Open Cloud Reference Architecture

Version 2.2 – April 2021

Abstract

The OpenNebula Cloud Reference Architecture is a blueprint to guide IT architects, consultants, cloud administrators, and field practitioners in the design and deployment of private, hybrid, and edge clouds fully based on **open source platforms and technologies**. It has been created from the collective information and experiences from hundreds of users and client engagements. Besides the main logical components and interrelationships, this document includes references to software products, specific configurations, and requirements of infrastructure platforms recommended for a **smooth OpenNebula installation**. Three optional functionalities complete this architecture: high availability, true hybrid and edge for workload outsourcing, and federation of geographically dispersed data centers.

This document describes the reference architecture for Basic and Advanced OpenNebula clouds, providing recommended software for the main architectural components, including support for **KVM-based Virtual Machines, LXC system containers, and Firecracker microVMs**. Each section provides information about other open source infrastructure platforms **tested and certified** by OpenNebula Systems to work in enterprise environments. To complement these certified components, the OpenNebula **add-on catalog** can be browsed for other options supported by partners and the OpenNebula Community. There are of course other components in the open cloud ecosystem that are not part of the reference architecture, but are nonetheless important to consider at the time of designing a cloud (e.g. configuration management and automation tools for managing cloud infrastructure and large numbers of devices).

Contents

Glossary

AD	Active Directory
COW	Copy on Write
DB	Database
DC	Datacenter
HA	High Availability
NFS	Network File System
NIC	Network Interface Card
VDC	Virtual Data Center
VM	Virtual Machine

1. What is OpenNebula?

Enterprise cloud computing is the next step in the evolution of data center (DC) virtualization. **OpenNebula is a powerful, but easy-to-use, open source platform to build and manage enterprise clouds and virtualized DCs**. It combines existing virtualization technologies with advanced features for multi-tenancy, automatic provision, and elasticity. The development of OpenNebula follows a bottom-up approach driven by the real needs of sysadmins, DevOps, and users.

OpenNebula is an **open source product** with a healthy and active community, and is commercially supported by OpenNebula Systems. OpenNebula releases are produced on a regular basis and delivered as a single package with a smooth migration path. More information on the benefits of running an OpenNebula cloud can be checked on the key features page.[1]

2. High Level Reference Architecture

A standard OpenNebula Cloud Architecture consists of:

- The **Cloud Management Cluster** with the Front-end node(s), and
- The **Cloud Infrastructure**, made of one or several workload **Clusters** with the hypervisor nodes and the storage system, which can be located at multiple geographical locations, all interconnected with multiple networks for internal storage and node management, and for private and public guest (VM or container) communication.

An OpenNebula Cloud Infrastructure can combine multiple clusters with different configurations and technologies to better meet your needs. In general, there are two types of Cluster models that can be used with OpenNebula:

- **Edge Clusters:** can be deployed on demand both on-premises and on public cloud and edge providers, with a high degree of integration and automation. For more details, please refer to our white paper on the Edge Cloud Architecture.[2]
- **Customized Clusters:** typically these are deployed on-premises to meet specific requirements. You can find more details about this model below.

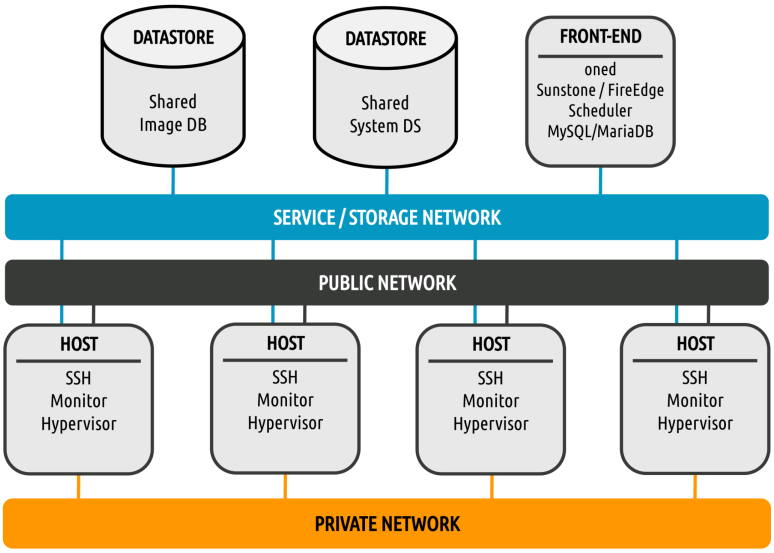

Figure 1. Reference Architecture, a bird's eye view.

