Open Cloud Reference Architecture

Version 2.4 – June 2024

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 is based on the collective information and experiences of hundreds of users and client engagements. Besides the main logical components and interrelationships within the architecture, 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. It provides recommendations for the main architectural components, including support for **KVM-based Virtual Machines**. Each section provides information about other open source infrastructure platforms that have been **tested and certified** by OpenNebula Systems to work in enterprise environments. As a complement for these certified components, the browseable OpenNebula **add-on catalog** offers further options supported by partners and by the OpenNebula Community.

This reference architecture does not include other components of the open cloud ecosystem which are important to consider when designing a cloud, such as configuration management and automation tools for managing cloud infrastructure or large numbers of devices.

Contents

Glossary

AD	Active Directory
ACL	Access Control List
COW	Copy on Write
DB	Database
DC	Data Center
HA	High Availability
NFS	Network File System
NIC	Network Interface Card
VDC	Virtual Data Center
VM	Virtual Machine
KVM	Kernel-based 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. Updated versions of OpenNebula are released regularly, 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** that contains the Front-end node(s), and
- The **Cloud Infrastructure**, which comprises one or several workload Clusters with the hypervisor nodes and the storage system. Clusters may reside at multiple geographical locations, all interconnected with multiple networks for internal storage and node management, and for private and public guest communications.

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

- **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 Edge Cloud Architecture.[2]
- **Customized Clusters:** typically deployed on-premises to meet specific requirements. You can find more details about this model below.

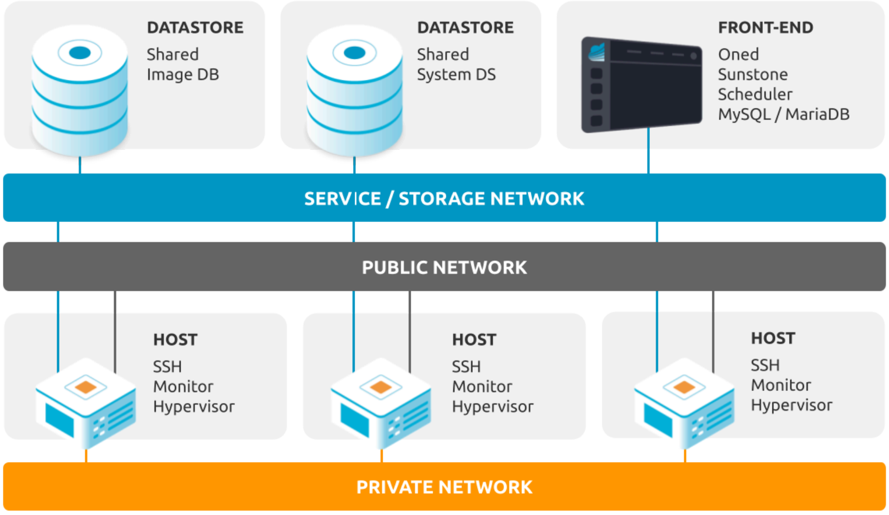

Figure 1. Reference Architecture, a bird's eye view.

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

Figure 1 shows a 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; they are connected to the hypervisors through the Service Network. The Front-end uses this network to monitor the status of the hypervisors and KVM VMs , as well as to initiate VMs, or for storage-related operations.

Virtualization hypervisors are also connected to the storage backend of the cloud through the Storage Network. Since OpenNebula requires low network traffic to operate, this network may be the same as the dedicated Service 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. It is possible to restrict access to each virtual network for specific users or groups, or to limit access using 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 requirements for performance, scalability and availability. 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	
Networking	VLAN 802.1Q	VXLAN
Storage	Local 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.

Both implementations, 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, in some cases VXLAN may apply to setups with few nodes, or 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 a 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 bear in mind that these recommendations are meant only as a guide.

Front-end minimum specifications are described in Table 2

Memory	16 GB
CPU	8 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 separate 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, specially 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) as well as execution resources (e.g. CPU, memory, or network access). The recommended Operating Systems for virtualization nodes are Ubuntu and CentOS/RHEL, using the KVM hypervisor.

