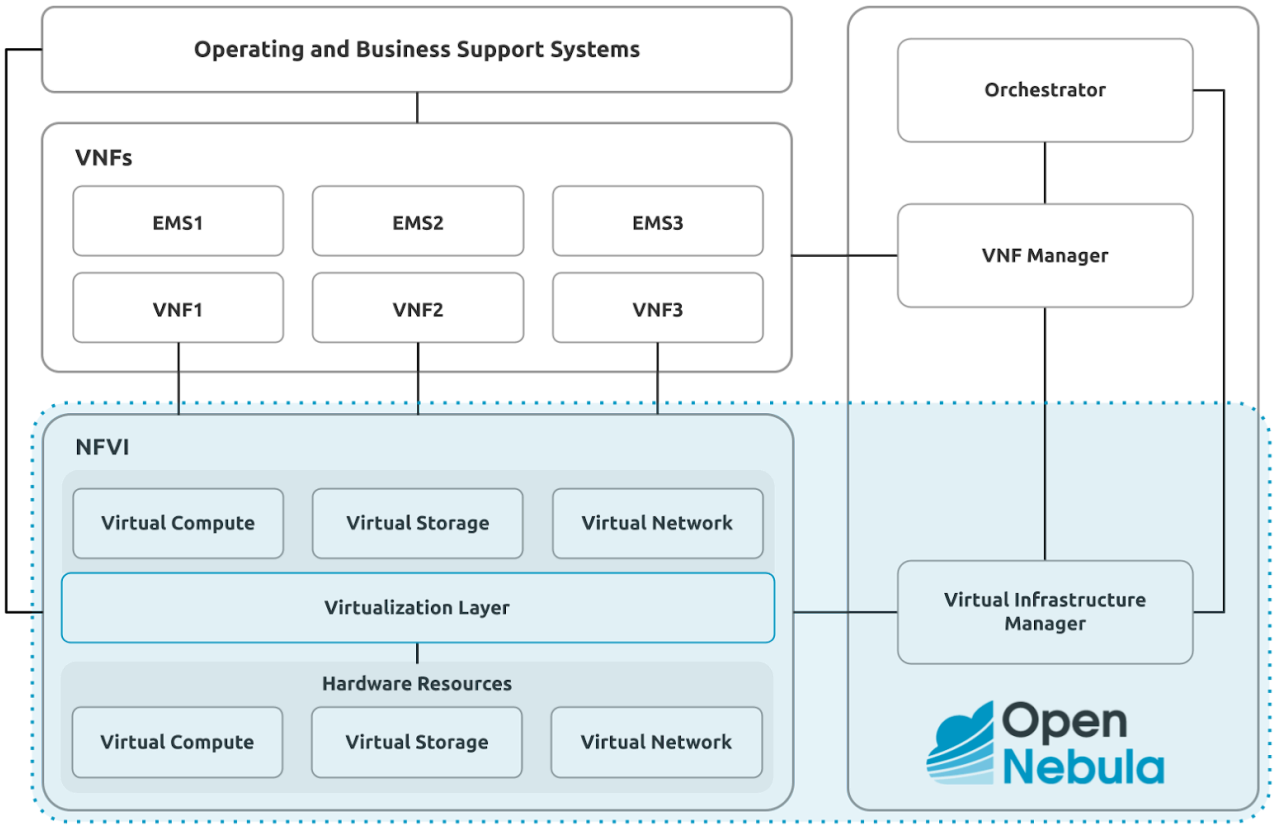


Figure 1: OpenNebula mapped over the NFV-MANO framework.

While various types of network operators are following different paths in their transition to cloud-native architectures, most of them are already implementing hybrid deployments containing CNFs and VNFs. In the case of the 5G networks, which are built to be cloud-native as one of their main goals, greenfield deployments are usually implemented following this pattern. Since everyone is aware of the advantages of virtualization, the adoption of telco-cloud-based networks is expected to grow. Migration to cloud-based network functions (positively) impacts all mobile network areas from RAN to Core Network, from Edge to OSS and BSS.

### 1.1. Re-Virtualization Trend in the Telco Cloud

Broadcom's VMware acquisition did not bring the best news for existing customers. According to Gartner's 2024 Hype Cycle for Data Center Infrastructure Technologies report,[4] Broadcom's new licensing approach, which combines the functionalities but also the costs of multiple VMware products into larger unified solutions, can increase TCO by 200 to 300%. This report calls attention to the two new concepts of de-virtualization and re-virtualization, highlighting that VMware's new licensing approach is considered one of the main drivers behind both of them.

The concept of re-virtualization was introduced into the market a few years ago. Multiple customers have chosen this step "to address a viability or commercial risk"[5] in order to overcome technical challenges or to support new trends and use cases in the telecom world. Undertaking such a process is not devoid of risks. It can cause an increase in total cost of ownership, create an additional operational burden, reduce the possibilities for fallback, introduce new needs in terms of training and development for staff, create an

[4] https://www.gartner.com/en/documents/5540595, retrieved October 2024.
[5] Ibid.

immature tooling landscape, cause interoperability issues and decrease reliability. Unless you choose OpenNebula.

OpenNebula Systems strongly believes in re-virtualization. It continues to work dynamically to remove the risks mentioned above, enhancing its feature set to ease users' path towards achieving this goal while minimizing downtime, as multiple users can confirm based on their successful experiences. Among OpenNebula's key features, we may highlight:

- Unified Management through a single pane of control.
- Easy migration from VMware (including dedicated features and workflow) as described in a dedicated White Paper.[6]
- Enhanced security using open-source solutions for both VMs and containers.
- Scalability, helping to adapt VM/container workloads to meet dynamic network demands.
- Coexistence and seamless integration with other platforms.

Gartner believes that re-virtualization or virtual-to-virtual migration has reached its peak, being applicable to 5-20% of companies.

### 1.2. OpenNebula - The Open Platform for Network Virtualization

OpenNebula[7] is a **simple, but powerful, open source solution to build and manage Enterprise Clouds and Edge environments**. It combines virtualization and container technologies with multi-tenancy, automatic provision, and elasticity to offer on-demand applications and services.