[1] https://opennebula.io/discover/
[2] https://support.opennebula.pro/hc/en-us/articles/360050302811-Edge-Cloud-Architecture-White-Paper

Figure 1 shows the bird's eye view of an OpenNebula cloud with a single Front-end node and a Customized Cluster. OpenNebula services run in the Front-end, connected to the hypervisors through the Service Network. The Front-end uses this network to monitor the status of the hypervisors and VMs (Virtual Machines when using KVM, system containers when using LXC, or microVMs when using Firecracker), as well as to initiate VM or for storage-related operations.

Virtualization or container hypervisors are also connected to the storage backend of the cloud through the Storage Network. Given the low network traffic required by OpenNebula to operate, this network can be the same as the dedicated Storage Network. The storage backend is responsible for providing storage support for the running VMs (System Datastore) and for the image repositories (Image Datastore).

VMs usually require two types of network interconnections: private and public. The Private Network implements isolated virtual networks (VLAN) for the internal communication of VMs. Access to each virtual network can be restricted to different users or groups or limited through quotas. Additionally, some VMs need to communicate to the world so access to a Public Network connected to the Internet is recommended for some hosts.

Deciding which is the right implementation for your cloud depends on your performance, scalability, and availability requirements. You should also take into account existing storage and network infrastructure, available budget for new hardware, licenses, and support, and the skills and IT staff resources that this operation might require. Based on our broad experience deploying clouds for our customers in a variety of scenarios, these are the two types of Customized Cluster implementations that we would recommend:

- **Basic**: Simple and low-maintenance implementation that can be deployed on any infrastructure. It scales to medium-sized clusters and offers periodic snapshots for availability.
- **Advanced**: Complex alternative that supports larger clusters and availability based on replication. This requires more experience, dedicated hardware, and a higher amount of IT staff resources.

The particularities of the expected workload and hardware characteristics may impact the implementation of each of those models. Table 1 below shows the pre-sets for each one:

	Basic	Advanced
Operating System	Ubuntu or CentOS/RHEL (Specific OpenNebula packages have to be installed)	
Hypervisor	KVM | LXC | Firecracker	
Networking	VLAN 802.1Q	VXLAN
Storage	OneStor storage solution using qcow2 format for Image and System Datastores	Ceph Cluster for Image and System Datastores
Authentication	Native authentication or Active Directory	

Table 1. Summary of the Basic and Advanced implementations.

Each implementation, Basic or Advanced, can be customized by adding specific authentication methods, access to external providers or even setting up multiple zones in geographically distributed datacenters. The lines separating the two architectures are flexible, for instance it may be the case that VXLAN applies to set ups with few nodes and, conversely, VLAN may apply to large scale infrastructures, which also may prefer OneStor over Ceph due to its workload characteristics.

3. OpenNebula Front-end

The OpenNebula Front-end is a special node—a physical server or VM—devoted to orchestrating all cloud resources. The recommended Operating Systems for the Front-end are CentOS/RHEL and Ubuntu, and the hardware recommendations can be checked in Table 2. Please take into account that these recommendations are meant only as a guide.

When running on a Front-end with the minimum specifications described in Table 2, OpenNebula is able to manage a vCenter infrastructure with:

- Up to 1.250 managed nodes
- Up to 20.000 VMs in total

Memory	32 GB
CPU	16 CPU cores
Disk size	200 GB
Network	2 NICs

Table 2. Front-end hardware recommendations.

