

OpenNebula on VMware: Cloud Reference Architecture

Version 1.0, November 2015

Abstract

The OpenNebula Cloud Reference Architecture is a blueprint to guide IT architects, consultants, 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 want to take a step toward liberating their stack from vendor lock-in.

The 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 reference documents software products, configurations, and requirements of infrastructure platforms recommended for a smooth OpenNebula installation. Three optional functionalities complete the 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. Authentication
7. Provisioning Model: Virtual Data Centers
8. Multi-Datacenter Deployments
9. Hybrid Clouds: Bursting
10. High Availability
11. Ready for a Test Drive?
12. Conclusions

Glossary

VM	Virtual Machine
DC	Datacenter
NIC	Network Interface Card
AD	Active Directory
DB	Database
VDC	Virtual Datacenter
HA	High Availability

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.** OpenNebula combines existing virtualization technologies with advanced features for multi-tenancy, automatic provision and elasticity. OpenNebula follows a bottom-up approach driven by sysadmins, devops and users real needs.

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 page¹.

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, data center federation or hybrid cloud computing to connect in-house vCenter infrastructures with public clouds.

OpenNebula also makes possible to take step towards liberating your stack from vendor lock-in. Once you have built your cloud with OpenNebula on VMware you can add new resources based on open-source hypervisors, like KVM and Xen, 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 that manages one or more ESX Clusters. Each ESX Cluster is presented in OpenNebula as an aggregated hypervisor. Note that OpenNebula scheduling decisions are therefore made at ESX Cluster level, vCenter then uses the DRS component to select the actual ESX host and Datastore to deploy the Virtual Machine.

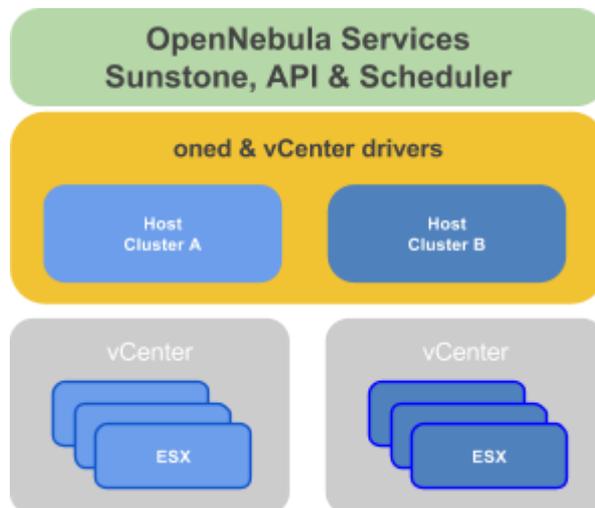


Figure 1. Reference Architecture, a bird's eye view

¹ <http://openebula.org/about/key-features/>

As shown in Figure 1, OpenNebula components see two aggregated hypervisors (OpenNebula hosts) where each represents a cluster in a vCenter. You can further group these hosts into OpenNebula clusters to build complex virtual data centers for your user groups in OpenNebula.

Key elements of the reference architecture can be seen in Table 1.

	Description
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 4.12 Cotton Candy
VMware Cluster	vCenter is used to manage VMware vSphere environments with at least one cluster aggregating at least one ESX host

Table 1. Key Reference Architecture Elements

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

	Description
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)
Storage	Local and Networked (FC, iSCSI, SAS) (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 compute server or a VM- devoted to orchestrate 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 as a guidance.

Regarding the dimensioning of the cloud, when running on a front-end with the minimums described in Table 3, OpenNebula is able to manage a vCenter infrastructure of the following characteristics:

- Up to 4 vCenters
- Up to 40 ESXs managed by each vCenter
- Up to 1.000 VMs in total, each vCenter managing up to 250 VMs

	Recommendation
Memory	2 GB
CPU	1 CPU (2 cores)
Disk size	100 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: OneFlow, OneGate, econe

5. VMware vCenter Server

5.1 VMware ESX Hypervisors

One of the most important tasks defining a cloud infrastructure is to dimension the virtualization nodes (ie, 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.

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 ESX². When a VM is instantiated from a VM Template, the datastore associated with the VM template is chosen. If DRS is enabled, then vCenter will pick the optimal Datastore to deploy the VM.