OpenNebula provides a single, feature-rich and flexible platform with **unified management of IT infrastructure and applications** that **avoids vendor lock-in** and **reduces complexity, resource consumption, and operational costs**. OpenNebula manages:

- **Any Application**: Combine containerized applications from Kubernetes with Virtual Machine workloads in a common shared environment to offer the best of both worlds: mature virtualization technology and orchestration of application containers.
- **Any Infrastructure**: Open cloud architecture to orchestrate compute, storage, and networking driven by software.
- **Any Cloud**: Unlock the power of a true hybrid, edge and multi-cloud platform by combining your private cloud with infrastructure resources from third-party virtual and bare-metal cloud providers such as AWS and Equinix Metal, and manage all cloud operations under a single control panel and interoperable layer.
- **Any Time**: Add and remove new clusters automatically in order to meet peaks in demand, or to implement fault tolerance strategies or latency requirements.

OpenNebula provides the necessary tools for running containerized applications from Kubernetes while ensuring enterprise requirements for your DevOps practices. It helps organizations to easily embrace Hybrid and Edge Computing, allowing them to grow their Enterprise Cloud on demand with infrastructure resources from third-party Public Cloud and bare-metal providers such as AWS and Equinix Metal. This disaggregated cloud approach allows for a seamless transition from centralized private clouds to distributed edge-like cloud environments. Companies can grow their private cloud with resources at cloud and edge data center locations, to meet peaks in demand or the latency and bandwidth needs of their workload. This approach involves a single management layer where organizations can continue using existing OpenNebula images and templates, keep complete control over their infrastructure, and avoid vendor lock-in.

[6] https://support.opennebula.pro/hc/en-us/articles/17225311830429-Migrating-from-VMware-to-OpenNebula-White-Paper
[7] https://support.opennebula.pro/hc/en-us/articles/360036935791-OpenNebula-Overview-Datasheet

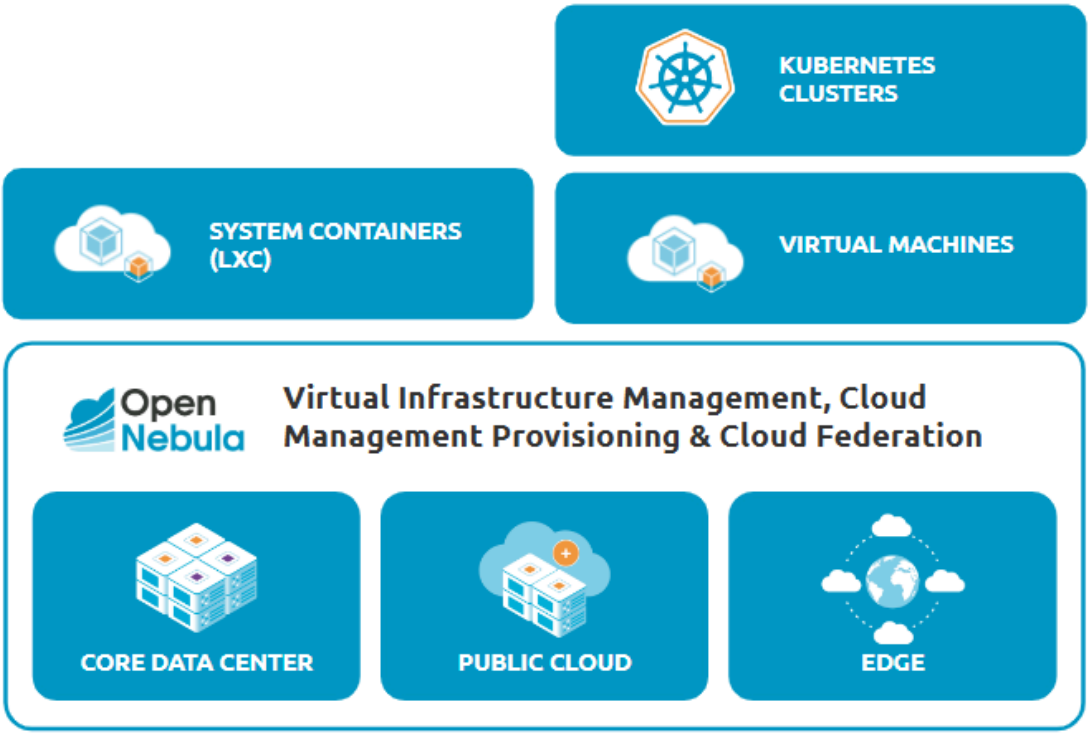


Figure 2: OpenNebula's flexibility.

The development of OpenNebula follows a bottom-up approach driven by the real needs of sysadmins, DevOps, and corporate users. OpenNebula is an **open source product** with a healthy and active community, commercially supported by OpenNebula Systems through its OpenNebula Subscription program. New versions are released on a regular basis and delivered as a single package with a smooth migration path. OpenNebula defines its short-term roadmap and plans the features for the next release guided by demands of its Sponsors, Customers, Users and Partners. A detailed list of planned features for the upcoming release of OpenNebula is available at the GitHub OpenNebula/One issues page. More information on the benefits of running an OpenNebula cloud can be found on the Key Features page.

Have a look at our Case Studies and Success Stories to learn more from our users about how they are putting OpenNebula to work.

## 2. PoC Setup and Testing Configuration

The next sections of this report include the required features and relevant details of the tests performed with a Telecom Operator in the context of its NFH Evolution PoC, where OpenNebula was placed as an Isolated VIM. The report aims to analyze the evolution of the current PoP architecture based on the VMware ESXi hypervisor, and to compare it with OpenNebula's functionalities as an open alternative.