The Front-end provides the following services:

- OpenNebula management daemon
- Scheduler
- MySQL / MariaDB
- Administration and User GUI and APIs
- Optional OpenNebula services like OneFlow or OneGate

Note that some of these services are optional and can be also deployed in a different host (e.g. a dedicated MySQL/MariaDB cluster or separated Sunstone or OneFlow servers).

The maximum number of servers (virtualization hosts) that can be managed by a single OpenNebula instance strongly depends on the performance and scalability of the underlying platform infrastructure, mainly the storage subsystem. The general recommendation is that no more than 2,500 servers and 10,000 VMs should be managed by a single OpenNebula instance.

4. Virtualization Nodes

Compute nodes are responsible for providing VMs (KVM), system containers (LXC), or microVMs (Firecracker) with execution resources (e.g. CPU, memory, or network access). The recommended Operating Systems for virtualization nodes are Ubuntu and CentOS/RHEL. The recommended hypervisors in the reference architecture are KVM (Virtual Machines), LXC (system containers), and Firecracker (microVMs).

The configuration of the nodes will be homogenous in terms of the installed software components, OpenNebula administration user, and accessible storage. The characteristics of virtualization nodes are the same for the Basic and Advanced architectures. The recommendation is to minimize the number of nodes whilst maximizing the number of cores per node.

A key task when defining a cloud infrastructure is to correctly dimension the virtualization nodes according to the expected workload. Memory-wise, the recommendation is to have at least 1GB per core, but this also depends on the expected workload, that is, the characteristics of the VMs that are going to be run in the cloud. You also need to consider the overhead induced by the OpenNebula components running on the node. As a reference, for a 256GB memory and 24-core Xeon node, OpenNebula can comfortably manage 500 KVM Virtual Machines / 500 LXC containers / 1500 Firecracker microVMs.

Network-wise, the recommendation is to have at least a dedicated NIC for the storage backend and the control plane; as well as a dedicated NIC to route the VM traffic.

Besides KVM, LXC and Firecracker, OpenNebula comes with native support for VMware vCenter.[3]

5. Storage

Storage is one of the most critical aspects of a cloud infrastructure and needs to be carefully planned to avoid bottlenecks. OpenNebula works with two different sets of datastores:

- The system datastore, which sustains the disks of the running VMs and other files associated with the VM, such as context CD images and VM checkpoints (i.e. for suspended VMs).
- The image datastore, which contains the catalog of images suitable to build new VMs.

OpenNebula provides a variety of ways for Virtual Machines and containers to access storage. It supports multiple traditional storage models, including NAS, SAN, NFS, iSCSI, and Fiber Channel (FC), which allow virtualized applications to access storage resources in the same way they would do on a regular physical machine. OpenNebula also supports a number of distributed Software-Defined Storage (SDS) solutions like Ceph, GlusterFS, StorPool, and LINSTOR. These allow cloud admins to create and scale elastic pools of storage and hyperconvergence deployments. For more details, please refer to our white paper “Choosing the Right Storage for your Cloud”.[4] We recommend the following two storage set-ups corresponding to the Basic and Advanced architecture.

Basic Architecture

The proposed storage for the Basic architecture is based on OneStor, a local storage solution fully supported by OpenNebula Systems that has been developed for the efficient management of disk images in OpenNebula cloud environments. It brings recovery and migration enterprise features while maintaining low-maintenance and low-cost benefits. OneStor combines a 3-tier global architecture for application image distribution with an enhanced SSH transfer mode with replica caching, backup, and snapshotting mechanisms that greatly improves its scalability, performance, and reliability. Application images are based on files with a qcow2 format, which reduces file transfer and instantiation times. Images can be persistent and their changes copied back to the Image Datastore after VM shutdown.

OneStor has been implemented with lightweight technology components that already exist in the Linux OS. It accommodates any deployment model, both on physical and virtual resources, increasing the reliability of the storage backend. It leverages modest hardware requirements, including standard SATA SSDs and 10Gbps networks. Moreover, its deployment follows a hyperconverged approach that does not require dedicated servers for a distributed storage system to be implemented. This approach reduces the overall complexity of this solution, enabling the use of the local storage area of the Cluster’s hosts.

