



OpenNebula on VMware: Cloud Reference Architecture

Version 1.6 – February 2020

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 public and private clouds based on OpenNebula on top of VMware vCenter. The reference architecture in this paper is intended for organizations with **existing VMware environments** or expertise who want to limit changes to their underlying VMware infrastructure, but see benefits in a common provisioning layer via OpenNebula to control compute workloads and/or want to take a step toward **liberating their stack** from vendor lock-in.

This document has been created from the collective information and experiences from hundreds of users and cloud client engagements. Besides 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, cloud bursting for workload outsourcing, and federation of geographically dispersed data centers.

Contents

1. What is OpenNebula?
2. OpenNebula on VMware vCenter
3. High Level Reference Architecture
4. OpenNebula Front-End
5. VMware vCenter Server
 - 5.1 VMware ESX Hypervisors
 - 5.2 Storage
 - 5.3 Networking
 - 5.4 Virtual Machines & Guest Support
6. MarketPlace
7. Authentication
8. Provisioning Model
9. Multi-Datacenter Deployments
10. Cloud Bursting
11. High Availability
12. Ready for a Test Drive?
13. Conclusions

Glossary

AD	Active Directory
DB	Database
DC	Datacenter
HA	High Availability
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 simple, feature-rich and flexible solution to build and manage enterprise clouds and virtualized DCs** that 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 a single **fully open source product** with a healthy and active community that 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. OpenNebula on VMware vCenter

OpenNebula is intended for companies willing to create a self-service cloud environment on top of their VMware infrastructure without having to abandon their investment in VMware and retool the entire stack. In these environments, OpenNebula seamlessly integrates with existing vCenter infrastructures to leverage advanced features—such as vMotion, HA or DRS scheduling—provided by the VMware vSphere product family. OpenNebula exposes a multi-tenant, cloud-like provisioning layer on top of vCenter, including features like Virtual Data Centers (VDCs), data center federation, or hybrid cloud computing connecting in-house vCenter infrastructure with public clouds. Resources like VM, VM templates, datastores and networks can be easily imported from vCenter infrastructures to OpenNebula.

OpenNebula also makes it possible to take steps towards liberating your stack from vendor lock-in. Once you have built your cloud with OpenNebula on VMware, you can then add new resources based on open source hypervisors—like KVM—and hence use OpenNebula as a migration framework to the open cloud.

3. High Level Reference Architecture

The VMware vCenter drivers enable OpenNebula to access one or more vCenter servers, which in turn manage one or more ESX clusters. Each ESX cluster is presented in OpenNebula as an aggregated hypervisor. Note that OpenNebula scheduling decisions are made at ESX cluster level, vCenter then uses the DRS component to select a ESX host and Datastore to deploy the Virtual Machine.

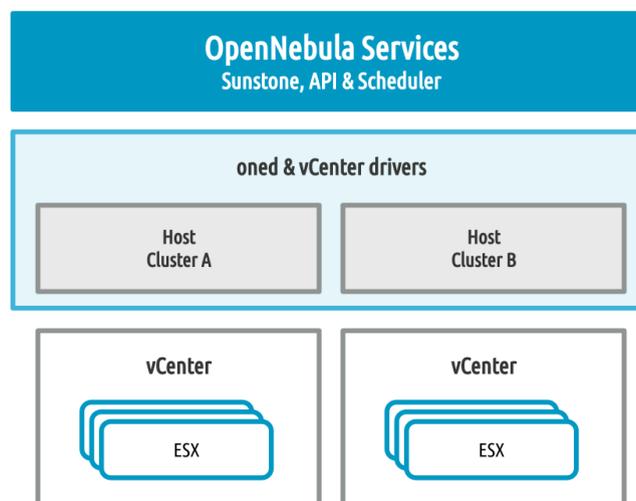


Figure 1. Reference Architecture, a bird's eye view.

¹ <http://opennebula.io/discover/>

As shown in Figure 1, OpenNebula components see two aggregated hosts, each one representing a vCenter cluster. You can further group these hosts into OpenNebula clusters to build Virtual Data Centers (VDCs) for your user groups in OpenNebula. Key elements of the reference architecture can be seen in Table 1.