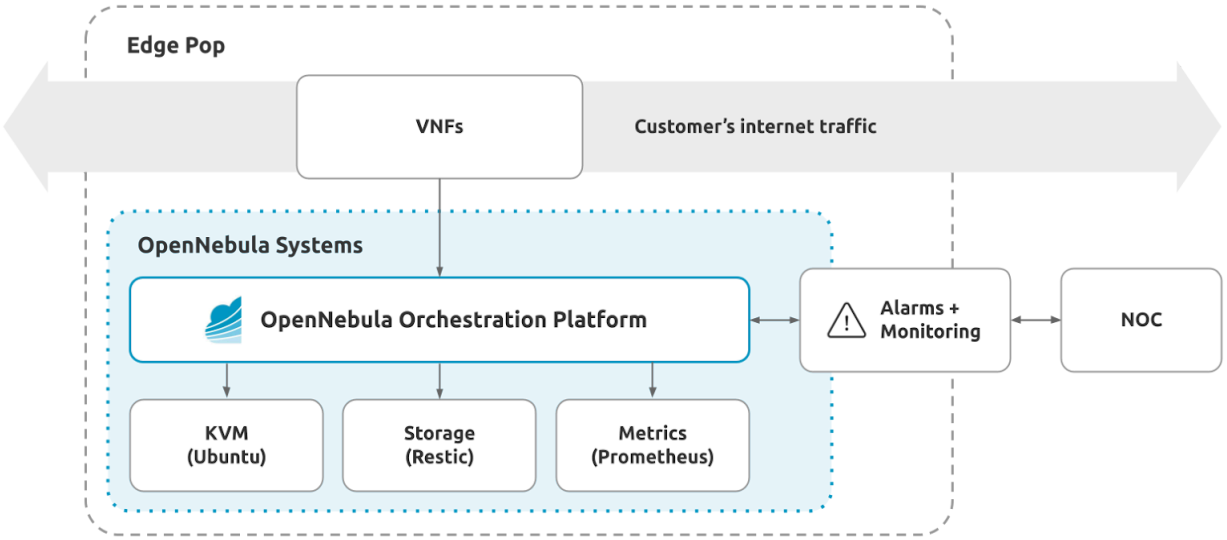


Figure 5: Edge PoP high-level architecture.

The detailed architecture used for the PoC is presented below, in Figure 6. It is based on two points of presence (PoP) using existing hardware.

The main components of the PoC were:

- **PoPs.** As highlighted above, the PoC included two points of presence. The Central PoP included the OpenNebula control daemons and the Image repository. Each PoP consisted of two servers, and featured a physical switch to emulate the PoP's internal network infrastructure.
- **Storage.** It was based on each host's local storage. Images were transferred to the host where the VM was deployed. To speed up transfer times, a cache space was configured in the Remote PoP.
- The Remote PoP configured a local NFS share to test the NAS drivers. This was based on a separate Datastore, in order to illustrate the differences between the two approaches.
- **Networking.** To increase coverage of the test cases, networking was based on Open vSwitch to implement the vSwitch components. Also, as a reference, some tests were made using Linux bridges. As part of the 5G edge cluster setup and bootstrapping, OpenNebula software automatically configures the 5G site ports, VLAN tags and flows in these switches.
- **Management Plane.** Cross-PoP communication for management was established from an L3 segment. In this network, we needed traffic on the SSH and the Monitor port (4124)
- **APIs and Web GUI** were installed on the management node and exposed on the 2633 and 9869 ports.

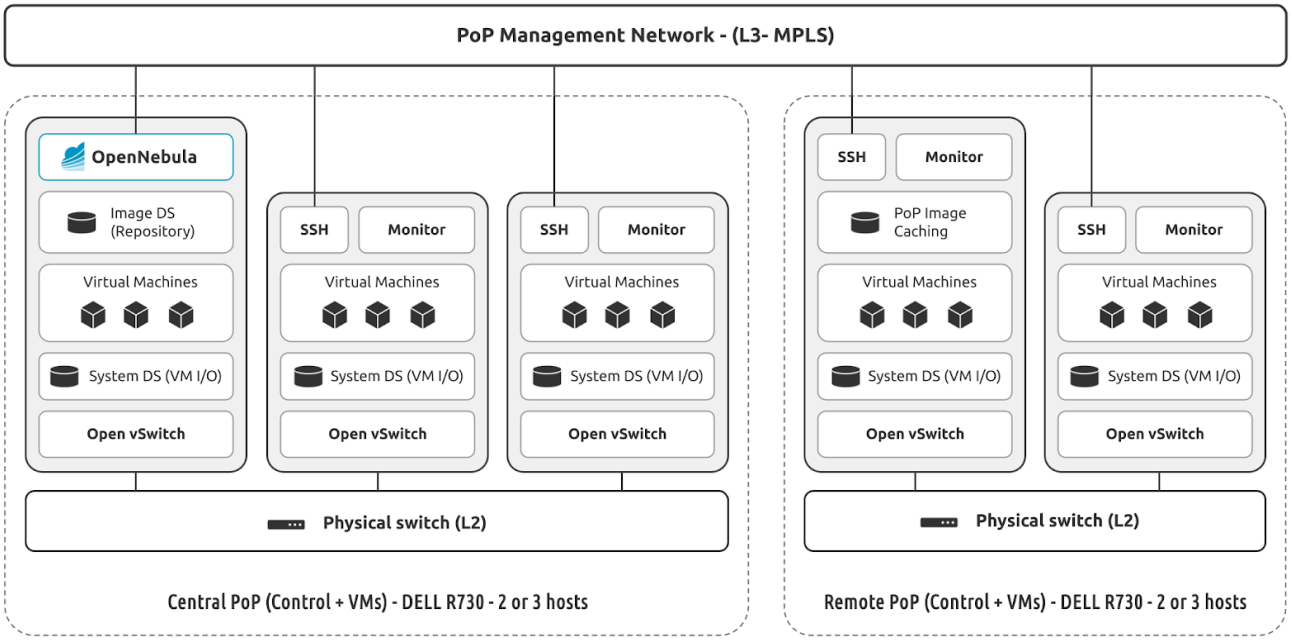


Figure 6: PoC architecture with 2 PoPs (Central and Remote) using 2 or 3 hosts each.