All nodes have the same configuration in terms of 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 while 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, i.e. the characteristics of the VMs that will run in the cloud. You will 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, OpenNebula can manage 500 KVM Virtual Machines.

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.

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, NFS, iSCSI, SAN cabinets exported by Fiber Channel (FC), local storage managed by SSH, and various file systems or block devices such as LVM or VMFS. This wide-ranging support for storage allows 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. 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 local storage. The local storage driver has been developed for the efficient management of disk images in OpenNebula cloud environments. It offers recovery and migration enterprise features while retaining low-maintenance requirements and the benefits of lower cost. The local storage driver combines a 3-tier global architecture for application image distribution, with an enhanced SSH transfer mode that includes replica caching, backup, and snapshotting mechanisms to greatly improve scalability, performance and reliability. Application images are based on files with the 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.

The local storage driver has been implemented with lightweight technology components that already exist in the Linux OS. It can accommodate 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 10 Gbps networks. Moreover, its deployment follows a hyperconverged approach that does not require dedicated servers for implementing a distributed storage system. This approach reduces the complexity of the solution, enabling the use of the local storage area of the Cluster’s hosts.

Advanced Architecture

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

A Ceph cluster provides high availability, making the data accessible even if one (or indeed more, depending on your replica policies) of the storage nodes is down. At least three servers are needed; these should be dedicated exclusively to the Ceph cluster.

6. Networking

To ensure reliability of the cloud infrastructure, networking needs to be carefully considered and designed. Below you will find two recommended configurations (depending on the size of the cloud) for Basic and

Advanced OpenNebula clouds. Both 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. To sustain these networks, the recommendation is to use at least 10 Gbps switches that support VLAN trunking, using different VLANs for different private networks.

On the Front-end, only one interface connected to the Service Network is needed. OpenNebula will use this interface to connect over the SSH protocol to the shared filesystem server and the virtualization nodes. It is possible to isolate different private virtual networks by using 802.1Q VLAN tagging.

Private Network	For 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 use 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

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. Using 10gbe switches is advised to sustain the Storage, Private, Public, and Service networks.

Private Network	For 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. VXLAN encapsulates Ethernet frames within UDP packets, and thus solves the 4096 VLAN limitation problem. For using VXLAN, the multicast address must be enabled on the Service Network. It is worth noting that a current limitation in the Linux kernel allows handling of a maximum of 20 different VXLAN IDs on the same hypervisor.

⚠ OpenNebula includes support for additional network technologies, to comply with requirements of more stringent security and enhanced flexibility. Available options include, for instance, ebtables for Layer 2 isolation, using 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

To enable OpenNebula to pass configuration and network details to the running VMs, Virtual Machine images must contain the OpenNebula contextualization packages. These packages allow sharing information and configuration data between the OpenNebula interface and the guest operating system running on a VM—for example, scripts that the VM will 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):[3]

- AlmaLinux >= 8, 9
- CentOS >= 7, 8
- Red Hat Enterprise Linux >= 7, 8, 9
- Debian >= 10, 11, 12
- Ubuntu >= 18.04, 20.04, 22.04, 24.04
- Windows >= 7, 10, 11
- Windows Server >= 2008R2, 2012R2, 2016, 2019, 2022

8. Marketplace

OpenNebula offers access to its own Marketplace, an online catalog that allows users to easily download and import virtual appliances ready to run on OpenNebula clouds. The Marketplace allows access to the OpenNebula public repository as well as to private repositories. Images in OpenNebula's public repository have been tested and certified by OpenNebula.

Users can download images into a datastore, to be used by existing VM templates or instances. Images are in the qcow2 format, prepared to run on KVM hypervisors. They include VMs, VM templates and OneFlow Services, composed of multiple VMs associated with images.

The Marketplace includes two types of private repositories:

- HTTP Marketplace, where images are accessible through an 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. Users can define a regular interval to save VM disks that have been marked for backup, along with the VM metadata as known by OpenNebula, to the selected private marketplace. A restore tool is also available to recover VMs.

9. Authentication

Either the native OpenNebula subsystem or an LDAP/Active Directory (AD) server can be used for authentication. 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, thus centralizing authentication. When a user logs in to OpenNebula using LDAP credentials, OpenNebula automatically creates the user in the OpenNebula database, without the need to manually create the user. Groups of users will be created as needed, and access to resources will be assigned through the definition

[3] https://github.com/OpenNebula/one-apps/releases

of Virtual Data Centers (VDCs).