[3] https://vOneCloud.opennebula.io
[4] https://support.opennebula.pro/hc/en-us/articles/360019581717-Choosing-the-Right-Storage-for-Your-Cloud-Report

Advanced Architecture

For larger clusters and mission-critical workloads, a Ceph cluster is recommended as the storage backend, using its own network for storage distribution. Ceph pools will back the OpenNebula images datastores to hold golden images, as well as the system datastores to hold runtime VM disks.

The Ceph cluster will provide high availability, making the data accessible even when one of the storage nodes is down. The recommendation is to have separate servers for the Ceph cluster and not to mix them with the virtualization nodes. At least three servers are needed, with 5 disks of 1TB each, 16GB of RAM, 2 CPUs of 4 cores each, and at least 2 NICs.

OpenNebula supports various storage systems—including SAN cabinets exported by Fibre Channel, NetApp and other NAS models, local storage managed by SSH—and various file systems or block devices—e.g. LVM, VMFS, etc.

6. Networking

Networking needs to be carefully designed to ensure reliability of the cloud infrastructure. Depending on the size of the cloud, two recommended configurations are given for Basic and Advanced OpenNebula clouds. Both proposals enable the use of Security Groups, allowing inbound/outbound traffic in VMs' network interfaces.

Basic Architecture

The proposed network for Basic architecture is based on three networks. The virtualization nodes are connected to all the networks that are part of the cloud infrastructure. The recommendation is to use at least 10Gbps switches that support VLAN trunking to sustain these networks.

Only one interface connected to the Service Network is needed in the Front-end. This interface would be used by OpenNebula to connect to the shared file system server and the virtualization nodes via SSH protocol. Isolation between different private virtual networks can be achieved using 802.1Q VLAN tagging, with different VLANs for different private networks.

Private Network	Communication between VMs. It is important to assign contiguous VLAN identifiers to ports in the switch connected to this network, since the network configuration of OpenNebula will be using 802.1Q VLAN tagging
Public Network	To serve VMs that need internet access
Service Network	For Front-end and virtualization node communication—including inter-node communication for live migration—as well as for storage traffic
Storage Network	To serve the shared file system to the virtualization nodes (optional)

Table 3. Proposed networks for Basic Architecture.

Advanced Architecture

For larger clouds, a dedicated Storage Network is recommended. The virtualization nodes are connected to all the networks that are part of the cloud infrastructure. The advice is to use 10Gbps switches to sustain the Storage, Private, Public, and Service networks.

Private Network	Communication between VMs
Public Network	To serve VMs that need internet access
Service Network	For Front-end and virtualization node communication—including inter-node communication for live migration—as well as for storage traffic
Storage Network	To serve the Ceph pools to the virtualization nodes

Table 4. Proposed networks for Advanced Architecture.

The Advanced architecture assumes the use of VXLAN, a network virtualization technology designed for dealing with large cloud deployments, encapsulating Ethernet frames within UDP packets and thus solving the 4096 VLAN limit problem. The requirement for this is to be able to use the multicast address for the Service Network. It is worth noting that there is a limitation in the Linux kernel whereby only the handling of a maximum of 20 different VXLAN ids in the same hypervisor is currently allowed.

OpenNebula comes with native support for other network technologies, suitable for additional security and flexibility requirements. The available options include, for example, ebtables for Layer 2 isolation, the use of OpenvSwitch for advanced network functionalities, port-group and vSwitch support for VMware, or the use of a Virtual Router for RADVD, DNS, DHCP, port forwarding, etc.

7. Virtual Machines & Guest Support

Virtual Machine images must contain the OpenNebula contextualization packages for OpenNebula to be able to correctly pass the necessary network and configuration details on to the running VMs. OpenNebula Contextualization packages allow configuration and information sharing between the OpenNebula interface and the guest Operating System of the VM (e.g. scripts can be passed to the VM so they are run at boot time).