# PART B: Validation Plan

## 1. VIM Basic Features

In the tests performed, OpenNebula was selected to provide Virtual Infrastructure Management (VIM). OpenNebula services were installed in a High Availability configuration, where the OpenNebula core synchronizes the state and a leader is elected using a custom implementation of the Raft algorithm.

| **1.1. Storage** |
|---|
| **Description:** The VIM shall include support for Host Aggregation, onboarding the Storage and Compute on the same Host without requiring any manual control/coordination. In the Local Storage architecture, the hypervisor consumes/uses only local Disk on the Host and, using this local storage, it “provides” the vDisk to the VMs. |
| **Compliance**: Full |
| **Implementation**: In OpenNebula, when local storage is used with Host Aggregation (Availability Zone/Region) of at least two Hosts, the Storage and Compute are always located (onboarded) on the same Host. This does not require any manual control/coordination.<br><br>The Storage is never located on a different Host than Compute. The desired “behavior” for the platform is that Compute and Storage always reside on the same Host. Local Storage Datastore uses the local storage area of each Host to run VMs. Additionally, a storage area is needed for the VM disk image repository. Disk images are transferred from the repository to the hosts using the SSH protocol.<br><br>OpenNebula can also register images locally in each PoP. This feature may be automated through OpenNebula APIs and provision components.<br><br>OpenNebula also includes features such as configuring deduplication per backup job, implementing “per job” policy integration—grouping several VMs in a single job—assigning priorities to VMs in backup jobs, and managing backup jobs through the graphical web interface. |

| |
|---|
| Additionally, OpenNebula allows in-place backup restore (replacing a VM disk with a backup). |
| **More info:** Virtual Machine Backups Guide |

| **1.2. Capacity Planning and Optimal Resource Usage** |
|---|
| **Description**: The VIM shall support CPU pinning for vCPUs in VMs, and configuring CPU Prioritization for some VMs. Additionally, the VIM shall support customizable CPU overcommitment. |
| **Compliance**: Full |
| **Implementation**: The NFV Edge environment requires VM/CPU prioritization. OpenNebula allows to prioritize the vCPU of some VMs over others, without pinning; tenants have the autonomy to create normal and prioritized "VMs."<br><br>Before allocating a VM to a Host, the Scheduler checks that the capacity (CPU and memory) requested by the VM fits in the available capacity of the Host.<br><br>OpenNebula's CPU Overcommitment works in such a way that prioritized VMs are honored, although some performance degradation may occur in overcommitted scenarios. Additionally, tenants cannot overcommit a Host, since capacity checks are always enforced. Cloud admins have the ability to overcommit. |
| **More info:** Capacity Planning Guide |

| **1.3. Redundancy, Resiliency, Fault Tolerance and Recovery** |
|---|
| **Description**: The VMs deployed in the remote data center shall continue to provide service even in the event that Management and/or Control communication is lost. Once communication is recovered, the complete system must recover its service without human intervention. After recovery, the services must continue working. |
| **Compliance:** Full |
| **Implementation**: In the event that either Management or Control communication is lost between the Remote Data Center and the Central PoP, the VMs deployed in the remote Data Center continue providing service. Once communication is reestablished, OpenNebula allows for the complete system to resume service, without human intervention.<br><br>Services need to be deployed and configured across several Hosts, and a distributed consensus protocol must be established to provide fault-tolerance and state consistency across them. This type of deployment can withstand the failure of at least one Host, depending on the total number of Hosts. In order to preserve a consistent view of the system across servers, modifications to the system state are performed through a special node, the leader. The servers in the OpenNebula cluster elect a single node to be the leader. The leader periodically sends heartbeats to the other servers—the followers—to maintain its leadership. If a leader fails to send the heartbeat, followers are promoted to candidates and start a new election.<br><br>If the service was correctly configured and deployed for High Availability (on several Hosts) for both the Remote Data Center and the Central PoP, even in case of failure of a single Host the other host(s) will continue working with no service alteration. |
| **More info**: High Availability Guide |

| 1.4. Backup and Recovery |
|---|
| **Description**: A backup solution for VMs, functional and fully integrated with the proposed infrastructure (VIM + agents, Hypervisors, all system software) shall be provided. The backup solution must be able to recover a full VM, as a "black box." This implies restoring all VM parameters (compute configuration, vNICs configuration and order, etc.). The restoration process must perform a *direct restoration* to the VM's original Host, or to another Host if needed. |
| **Compliance**: Full |
| **Implementation**: OpenNebula provides several backup and recovery options. For NFV Edge deployments, Restic is the recommended choice. The backup server assumes that the underlying storage volumes feature an HA mechanism that replicates the backup repository contents. The backup service itself (restic) does not provide additional HA configurations. The OpenNebula backup mechanism can restore VM disks as new OpenNebula images. In turn, these images exist as files in the target image datastore backend (since we will be using an NFS-based image datastore), and as such they can be inspected with tools such as `libguestfs`[8] to extract particular files, recovering the state of those files at the time that the backup was performed. |
| **More info:** Backup Datastore: Restic |

| 1.5. Maximum Latency Between Central and Remote PoP |
|---|
| **Description**: Test OpenNebula's correct operation and performance in Long Fat Networks. |
| **Compliance**: Full |
| **Implementation**: The scenario for the PoC was configured with a delay of 200 milliseconds between the central/backup manager and the Hosts.<br><br>There is no limitation to the maximum Round-trip Time (RTT) between OpenNebula and the hypervisors. For the interconnection, standard values for WAN latency (~200 ms) and bandwidth (100-1000 Mbps) were assumed. It is important to note that higher latencies are supported. The main component affected by latency is image transfer, so higher latencies will translate to higher transfer times but will not impact on the operations of the Cloud.<br><br>The backup solution can be optimized to operate with "Long Fat Networks" or LFN (high latency, high bandwidth), with latencies over 200 ms. |
| **More info**: Backup Datastore: Restic |

| 1.6. Capacity Planning |
|---|
| **Description**: Describe the VIM's capacity and limits, and the response times at the threshold. |
| **Compliance**: Full |
| **Implementation**: To achieve a stable load of **30 API requests per second**, the following limits are |

[8] https://www.libguestfs.org/

recommended:

- Number of hosts = 1250
- Number of VMs = 20,000
- Average VM template size = 7 KB
- Maximum number of VMs per KVM Hypervisor = 500

With these limits, and under a load of 30 API req/s, the OpenNebula oned daemon can sustain a response time of 0.3s for single-object operations and 5s to retrieve the whole VM pool of 20,000 VMs. The number of requests per second may condition the response times of the manager.
The completion times of some operations, for example the deployment time of a VM, may depend on the number of concurrent requests (i.e. simultaneous VM deployments).