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 these 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.

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

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

network interfaces) and have been pre-configured leveraging the functionality provided by vCenter.

The pre-configured VM Templates can be fine tuned (e.g. by adding new network interfaces or changing capacity) by the users through the provision portal at deployment time. 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 OS of the VM. The following guestOS are supported for OpenNebula Contextualization:
 - CentOS/RedHat 6 and 7
 - Debian 7 and 8
 - Ubuntu from 12.04 to 15.05
 - Windows 2008 R2, 2012, 2012 R2
- **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.

Existing VMs or VMs not created through OpenNebula are referred to as Wild VMs, and can be imported to be managed through OpenNebula. This functionality is present for vCenter VMs and is included for all supported public clouds (see Section 9), enabling the fast import of the existing workload both in the local and public infrastructure to be managed through OpenNebula.



Besides VMware vCenter, OpenNebula supports natively XEN and KVM. Hence, OpenNebula can be used to achieve a smooth migration from VMware based infrastructures to another infrastructure based on open source hypervisors.

6. Authentication

Depending on the corporation, a native OpenNebula authentication subsystem or an Active Directory/LDAP server can be used. In both cases, the OpenNebula cloud will be accessible to users by both the CLI and the Sunstone web interface. In the native OpenNebula authentication subsystem, users will be kept in the OpenNebula database as native ones and the authentication is done using user/password. Groups will be generated as needed for the different users that will use the system.

Alternatively, users can be authenticated against a corporate Active Directory (AD)/LDAP server, which has to be accessible through the service network. Users are created in the OpenNebula user DB table after the first use. Groups of users will be created as needed for the different users that will use the system, and assign them access to resources through Virtual Datacenters (VDCs).



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

7. Provisioning Model: Virtual Data Centers

The provisioning 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, would only see the virtual resources allocated to them and not the underlying physical infrastructure.

VDCs effectively allow to create 'logical' pools of isolated physical resources and allocate them to user Groups. Users can then provision virtual machines through a simple cloud-like interface, completely abstracted from both the VMware infrastructure and the logical allocation performed by the cloud

administrators, see Figure 3.

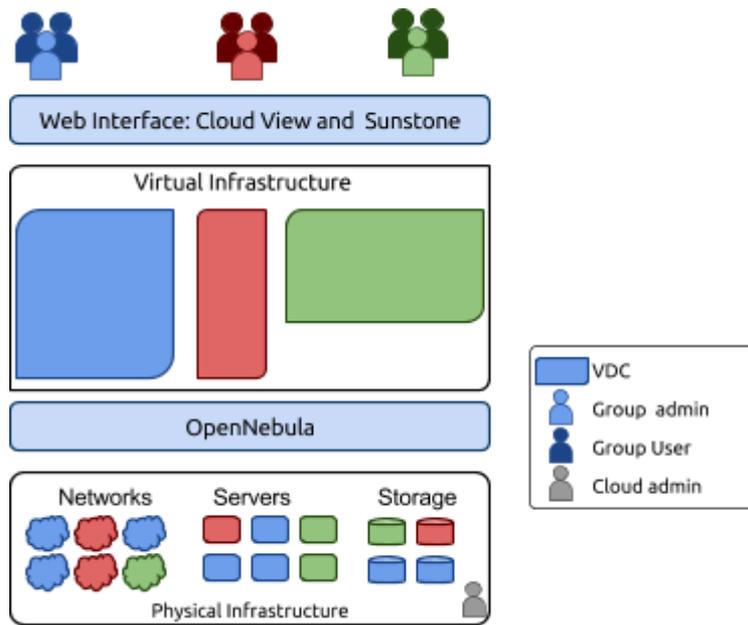


Figure 3: Resource Provisioning Model in OpenNebula

8. 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 instances can orchestrate several vCenter instances remotely located in different data centers. Connectivity between data centers needs to have low latency in order to have a reliable management of vCenter from OpenNebula, see Figure 4.

When 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 master and the others as slaves. An OpenNebula federation is a tightly coupled integration, all the instances will share the same user accounts, groups, and permissions configuration, see Figure 5.