The following list contains a sample of guest OSs that are supported by the OpenNebula Contextualization packages on KVM (an exhaustive list is available in the official documentation):[5]

- CentOS >= 7
- Red Hat Enterprise Linux >= 7
- Debian >= 8
- Ubuntu >= 14.04
- Windows >= 7
- Windows Server >= 2008

LXC can only create Linux instances given that hosts and containers must be able to share the same kernel. Similarly, Firecracker only supports Linux guests.

8. Marketplace

OpenNebula has access to its own Marketplace, which enables users to import images from a public repository (containing images of common use that have been tested and certified by OpenNebula Systems) or from private repositories. These images can be added to a datastore and used by existing VM templates or instances. OpenNebula's Public Marketplace features images in qcow2 format. In this way, images can be used by KVM hypervisors, LXC system containers, and vCenter Servers.

[5] https://github.com/OpenNebula/addon-context-linux & https://github.com/OpenNebula/addon-context-windows

OpenNebula users can easily download, contextualize, and add from other public marketplaces, including Linux Containers and TurnKey Linux. From version 5.12 onwards, OpenNebula also comes with seamless integration with Docker Hub, permitting direct execution of Docker Hub images on any hypervisor in your OpenNebula open cloud.

	KVM	LXC	Firecracker	vCenter
OpenNebula Marketplace[6]	✔			✔
Docker Hub[7]	✔	✔	✔	
Linux Containers[8]	✔	✔	✔	
TurnKey Linux[9]	✔	✔	✔	

Table 5. Compatibility between public marketplaces and virtualization technologies supported by OpenNebula.

As for private repositories, there are two possibilities:

- HTTP Marketplace, where images are accessible through a HTTP server (e.g. Apache, Nginx).
- S3 Marketplace, where images are accessible through an Amazon S3 API.

Private marketplaces can be used for automatic VM backups. A regular interval can be defined where the disks of VMs that have been marked for backup will be saved (along with the VM metadata as known by OpenNebula) to the chosen private marketplace. A restore tool is also available to recover the VM.

9. Authentication

Either the native OpenNebula subsystem or an LDAP/Active Directory (AD) server can be used for authentication purposes. In both cases, the OpenNebula cloud will be accessible to users through the CLI and the Sunstone GUI. With the native OpenNebula authentication subsystem, users' details and credentials (username/password) will be kept in the OpenNebula database and groups will be generated as needed.

Alternatively, users can be authenticated against a corporate LDAP/Active Directory (AD) server, which has to be accessible through the service network. Users are created and added to the appropriate OpenNebula user DB table after the first use. Groups of users will be created as needed and access to resources will be assigned to them through the definition of Virtual Data Centers (VDCs).

OpenNebula natively supports several authentication mechanisms, like SSH keys and X509 credentials.

10. Cloud Access Model

The Cloud Access Model in OpenNebula is based on VDCs (Virtual Data Centers). A VDC is a fully-isolated virtual infrastructure environment where a Group of users (or optionally several Groups of users), under the control of a Group Admin, can create and manage compute and storage capacity. The users in the Group,

[6] https://marketplace.opennebula.io
[7] https://hub.docker.com
[8] https://images.linuxcontainers.org
[9] https://www.turnkeylinux.org

including the Group admin, would only see the virtual resources and not the underlying physical infrastructure. The Physical Resources allocated to the Group are managed by the cloud administrator through a VDC. The resources grouped in the VDC can be dedicated to the Group, providing isolation at the physical level, too.