With 500 VMs in a single hypervisor, a monitoring sweep is completed in 43 seconds. The frequency of the monitor probes can be adjusted using this value. For example, for high-density servers VMs should be monitored every 60 seconds.

**More info**: Scalability Testing and Tuning

| **1.7. Authentication** |
|---|
| **Description**: The proposed solution must provide advanced authentication procedures. |
| **Compliance**: Full |
| **Implementation**: OpenNebula's authentication subsystem supports several authentication backends that can be configured concurrently:<br>- Built-in user/password and token authentication<br>- SSH key-based authentication<br>- X509 certificates-based authentication<br>- LDAP and AD authentication |
| **More info**: Authentication Configuration |

| **1.8. Access Control Mechanism** |
|---|
| **Description**: The proposed solution must provide an access control mechanism capable of limiting the usage of critical functions by non-allowed individuals or groups. |
| **Compliance**: Full |
| **Implementation**: OpenNebula's access control mechanism is based on permissions and groups. In OpenNebula, roles are associated to the permission of a group:<br>- Users can be part of multiple groups.<br>- Users can switch the group that they are operating within a session.<br>- Groups are associated with access rules for each resource type and access level (use, manage or admin).<br>- Resources are owned by a group, whose members can access the resources at the specified level.<br>- ACLs can be used to grant users or groups additional access to a resource or to sets of resources, at |

| |
|---|
| different levels<br><br>When creating a Virtual Machine, cloud admins can assign it to a given user or group. Additionally, VMs may be moved or assigned in bulk to a given group. |
| **More info**: Users and Groups |

| **1.9. Other Basic Features** |
|---|
| **Description**: Tenants must be able to manually configure the UUID of the VMs/instances.<br>Tenants must be able to configure the distribution of vCPUs between different socket configurations.<br>VM guests must be able to use several different vNIC MACs apart from the one created or provided by the Hypervisor. |
| **Compliance**: Full |
| **Implementation**: In OpenNebula, tenants can manually configure the UUIDs of VMs/instances, without the intervention of administrators. However, each tenant may only use this feature on the VMs that belong to it.<br><br>The same holds true for NUMA topology: tenants can configure the distribution of vCPUs among different VM socket VM configurations—as well as the number of sockets in a VM—without the need for administrator intervention.<br><br>Additionally, with no need for administrator intervention VM guests can use, on their own VMs, one or more vNIC MACs different from the one created or provided by the Hypervisor. Tenants can also configure the MAC addresses of vNICs. |
| **More info**: Virtual Topology and CPU Pinning |

## 2. VIM Advanced Features

| **2.1. Tenant Management** |
|---|
| **Description**: The VIM must offer a complete multi-tenant service, including the possibility of having multiple groups of users with different access rights and, implicitly, with different properties. |
| **Compliance**: Full |
| **Implementation**: OpenNebula is natively multi-tenant and allows for tenants to be defined by creating groups with limited permissions over the different resources. These permissions include:<br><br>• Instantiating Virtual Machines<br>• Loading tenant images<br>• Allowing custom templates for each tenant<br><br>Additionally, each tenant has its own independent Virtual Networks.<br>OpenNebula administrators can create new tenant groups, and limit these permissions. Tenant groups can have their internal administrators, who can create new users in the group.<br>It is possible to create images and change their ownership. Tenants can create their own images or use the ones already created by administrators. The administrator can limit quotas (CPU, RAM, etc.). |

**More Info:** Users and Groups Guide

**2.2. Images and Flavors**

**Description**: The VIM shall be able to expose an Image Datastore to all PoPs, and to provide access to a global marketplace that includes all the relevant network applications. Additionally, the VIM shall rely on a predefined set of quotas that are managed by various types of groups according to their permissions.

**Compliance**: Full

**Implementation**: In addition to those described in the previous section, in relation to the NFV Edge requirements OpenNebula includes the following features:

- OpenNebula features a VM image repository based on disk images, controlled by the central manager. Several default datastores (files, images and system) are used to store the disk images and deploy them to the hypervisors when a VM is instantiated.
- A VM can be instantiated in "persistent mode." In this mode, when the VM is deleted the disks are copied to the central manager as new images and a new template, which can be used to instantiate the VM again.
- Tenants' autonomy and quota allow them to create images and change their ownership. Tenants can create their own images or use the ones already created by administrators. The administrator can limit quotas (CPU, RAM, etc).
- OpenNebula supports quotas/limitations for special configurations such as VMs with "top" or "high" priority CPUs. These quotas allow setting specific provisioning limits (number of CPUs) for high VM/CPU priority VMs. This quota must be compatible with the quotas per data center.

**More info**: Managing VM Instances, Scheduler Configuration

**2.3. Automatic Host Placement**

**Description**: The VIM must also support automatic hosts placement for VMs. Automatic placement criteria for VMs must consider, at least, the use of CPU, RAM and storage in the target host. Additionally, the VIM must support requesting VM host selection from an external SW, via a REST API or similar mechanism.
The proposed solution must support automatic replacement of VM hosts during the whole life cycle, in accordance with VMs' resource usage (at least CPU, RAM and disk usage).

**Compliance**: Full

**Implementation**: OpenNebula's Scheduler considers CPU, RAM, and storage for automatic VM and Host Placement for each VM, and is also able to consider affinity/anti-affinity rules.
The OpenNebula Scheduler uses a matchmaking algorithm that implements the Rank Scheduling Policy to allocate VMs to Hosts. The goal of this policy is to prioritize those resources more suitable for the VM. A matchmaking request consists of two parts:

- Requirements, which the target resource needs to fulfill in order to be considered for allocation to the VM. Resources that do not fulfill the requirements are filtered out.
- Rank, or preferences, a function that ranks the suitable resources to sort them. Resources with a higher rank are used first.