OpenNebula natively supports several authentication mechanisms, such as 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, including the Group admin, see only the virtual resources and not the underlying physical infrastructure. 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.

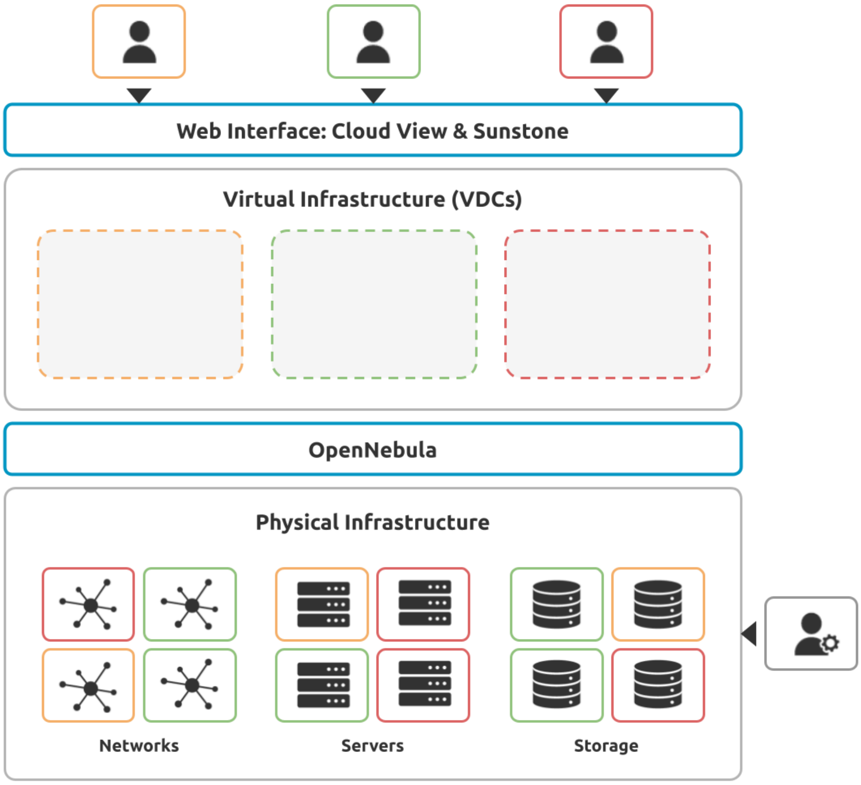
Web Interface: Cloud View & Sunstone
Virtual Infrastructure (VDCs)
OpenNebula
Physical Infrastructure
Networks
Servers
Storage

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).

The powerful and configurable Access Control List (ACL) system provided by OpenNebula allows administrators to enable different authorization scenarios, defining everything from Group Admins themselves to the privileges of users authorized to 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-premises Private Clouds serving multiple projects, departments, units or organizations**. On-premises private clouds in large organizations require powerful and flexible mechanisms to manage access privileges to their virtual and physical infrastructure, and to dynamically allocate 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 customers have full control and capacity to administer its users and resources. This use case 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 according to the access configurations previously defined by Cloud Operators or Group Administrators.