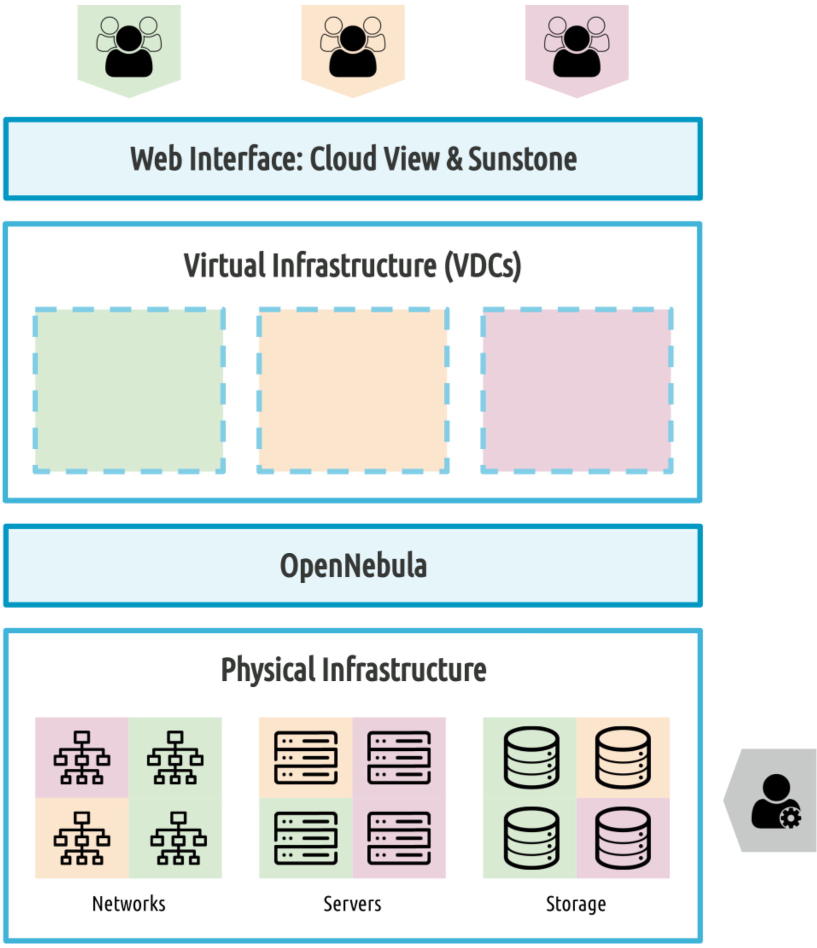

Figure 2. Resource Provisioning Model in OpenNebula.

Users are organized into Groups (similar to Projects, Domains or Tenants, as they are known in other environments). A Group is an authorization boundary that can be seen as a business unit—if you are considering it as a private cloud—or as a completely separate company—if it is a public cloud. While Clusters are used to group Physical Resources according to common characteristics such as networking topology or physical location, Virtual Data Centers (VDCs) allow the cloud administrator to create “logical” pools of Physical Resources (which can belong to different Clusters and Zones) and allocate them to specific user Groups, so enabling their consumption only by users in those Groups (see Figure 2).

Different authorization scenarios can be enabled with the powerful and configurable ACL system provided by OpenNebula, from the definition of Group Admins to the privileges of those users that can deploy Virtual Machines. Each Group can execute different types of workload profiles with different performance and security requirements.

The following are common enterprise use cases in large cloud computing deployments:

- **On-premise Private Clouds serving multiple projects, departments, units or organizations**. On-premise private clouds in large organizations require powerful and flexible mechanisms

to manage the access privileges to their virtual and physical infrastructure, and to dynamically allocate the available resources among different projects and departments. In these scenarios, the cloud administrator would define a VDC for each department, dynamically allocating resources according to their needs and delegating the internal administration of the Group to the department's IT administrator.
- **Cloud providers offering Virtual Private Clouds**. Cloud providers provide customers with a fully-configurable and isolated environment where they have full control and capacity to administer its users and resources. This combines a public cloud with the control usually seen in an enterprise private cloud system.

The Cloud will therefore have four different types of users:

- **Cloud Admins**: Role reserved for the company's IT staff with full admin privileges or to the company offering managed services.
- **Cloud Operators**: Users that operate and manage the Cloud Service.
- **Group Admins**: These users are allowed to manage virtual resources that belong to a specific Group, as well as its users. They are allowed to use physical resources associated with each of the VDCs the Group has access to, in a transparent way.
- **Customers / End-users**: Allowed to instantiate and manage VMs based on the predefined set-ups defined by Cloud Operators or Group Administrators.