Operating System	The reference operating systems for deployments of OpenNebula on VMware are CentOS/RHEL and Ubuntu
Cloud Management Platform	vCenter support is available in OpenNebula since version 4.12
VMware Cluster	vCenter is used to manage VMware vSphere environments with at least one cluster aggregating at least one ESX host
NSX	NSX-t and NSX-v support is available in OpenNebula since version 5.10

Table 1. Key elements of the Reference Architecture.

A summary of the implementation of this reference architecture can be consulted in Table 2:

Front-end	Supported OS (Ubuntu or CentOS/RHEL) Specific OpenNebula packages installed
Hypervisor	VMware vSphere (managed through vCenter)
Networking	Standard and Distributed Switches (managed through vCenter), NSX networks (managed through NSX manager)
Storage	Local and Networked (FC, iSCSI, SAS, NFS)—managed through vCenter
Authentication	Native authentication or Active Directory

Table 2. Summary of the implementation.

4. 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 3. Please take into account that these recommendations are meant only as a guidance.

When running on a front-end with the minimums described in Table 3 (more resources can be managed with more front-end specs), OpenNebula is able to manage a vCenter infrastructure with:

- Up to 4 vCenters clusters (managed by the same or different vCenter instances)
- Up to 40 ESXs in each vCenter cluster
- Up to 1.000 VMs in total

Memory	8 GB
CPU	2 CPU (4 cores)
Disk size	200 GB
Network	2 NICs

Table 3. Front-end hardware recommendations.

The front-end provides the following services:

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

5. VMware vCenter Server

5.1 VMware ESX Hypervisors

A key task when defining a cloud infrastructure is to correctly dimension the virtualization nodes (i.e. the ESX hosts) according to the expected workload. Memory wise, the recommendation is having 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. For instance, to run 80 VMs with 2GB of RAM and 1 vCPU each, the cloud would need 2 servers with 128GB each (96GB will be free to be distributed among these 80 VMs in any way seem fit) and 80 execution threads which can be delivered by two processors with 10 cores each and two execution threads per core.

Resource pools defined in vCenter are supported by OpenNebula so available memory and CPU can be partitioned. When launching a VM from OpenNebula, a resource pool can be selected automatically or the user can choose one.

5.2 Storage

In terms of storage management, OpenNebula interacts as a consumer of vCenter storage, and as such, supports all the storage devices supported by ESX and storage resources like Datastores and VMDKs.²

OpenNebula supports uploading, cloning and deleting VMDK files. When a VM is instantiated from a VM template, the associated datastore can be chosen. If DRS is enabled, then vCenter will pick the optimal datastore to deploy the VM. Datastores in a Storage DRS can be used as individual datastores by OpenNebula. New disks can be hot-plugged and OpenNebula can be informed no to erase the VM disks if a shutdown or cancel operation is applied to a VM, so users will not lose data accidentally.

Datastore migration is also supported. An instance can be relocated in another datastore using cold or live migration. Virtual Machines with non persistent images (changes are lost upon VM termination) and persistent images (changes are saved to the image upon VM termination) are both supported.

² <http://www.vmware.com/resources/compatibility/search.php?action=base&deviceCategory=san>

Storage Pods—a composite of vSphere datastores, a datastore cluster—are supported in OpenNebula and can be represented as a single datastore. Whenever a VM is deployed on a particular Storage Pod, SDRS is consulted to find the best datastore candidate.

5.3 Networking

Networking in OpenNebula is handled by creating or importing Virtual Network representations of vCenter Networks and Distributed vSwitches. In this way, new VMs with defined network interfaces will be bound by OpenNebula to these Networks and/or Distributed vSwitches. OpenNebula can create a new logical layer of vCenter Networks and Distributed vSwitches. In particular three types of Address Ranges can be defined per Virtual Network representing the vCenter network resources: plain Ethernet, IPv4 and IPv6. This networking information can be passed to the VMs through the contextualization process. A bandwidth limit per VM network interface can be applied.