OpenNebula uses this algorithm to schedule all resources types:

- Hosts, to select the Host where the VM will run. This takes into account CPU and RAM.
- System Datastores, to select the System Datastore to be used. This takes into account the storage requirements (size of the VM disks and available space in the datastores).
- Virtual Networks, to select the Virtual Networks to attach the VM interfaces in auto mode.

While the above fulfills the CPU, RAM and Storage requirements for this item, the matchmaking algorithm is very flexible and allows any arbitrary condition to be taken into account, if needed. For instance, it is possible to assign an arbitrary label named `COLOR` to the Hypervisor, specifying different values; and then configure VM templates to prefer or to deploy only on specific colors.
OpenNebula supports requesting deploying external software (via REST API or similar mechanism) to perform the VM Host Selection.

- When a VM is created, VIM sends a request to an external SW. The external SW will select the Host.
- The request will contain the VM's characteristics: Data Center, CPU, RAM, vDisk, vNIC, VM Flavor, VM prioritization, etc.
- The external SW will select the Host associated with the VM. It will reply to the VIM, informing the selected Host and Storage/Datastore.

**More info**: Scheduler Configuration Guide

**2.4. Manual Host Placement**

**Description**: The proposed solution must support manual host placement/selection for VMs. This means that the user creating the VM must be able to choose the Host and Datastore. Normally, manual Host placement for VMs is used in non-multitenant scenarios, where the cloud administrator selects the Host.

**Compliance**: Full

**Implementation**: Tenants with the appropriate permissions can manually select the Host to deploy a VM. VDC (Virtual Data Center) control can also be implemented in this case.

**More info**: Scheduler Configuration Guide

**2.5. Hosts Classification and Organization**

**Description**: The VIM must be able to classify and organize Hosts in groups. Administrators must be able to allocate a name to each group. The groups and the Hosts belonging to each group must be clearly represented and "linked" in the GUI and API.

**Compliance**: Full

**Implementation**: OpenNebula Hosts can be associated to a "Label" defined through the CLI, API or GUI. The label is visually represented as a tag in the GUI, which can be used to display only the Hosts with that particular label.

Additionally, Hosts can be aggregated in clusters that link together Hosts with common traits (such as sharing the same location, being homogeneous, etc). Clusters can also be labeled, so for instance it is possible to define the "Brazil" cluster which aggregates all clusters present in Brazil, which in turn aggregate all Hosts present in the country.

| |
|---|
| Labels can be dynamically defined, in which case they need to have at least one associated resource to exist. Alternatively they can be defined system-wide (available to all users of the platform), or defined by user (only available for this particular user, even if no resource is associated with the label) or by group (only available for this particular group, even if no resource is associated with the label). |
| **More info**: Hosts Management Guide |

| **2.6. Snapshotting** |
|---|
| **Description**: The proposed solution must support VM snapshots even if requested by tenants. |
| **Compliance**: Full |
| **Implementation**: OpenNebula allows tenants to request snapshots. There are no quotas associated with snapshots, though there are quotas for datastore usage. The amount of VM snapshot to account for can be configured per OpenNebula instance.<br><br>The snapshot preserves the state and data of a virtual machine at a specific point in time including disks, memory, and other devices, such as virtual network interface cards.<br><br>Persistent images can have snapshots if they are created during the life cycle of the VM. Images with snapshots cannot be cloned or made non-persistent. To run either of these operations the user would need to flatten the image first. |
| **More info**: Managing Virtual Machine Instances |

| **2.7. Manual Initiation of VM Movement and Relocation** |
|---|
| **Description**: The VIM shall support manual migration of VMs by the administrators, without service loss. |
| **Compliance**: Full |
| **Implementation**: Only the administrator can manually migrate VMs; tenants do not have permissions to migrate their VMs.<br><br>OpenNebula offers two types of migration:<br><br>• "Cold" migration: the VM is saved, powered off, or powered off hard; and VM files are transferred to the new resource.<br>• Hot migration: the VM stays powered on.<br><br>Also, compute and storage can be manually moved. |
| **More info**: Managing Virtual Machine Instances |

| **2.8. Infrastructure Usage/Consumption** |
|---|
| **Description**: The VIM shall offer real-time metrics for the guest's resource consumption. |
| **Compliance**: Full |

| |
|---|
| **Implementation**: OpenNebula features a guest-agent probe that provides real VM Guest memory consumption and other metrics on guests' consumption of Host resources. |
| **More Info**: Monitoring Driver |

## 3. Networking and EPA Features

OpenNebula's Enhanced Platform Awareness (EPA) implementation enables fine-grained matching of workload requirements to platform capabilities. EPA features provide OpenNebula with an improved understanding of the underlying platform hardware (HW), which allows it to accurately assign the workload to the best HW resource.[9]

In the tests performed, the first requirements for implementing EPA features emerged. Since then OpenNebula has evolved, and the number of its EPA features has continuously grown.

Initial demands included NUMA Affinity, and CPU Pinning and Threading policies. NUMA Affinity concerned VMs' CPU and RAM allocation and how it is mapped to the NUMA nodes. CPU Pinning referred to the exclusive usage of CPU cores. Virtual Machine vCPUs always run on the same physical core, avoiding movement across available physical cores; an assigned physical core will only change if the VM is stopped and restarted. The logical core is not used by other VMs or systems. CPU isolation involves not sharing the logical core with other vCPUs belonging to the system (hypervisor, agents...) or tenant loads. This feature is controlled by a boolean parameter:

- `Isolation=YES`. No vCPU can be scheduled in the same core as this VM. Equivalent to disabling HyperThreading.
- `Isolation=NONE`. vCPUs of this VM can be scheduled in the same core (different thread) as other VMs.