11. Datacenter Federation

If administration domains need to be isolated, or the interconnection between datacenters does not allow a single controlling entity, OpenNebula can be configured in a federation. Each OpenNebula instance of the federation is called a Zone, one of them configured as primary and the others as secondaries. An OpenNebula federation is a tightly coupled integration, with all the instances sharing the same user accounts, groups and permission configurations (see Figure 3).

Federation allows end-users to consume resources allocated by the federation administrators regardless of their geographic location. The integration is seamless, meaning that a user logged into the Sunstone GUI of a Zone will not have to log out and enter the address of another Zone. Sunstone allows users to change the active Zone at any time, and it will automatically redirect the requests to the right OpenNebula instance.

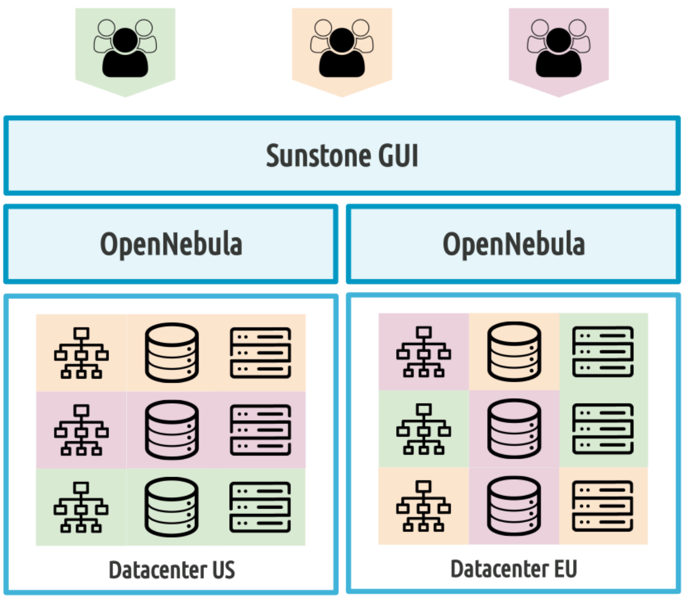

Figure 3. Federation Architecture.

12. True Edge, Hybrid, and Multi-Cloud

OpenNebula brings the provisioning tools and methods needed to dynamically grow on-demand a private cloud infrastructure with resources running on remote cloud and edge providers to enable powerful true hybrid and multi-cloud computing, and support all major clouds. This disaggregated cloud approach allows a seamless transition from centralized private clouds to distributed edge-like cloud environments. Companies are able to 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 the existing OpenNebula images and templates, keep complete control over the infrastructure and avoid vendor lock-in.

OpenNebula allows the deployment of a fully operational **Edge Cluster** in a remote provider and the management of its full life-cycle, starting with its provision and maintenance, until the unprovision. Each cloud or edge location (the "**provision**") is defined as a group of physical hosts allocated from the remote bare-metal or virtual provider. They are fully configured with the user-selected hypervisor and enabled in the cloud stack for the end-users.