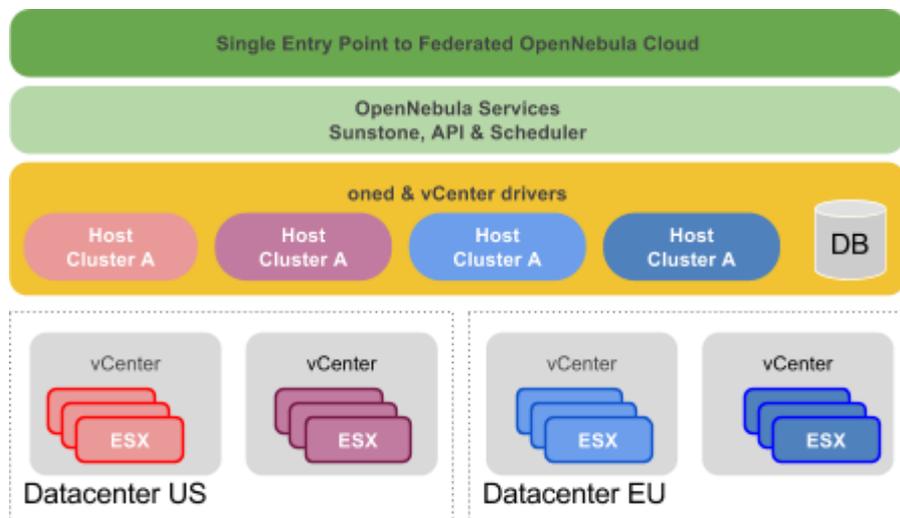


Figure 4. Single OpenNebula multi-datacenter deployment

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 web interface of a Zone will not have to log out and enter the address of another Zone. Sunstone allows to change the active Zone at any time, and it will automatically redirect the requests to the right OpenNebula at the target Zone.

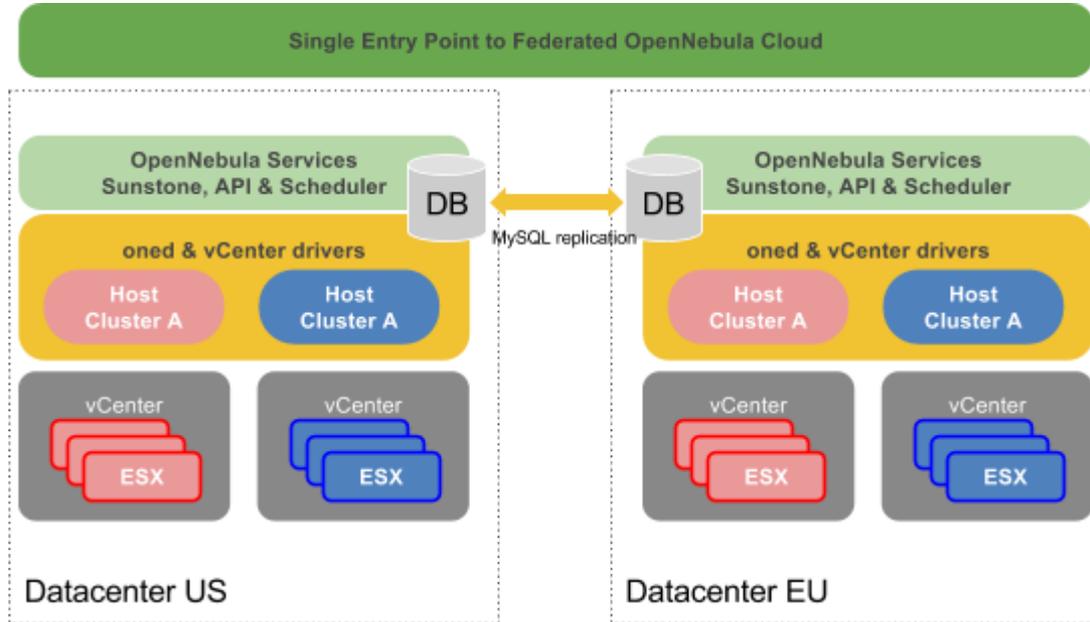


Figure 5. Multi-datacenter deployment based on two OpenNebula zones

9. Hybrid Clouds: Bursting

Cloud bursting is the model in which local resources of a private cloud are combined with resources from remote cloud providers hence creating a hybrid cloud. The remote provider could be a commercial cloud service, OpenNebula supports Amazon EC2, IBM SoftLayer and MS Azure. This support for cloud bursting enables highly scalable hosting environments. The architecture for hybrid clouds is described in Figure 6.