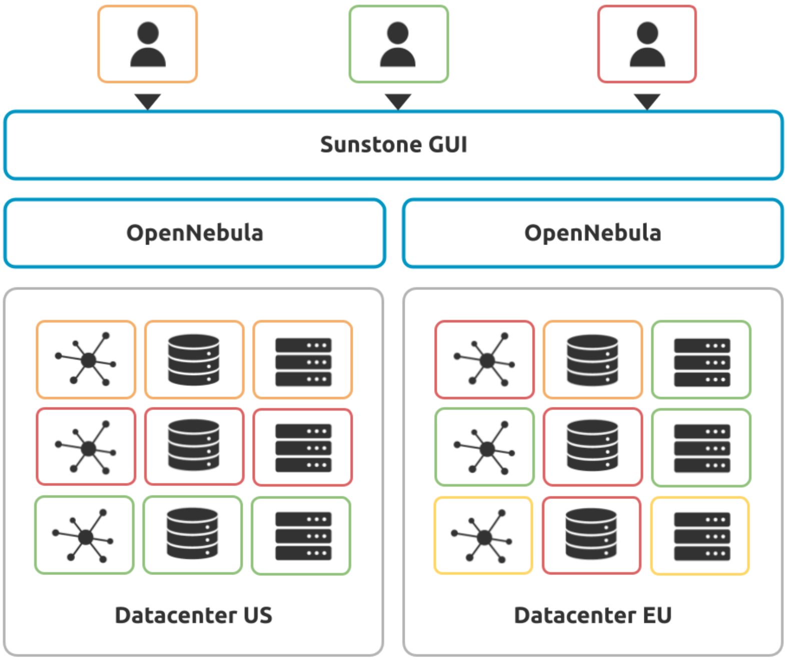

Figure 3. Federation Architecture.

11. Data Center Federation

If administration domains need to be isolated, or if the interconnection between data centers makes it unfeasible to control the cloud from a single entity, OpenNebula can be configured in a federation. Each OpenNebula instance in the federation is called a Zone. One of them is configured as the primary, the others as secondaries. An OpenNebula federation is a tightly coupled integration in which all instances share the same user accounts, groups and permission configurations (Figure 3).

Federation allows end-users to consume resources allocated by the federation administrators regardless of their geographic location. 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 will automatically redirect requests to the right OpenNebula instance.

12. True Edge, Hybrid, and Multi-Cloud

OpenNebula brings the provisioning tools and methods needed to dynamically grow a private cloud infrastructure that includes resources running on remote cloud and edge providers, enabling powerful, true hybrid and multi-cloud computing with support for all major clouds. 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 the existing OpenNebula images and templates, keep complete control over their infrastructure, and avoid vendor lock-in.

OpenNebula allows you to deploy a fully operational Edge Cluster in a remote provider, and to manage its full life cycle from provisioning and maintenance to unprovisioning. 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, available for end-users.

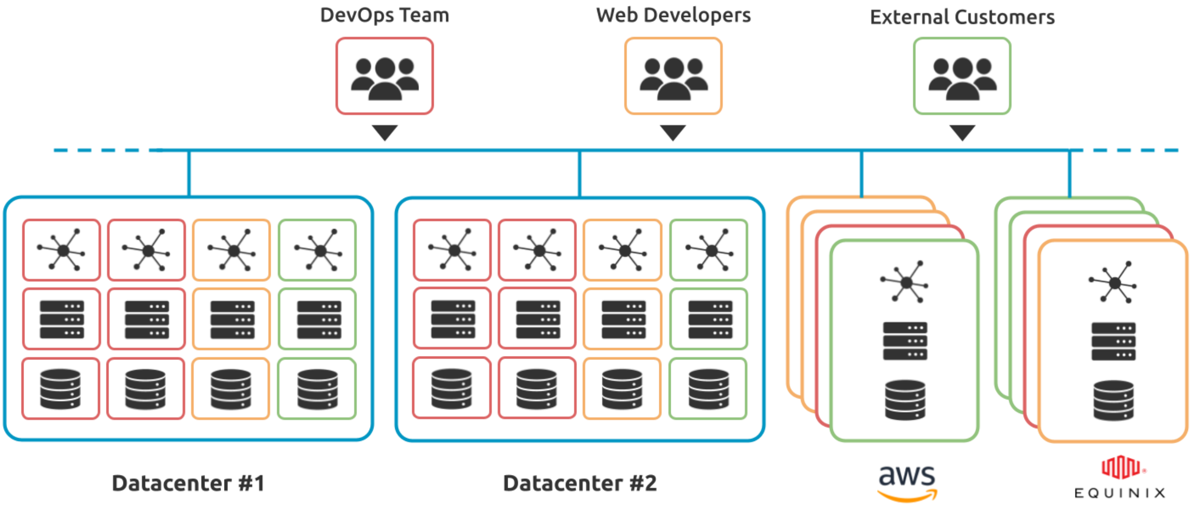

Figure 4. Hybrid Cloud architecture enabling cloud bursting.