OpenNebula supports NSX-T and NSX-V logical switches through the NSX Manager API. Some of the key points of this integration are:

- NSX Manager is automatically detected from vCenter Server.
- Transport Zones are automatically discovered.
- Logical switches are created in the NSX Manager after a Virtual Network is created in OpenNebula.
- Link VM NICs to these logical switches through the OpenNebula GUI, both at deployment and run times.
- Import or remove existing logical switches.

5.4 Virtual Machines & Guest Support

OpenNebula creates new Virtual Machines by instructing VMware vCenter to deploy specific vCenter VM templates. These VM templates contain the full definition of the final Virtual Machine (e.g. virtual disks or network interfaces) and have been pre-configured leveraging the functionality provided by vCenter.

The pre-configured VM templates can be fine-tuned at deployment time (e.g. by adding new network interfaces or changing capacity) through the provision portal. OpenNebula supports this tailoring functionality of the VM guest Operating System through two mechanisms:

- **OpenNebula Contextualization** allows 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 (an exhaustive list is available in the official documentation):³
 - CentOS >= 6
 - Red Hat Enterprise Linux >= 7
 - Debian >= 8
 - Ubuntu >= 14.04
 - Windows >= 7
 - Windows Server >= 2008
- **vCenter Customization Specifications** ties the template with a vCenter Customization Specification so it is configured on VM creation. This method can be used in all the guest OSs supported by vCenter.

³ <https://github.com/OpenNebula/addon-context-linux>

Existing VMs can be imported and managed through OpenNebula. This functionality is available for vCenter VMs and is included for all supported public clouds (see Section 10), enabling the fast import of the existing workload—both in the local and public infrastructure—and its management through OpenNebula.

A VM deployed from OpenNebula can be configured in a way that whenever the VM life-cycle ends, OpenNebula will instruct vCenter to create a new vCenter VM template out of the VM including the settings for any new disk or network interface attached to it.

OpenNebula can change the vCenter cluster where the VMs are running through vCenter vMotion. When VMs are migrated to another cluster, its compute resources (CPU & RAM) will run on a new ESX belonging to the target cluster.



Besides VMware vCenter, OpenNebula comes with native support to KVM-based VMs and LXD system containers. Hence, OpenNebula can be used to achieve a smooth migration from VMware infrastructures to a vendor-independent alternative based on fully open source virtualization technologies.

6. MarketPlace

OpenNebula has access to its own MarketPlace, which enables the user 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 in a vCenter Server and used to add them to existing VM templates or instances.

OpenNebula public marketplace features images in qcow2 format. In this way images can be used by KVM hypervisors, LXD system containers and vCenter Servers. For vCenter there is an automatic conversion process to VMDK when importing an image in qcow2 format.

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.

7. Authentication

Either the native OpenNebula subsystem or a 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 for instance SSH keys and X509 credentials.

8. Provisioning Model

The Provisioning Model in OpenNebula is based on Virtual Data Centers (VDCs). A VDC is a fully-isolated virtual infrastructure where a Group of users (or several Groups of users), under the control of a Group administrator, can create and manage compute and storage capacity. Users in a Group, including the administrator, would only be able to see virtual resources, not the underlying physical infrastructure. Physical resources allocated to a Group are managed by the cloud administrator through a VDC. Resources in a VDC can be dedicated in full to a Group, thus providing isolation at the physical level too.

Users are organized into Groups (similar to what in other environments are called Projects, Domains or Tenants). 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.