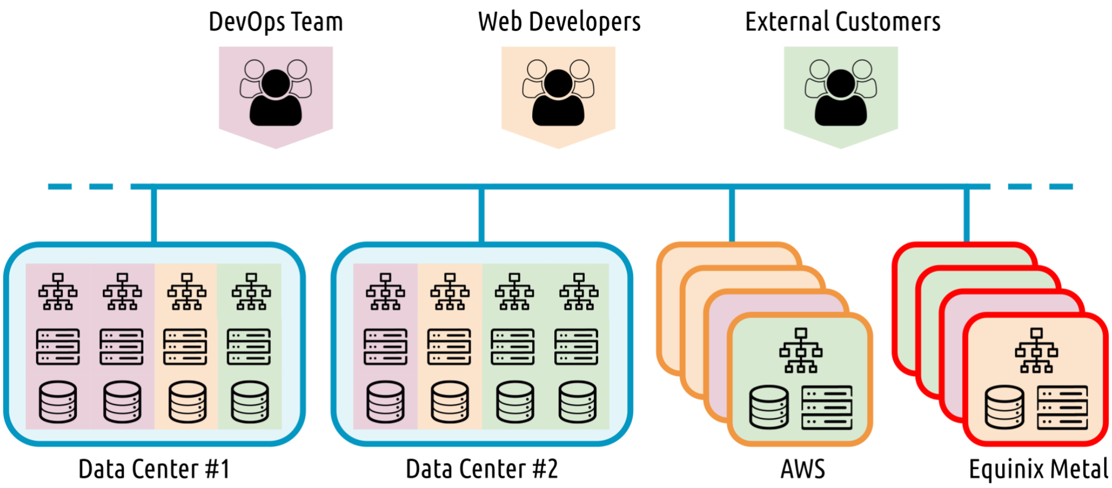

Figure 4. Hybrid Cloud architecture enabling cloud bursting.

13. High Availability

OpenNebula uses a distributed consensus protocol, based on Raft, to provide fault-tolerance and state consistency across OpenNebula services.

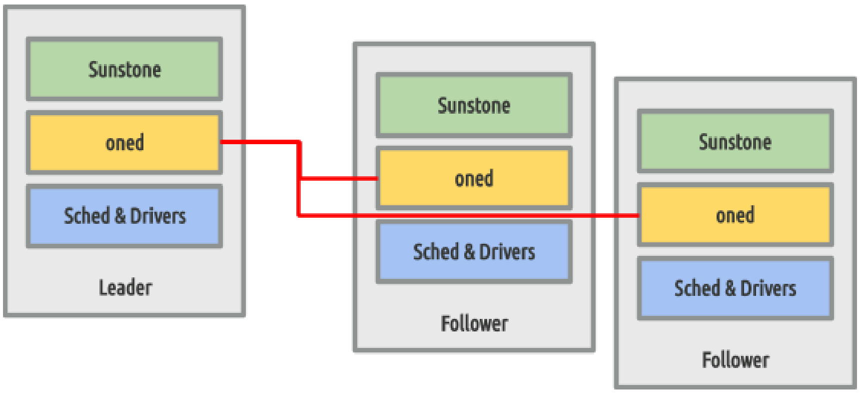

Figure 5. Overview of the HA architecture and main components.

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 keep its leadership. If a leader fails to send the heartbeat, followers are promoted to candidates and start a new election. Read-only operations can be performed through any OpenNebula server (oned) in the cluster; this means that reads can be arbitrarily stale but generally within the round-trip time of the network. A minimum of three Front-ends needs to be deployed in order to support one node failure. HA can also be configured for VMs (i.e. to be re-launched if they enter a fail state) or it can be configured for virtualization nodes, so all the VMs running in a crashed node get moved automatically to another node.

14. Ready for a Test Drive?

You can evaluate OpenNebula and build a cloud in just a few minutes by using **miniONE**,[10] our deployment tool for quickly installing an OpenNebula Front-end inside a Virtual Machine or a physical host, which you can then use to easily add remote Edge Clusters based on KVM, LXC or Firecracker.

15. Conclusions

The reference architecture described in this document has been created from the collective information and experiences from hundreds of users and cloud client engagements to help in the design and deployment of open cloud infrastructures. This document recommends software products and configurations for a smooth OpenNebula installation. However, in many cases, there are other aspects to be taken into account, like infrastructure platforms and pre-existing services in the data center, as well as specific provisioning processes within the company. In these scenarios, OpenNebula can be easily adapted to fit your data center and corporate policies. Contact us —we look forward to helping you at any stage of cloud computing journey.

[10] https://minione.opennebula.io

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

MANAGED SERVICES

Our team of experts can fully manage and administer your OpenNebula cloud for you

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_20210428