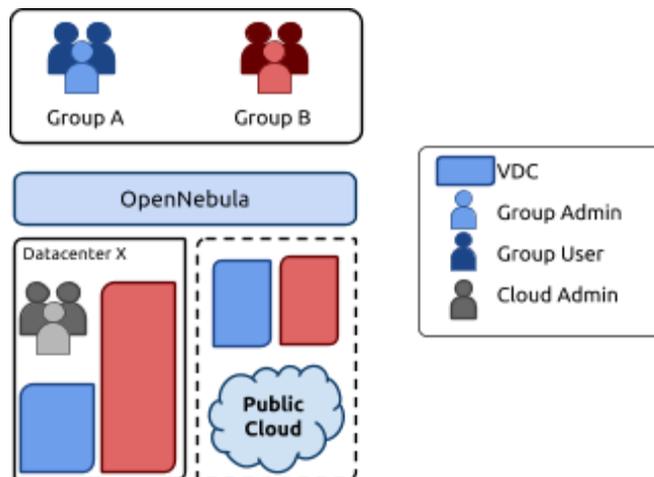


Figure 6: Hybrid Cloud architecture enabling cloud bursting

OpenNebula approach to cloud bursting is unique. The reason behind 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 an AWS EC2 region is modelled as any other host (albeit of potentially a much bigger capacity), so the scheduler can place VMs on EC2 as well as to import VMs running on EC2 that were not launched through OpenNebula. On the other hand, the transparency to end users is offered through the hybrid template functionality: the same VM template in OpenNebula can describe the VM if it is deployed locally and also if it gets deployed in Amazon EC2. With this in place, users just have to instantiate the template, and OpenNebula will transparently choose if that is executed locally or remotely.

10. High Availability

The OpenNebula controller can be deployed in a high-availability configuration (active-passive) using a shadow VM deployed in a separated host to achieve high availability (HA). In this configuration, two physical servers are used as front-ends, each of them running a VM with the OpenNebula services. Moreover, a MySQL database cluster will be created in the two frontend nodes used to run Frontend VMs. This database keeps all the data related to the objects that OpenNebula manager (hosts, networks, VMs, users, etc). The data is synchronized between the cluster nodes so in case of failure of one node all the data is still reachable.

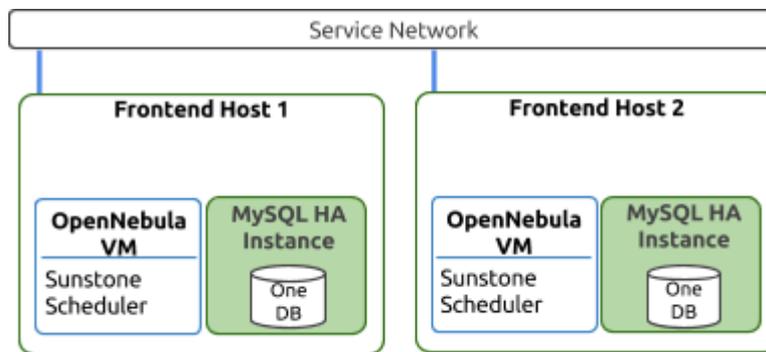


Figure 7: Overview of the HA architecture and main components

HA can also be configured for VMs, to be relaunched 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.

11. Ready for a Test Drive?

You can evaluate OpenNebula on your vCenter environment and build a cloud in only 5 minutes by downloading vOneCloud³, a virtual appliance that contains all required OpenNebula services within a single CentOS Linux appliance.



Building a Cloud on VMware vCenter? → Try the new vOneCloud app!

³ <http://vOneCloud.today>

12. 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 on VMware. The 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 services preexisting in the datacenter as well as the provisioning processes of the corporation. In these scenarios, OpenNebula can be easily adapted to fit into your data center and 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 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 business and technical risks

Sign up for updates at OpenNebula.org/getupdated

© OpenNebula Systems 2015. 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 a pending trademark 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.

Rev1.0_20151113