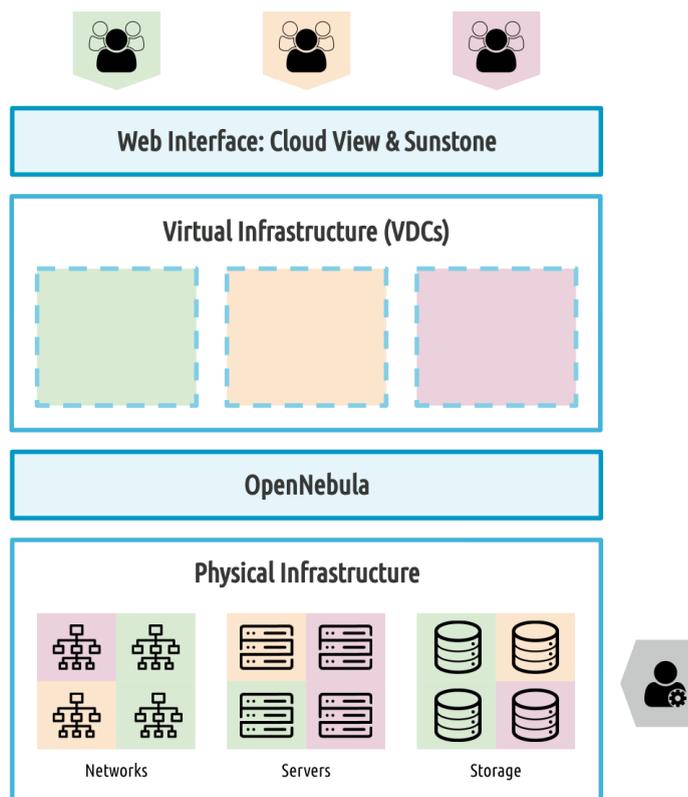


Figure 2. Resource Provisioning Model in OpenNebula.

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 three different types of users:

- **Cloud Admins:** Role reserved to the company's IT staff will full admin privileges.
- **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 both the Cloud and Group Administrators.

9. Multi-Datacenter Deployments

OpenNebula interacts with the vCenter instances by interfacing with its SOAP API. This API allows remote communication so enabling architectures where the OpenNebula instance and the vCenter environment are remotely located. A single OpenNebula instance can orchestrate remotely several vCenter instances located in different data centers. Connectivity between data centers needs to have low latency in order to have a reliable management of the remote vCenter instances from OpenNebula (see Figure 3).

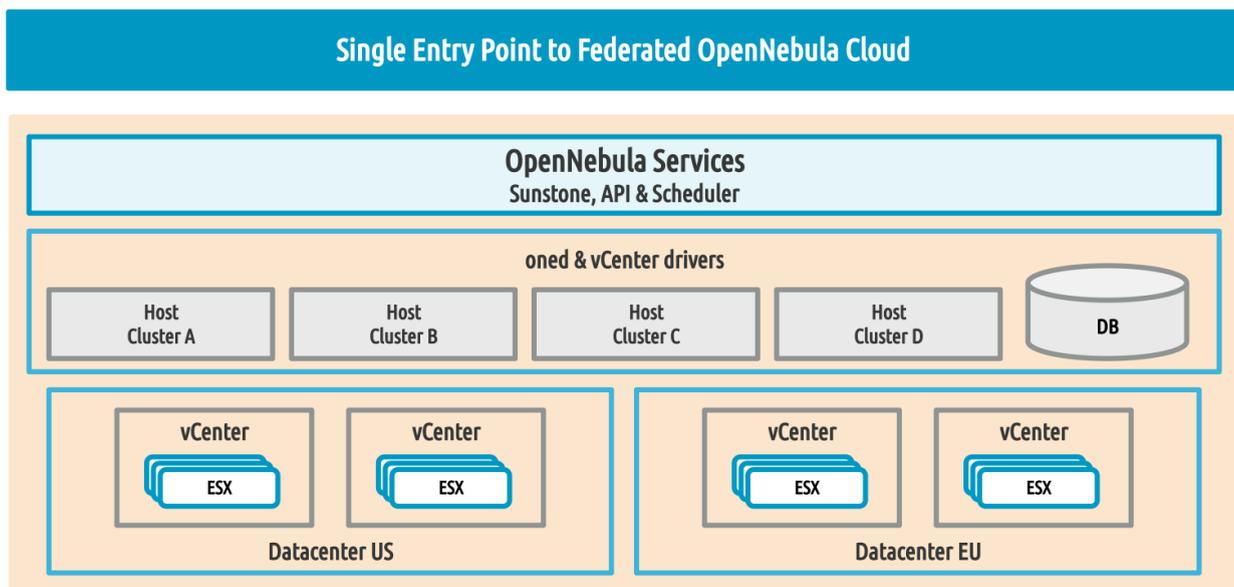


Figure 3. Single OpenNebula multi-datacenter deployment.