In this context, for these features the following tests were performed:

- Individual tests for the three features (Pinning, Isolation, NUMA Affinity).
- Verification of Tenants' autonomy, and how to use/configure it.
- Verifications of how an administrator can control/limit EPA consumption.
- Verification of how the NUMA node is selected during VM onboarding, and if the NUMA node remains the same during the VM's life cycle.

As described in the dedicated White Paper, OpenNebula offers a rich EPA feature set. This is the result of a continuous evolution in the past years, adapting to customer demands and adopting the latest open-source technologies.

The networking features were tested taking the following into consideration:

- The test is focused on L2 connectivity.
- The Open Cloud Networking Setup.
- Traffic must follow the shortest path between origin and destination.
    - For example, for communications between a VM and the IP network, data must follow the path VM -> Physical Switch -> IP Network.

[9] https://opennebula.io/white-papers/get-opennebula-enhanced-platform-awareness-white-paper/

| **3.1. OVS-DPDK** |
|---|
| **Description**: The VIM shall support vSwitches and their configuration without any manual intervention in the managed Hosts. |
| **Compliance**: Full |
| **Implementation**: OVS-DPDK is an industry standard virtual switch accelerated by DPDK. In this mode OpenNebula will create and configure OVS bridges and ports. This mode is recommended for use with NUMA+Hugepages. When using the DPDK backend, the OpenNebula drivers will automatically configure the bridges and ports accordingly.[10]<br><br>Open vSwitch with accelerated datapath (DPDK) is completely configured through the OpenNebula drivers and deployment system.<br><br>In OpenNebula the utilization of virtual switches encompasses the creation of:<br><br>• Flat Networks<br>• VLAN External Networks with the following functionality:<br>  ○ Parameters can be overwritten in the guest OS.<br>  ○ The IP parameters in Virtual Networks include automation and leasing control.<br>  ○ If all networks have the same IP parameters:<br>  ■ Tenants can configure a network in the guest once the VM is deployed.<br>  ■ When networks have the same configuration parameters (such as IP leasing), conflicts are avoided, and cannot arise since OpenNebula is aware of the leases.<br>  ○ The same VLAN can be used by two or more tenants.<br>• VLAN compliance with 802.1q for Access and Trunk<br>• QinQ tunneling<br>• OVS configuration is global, managed by OpenNebula, and distributed across all hosts.<br>• Uplink topology allows for one ToR switch failure without losing features/performance. |
| **More Info**: Open vSwitch Networks |

| **3.2. SR-IOV** |
|---|
| **Description**: SR-IOV will need to be supported and configurable.<br>The sharing of physical NICs shall be managed by administrators. |
| **Compliance**: Full |
| **Implementation**: In OpenNebula the utilization of SR-IOV encompasses the creation of:<br><br>• Flat Networks<br>• VLAN External Networks<br><br>OpenNebula uses a scheduler to assign the NIC and VF to the VM, but currently it is not possible to assign the NIC manually.<br><br>• Once the VM is instantiated, SR-IOV/PCI-PT cannot be attached or detached via the GUI—currently, it is only possible to pre-define the VM using the template. Once the VM is instantiated, it is not possible to attach new SR-IOV or PCI-PT NICs. |

[10] https://docs.opennebula.io/stable/open_cluster_deployment/networking_setup/openvswitch.html#open-vswitch-with-dpdk

- It is possible to attach (and live-attach) PCI and SR-IOV interfaces by simply selecting the device by its address, ID, vendor or class.

It is possible to assign a specific PCI to a vNIC, whether the VM is running or powered off. It is also possible to assign a Virtual Network, and advanced parameters such as “spoof check” or “trusted mode.”

In OpenNebula, only administrators can deploy SR-IOV/PCI-PT interfaces and make the corresponding configuration in the Hypervisors and the Central Manager; there are no quotas or delegation of quotas for tenants.

**More info**: Managing Virtual Machines Instances

| **3.3. PCI-PT (Passthrough)** |
|---|
| **Description**: The administrators of the VIM can deploy PCI-PT interfaces and assign a specific PCI to a vNIC. |
| **Compliance**: Full |
| **Implementation**: OpenNebula discovers, tracks and allocates devices to VMs in the KVM hypervisor, and also allows admins to select which devices can be hotplugged. In OpenNebula it is possible to assign a specific PCI to a vNIC, with the VM running or powered off. It is possible to control VLANs, since a Virtual Network can be assigned to a PCI-PT vNIC in the VM; and only administrators can deploy SR-IOV/PCI-PT interfaces. Network devices are also integrated with the Network stack, allowing guests to identify passthrough devices.<br><br>OpenNebula supports “live/warm” configurations (access/trunk mode, VLAN, etc.); and the ability to define the maximum number of vNICs per VM connected as SR-IOV and as PCI-PT. |
| **More info**: PCI Passthrough |

| **3.4. NUMA Awareness** |
|---|
| **Description**: The VIM must support the definition of multiple virtual NUMA topologies including asymmetric configurations. |
| **Compliance**: Full |
| **Implementation**: OpenNebula offers great flexibility to define virtual NUMA topologies and map them to the physical configuration of the Host, including several pinning policies and support for asymmetric configurations. This setup ensures that vCPUs executing processes and the memory used by these processes are on the same NUMA node. It opens the possibility of defining the placement of the sockets (NUMA nodes) in the Hypervisor NUMA nodes. In this scenario, each VM socket will be exposed to the guest OS as a separate NUMA node with its own local memory. Additionally, by modifying the attributes of NUMA nodes, it is possible to create asymmetric NUMA configurations, distributing the VM resources unevenly across the nodes. |
| **More info**: Virtual Topology and CPU Pinning |