13. High Availability

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

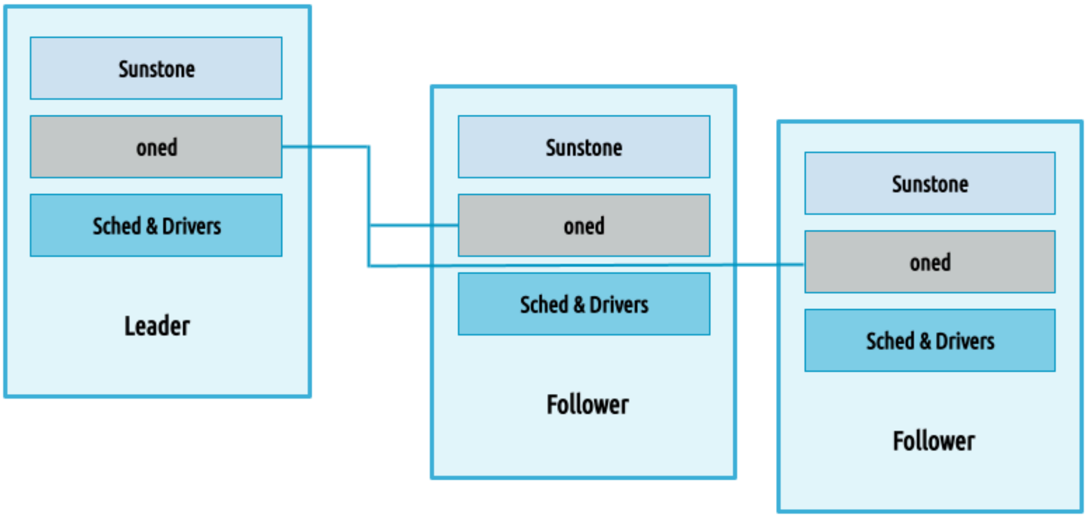

Figure 5. Overview of the HA architecture and main components.

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 retain 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 relaunch them if they enter a fail state) or for virtualization nodes, to ensure all VMs running in a crashed node are automatically moved to another node.

14. Monitoring and Alerts

Users of OpenNebula Enterprise Edition can benefit from OpenNebula's integration with Prometheus and Grafana to gather and visualize metrics, and to automatically generate alerts based on the overall cloud or on individual VMs running on KVM.

OpenNebula's comprehensive integration with Prometheus greatly simplifies gathering metrics and configuring alerts based on sets of rules. An OpenNebula exporter is used to provide information about the OpenNebula cloud itself, and the libvirt exporter to gather and provide metrics about VMs. Sample rule files—which provide the rule sets to trigger alerts—are generated based on metrics gathered by OpenNebula. Finally, the integration also includes Grafana dashboard templates designed for visualizing metrics on OpenNebula infrastructure and VMs; these templates can be easily imported and customized.

To install the integration and configure alerts, simply download and install the software package, and install Grafana which is also very simple to install. To configure monitoring and alerts, log into Grafana and use the Dashboard to add Prometheus as a new data source. You can then configure alerts by defining trigger conditions.

15. Backup and Restore

OpenNebula features native support for backup and restore. Backups are implemented through datastore and image abstraction, which enables backing up configurations such as access control policies or quotas.

The backup integration offers a choice of two backends: restic and rsync. rsync is a widely-used open source utility for file transfer and synchronization, included by default in most Linux distributions. restic, also open source, offers features such as encryption, multi-level compression or deduplication.

OpenNebula supports full and incremental (for KVM and qcow2 images). Backups for large numbers of VMs can be scheduled and managed through Backup Jobs. Cloud administrators can create a backup job definition that includes the target VMs, backup parameters, schedule and priority.

Backups may be defined, performed and managed using the OpenNebula API, CLI and the Sunstone GUI.

16. Ready for a Test Drive?

You can evaluate OpenNebula and build a cloud in just a few minutes by using **miniONE**,[4] 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.

miniONE

17. Conclusions

The reference architecture described in this document was 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 other aspects need to be considered, such as 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 your cloud computing journey.

[4] 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

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

`Rev2.4_20240601`