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 4).

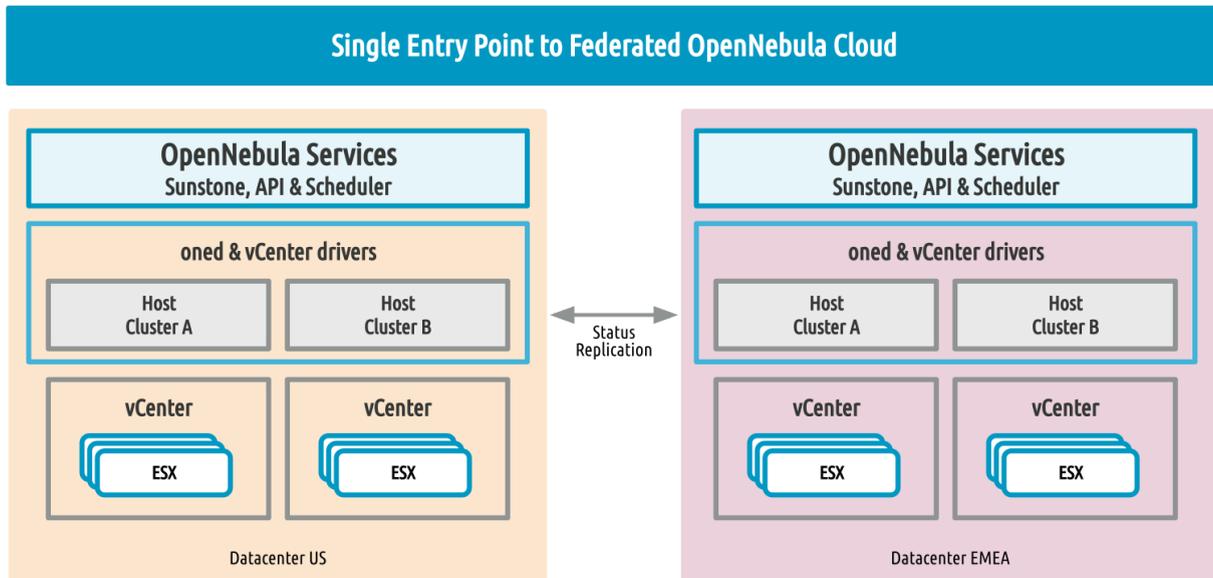


Figure 4. Multi-datacenter deployment based on two OpenNebula zones.

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.

10. Cloud Bursting

Cloud bursting is the model in which local resources of a private cloud are combined with resources from a remote cloud provider hence creating a hybrid cloud. The remote provider could be a commercial cloud service, such as Amazon Web Services (AWS), Microsoft Azure, or even some other OpenNebula cloud.