| 3.5. IO-based NUMA Scheduling and NUMA IO Affinity |
|---|
| **Description**: The VIM shall offer the possibility of implementing NUMA, based on its architecture and requested VM topology. |
| **Compliance**: Full |
| **Implementation**: The scheduling process is slightly modified when a pinned VM includes PCI passthrough devices. In this case, the NUMA nodes where the PCI devices are attached are prioritized to pin the VM vCPUs and memory to speed up I/O operations. No additional configuration is needed.<br><br>OpenNebula's NUMA-IO Affinity is only considered by pinned VMs, and enabled automatically when a PCI device is included in the VM definition. NUMA allocation is performed considering the NUMA architecture of the Host and the requested VM topology. This functionality creates an affinity that associates a VM with the same NUMA nodes as the PCI device passed into the VM. |
| **More info**: CPU and NUMA Pinning |

| 3.6. CPU and NUMA Pinning |
|---|
| **Description**: The VIM shall support CPU and NUMA pinning in order to match virtual topologies. |
| **Compliance**: Full |
| **Implementation**: OpenNebula retrieves information about which models and CPU features are available on the Hypervisors. It exposes host CPU features to managed guests. It supports several pinning policies, and also includes a NUMA scheduler in order to match virtual topologies and hypervisor configurations as closely as possible. When it is necessary to expose the NUMA topology to the guest, a pinning policy will need to be defined in order to map the resources of each virtual NUMA node (memory and vCPUs) onto the hypervisor nodes.[11]<br><br>When no pinning policy or NUMA node affinity is set, OpenNebula will select the node with the highest number of free hugepages, to try balancing the load. |
| **More info**: CPU and NUMA Pinning |

| 3.7. CPU Threading Policies |
|---|
| **Description**: The VIM will support CPU isolation, avoiding simultaneous usage of the two logical cores. |
| **Compliance**: Full |
| **Implementation**: OpenNebula can work with four different policies:<br><br>• **Core**: each vCPU is assigned to a whole hypervisor core. No other threads in that core will be used. This policy can be useful to isolate the VM workload for security reasons.<br>• **Thread**: each vCPU is assigned to a hypervisor CPU thread.<br>• **Shared**: the VM is assigned to a set of the hypervisor CPUs shared by all the VM vCPUs. |

[11] https://docs.opennebula.io/stable/management_and_operations/host_cluster_management/numa.html#cpu-and-numa-pinning

| |
|---|
| • **None**: the VM is not assigned to any hypervisor CPUs. Access to resources (i.e. CPU time) will be limited by the CPU attributes.<br>VM memory is assigned to the closest hypervisor NUMA node where the vCPUs are pinned, to prioritize local memory access. |
| **More info**: CPU and NUMA Pinning |

| **3.8. Hugepages Support** |
|---|
| **Description**: The proposed solution must support “hugepages” memory configuration for VMs. |
| **Compliance**: Full |
| **Implementation**: This feature allows the usage of memory pages larger than the standard size. To enable the use of hugepages for the memory allocation of the VM, it is only necessary to add the desired page size in the TOPOLOGY attribute. Scheduling includes hugepage requirements and availability in its placement algorithm—OpenNebula will look for a Host with enough free pages of the requested size to allocate the VM. The resources of each virtual node will be placed as close as possible to the node providing the hugepages. |
| **More info**: Using Hugepages |

The latest versions of OpenNebula have increased and improved compliance with EPA features. As highlighted above, OpenNebula 6.10 supports a suite of multiple enablers for precisely mapping the most innovative workload requirements to platform capabilities, and to ensure optimal use of resources in a wide range of use cases. OpenNebula’s EPA White Paper shares further details about other complementary features needed for managing distributed VNFs and CNFs.

## 4. Ready for a Test Drive?

You can evaluate OpenNebula and build a cloud in just a few minutes by using miniONE, our deployment tool for quickly installing an OpenNebula Front-end inside a Virtual Machine or physical host, which you can then use to easily add remote resources.

## 5. Conclusions

Virtualized architectures allow companies to overcome the limitations of traditional network infrastructures. By decoupling the network functions from the hardware, operators can benefit from an agile methodology for deploying new services and upgrading existing offerings. The Management and Orchestration layer is a key feature in a virtualized environment, efficiently handling management, control, monitoring and provisioning not only of the physical and virtual infrastructure resources, but also of the software deployed on top.

OpenNebula Systems has facilitated this new technology paradigm through its vendor-agnostic approach and through a continuous upgrade of its marketplace, allowing users to choose from a variety of vendors but also to retain the ability to engage new ones according to their requirements.

The tests performed with various users, including Tier-1 carriers, are demonstrating the suitability of OpenNebula as a VIM and Cloud-Edge manager for heterogeneous virtualized environments. Considering the foreseen market increase for this share, OpenNebula is becoming an extremely attractive alternative to big players such as OpenStack or VMware. Its open-source-driven approach, its robust feature set, its flexibility and continuous-improvement-driven development makes it extremely relevant in the dynamic telecom world.

The complex use cases and workloads outlined in this use case require perfect coordination between physical and virtual resources in an extremely distributed environment. OpenNebula has positioned itself, through the advantages it offers and its working approach, as the answer to these demands. OpenNebula's latest evolution in its last three major releases enforces its suitability and enriches its feature set in accordance with the goals of EU-driven collaborative projects, the dynamic needs of telecom companies, and the constant evolution of the market.

This white paper depicts some of the most important features that OpenNebula offers in support of Virtual Infrastructures and their management. Companies and organizations can step up their distributed network modernization and/or re-virtualization towards a simpler and more efficient way of designing, deploying and managing virtual infrastructures, with the help of OpenNebula.

## LET US HELP YOU DESIGN, BUILD, AND OPERATE YOUR CLOUD

### CONSULTING & ENGINEERING

Our experts will help you design, integrate, build, and operate an OpenNebula cloud infrastructure

### OPENNEBULA SUBSCRIPTION

Get access to our Enterprise Edition and to our support and exclusive services for Corporate Users

### CLOUD DEPLOYMENT

Focus on your business and let us take care of setting up your OpenNebula cloud infrastructure

**Sign up for updates at OpenNebula.io/getupdated**



LINUX FOUNDATION SILVER MEMBER · CLOUD NATIVE COMPUTING FOUNDATION SILVER MEMBER · LF EDGE GENERAL MEMBER · GAIA-X data-infrastructure.eu

Rev1.0_20241126