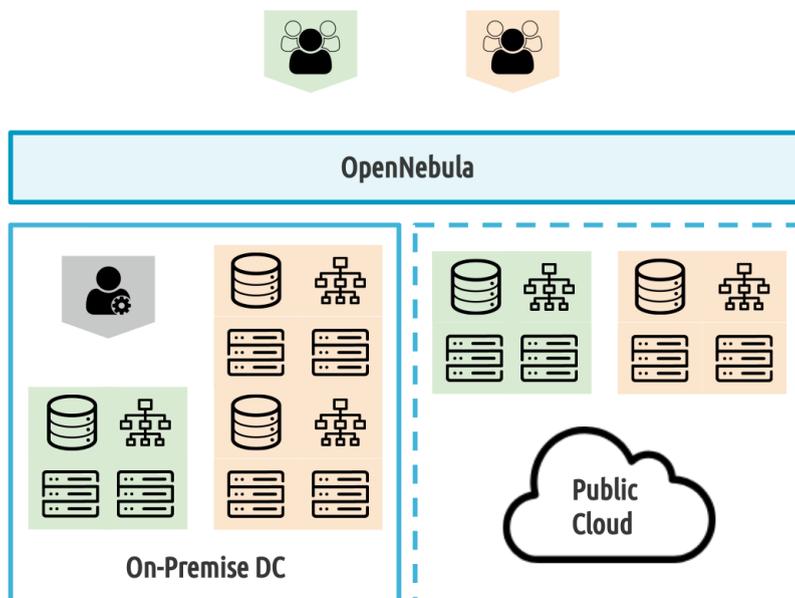


Figure 5: Hybrid Cloud architecture enabling cloud bursting.

The architecture for hybrid clouds is described in Figure 5. The support that OpenNebula provides for cloud bursting enables highly scalable hosting environments. In fact, OpenNebula’s approach to cloud bursting is unique. The reason for this uniqueness is the transparency to both end users and cloud administrators to use and maintain the cloud bursting functionality. Transparency to cloud administrators comes from the fact that a public cloud region is modelled as any other host (albeit of potentially a much bigger capacity), so the scheduler can place VMs at any cloud region.

11. High Availability

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

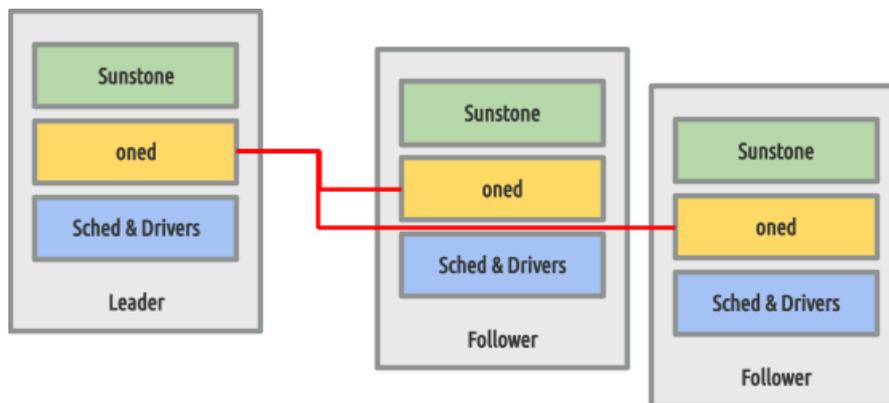


Figure 6: Overview of the HA architecture and main components.

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 promote 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. To ensure a proper functioning of the HA cluster, a minimum of three front-end needs to be deployed, to support one node failure. This number can be increased provided the target number is odd.

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 virtualization node.

12. Ready for a Test Drive?

You can evaluate OpenNebula on your vCenter environment and build a cloud in only 5 minutes by downloading vOneCloud,⁴ a virtual appliance that contains all required OpenNebula services within a single CentOS Linux appliance.

vOneCloud

⁴ <http://vOneCloud.today>

13. 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 cloud infrastructures based on OpenNebula and VMware. 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 into your data center and corporate policies. Contact us, we look forward to helping you at any stage of cloud computing adoption.

LET US HELP YOU DESIGN, BUILD AND OPERATE YOUR CLOUD



CONSULTING & TRAINING

Help you design, integrate, build and operate an OpenNebula private or hybrid cloud infrastructure



SUPPORT SUBSCRIPTION

From OpenNebula developers with product influence and privacy guarantee at a competitive cost



JUMPSTART PACKAGES

Help you springboard your productivity, speed time to deployment, and reduce risks

Sign up for updates at OpenNebula.io/getupdated

© OpenNebula Systems 2020. This document is not a contractual agreement between any person, company, vendor, or interested party, and OpenNebula Systems. This document is provided for informational purposes only and the information contained herein is subject to change without notice. OpenNebula is a trademark in the European Union and in the United States. All other trademarks are property of their respective owners. All other company and product names and logos may be the subject of intellectual property rights reserved by third parties.

Rev1.6_20200229