

Cloud: A different way of storage by Open Nebula

^{#1}Sarita Shankar Pol, ^{#2}Prof. Shyamrao V Gumaste

¹polsarita@gmail.com

²svgumaste@gmail.com

^{#1}Computer Engineering-SPPU, Pune, ^{#2}Computer Engineering-SPPU, Pune



ABSTRACT

Cloud Computing is always giving facility for user to access cloud services over internet when data is needed. This paper discusses about implementing a private cloud using open nebula, platform for private cloud. This private cloud is capable of providing facilities to institute for private use for infrastructure(IaaS) and platform as a service(PaaS).Infrastructure includes memory storage, servers, virtualization, compute and network services and platform as a service(PaaS) includes operating system, middle-ware and runtime environment. For the private cloud implementation in small institutes which refers to use the data at private security open nebula open source architecture is done.

Keywords— Internet, private cloud, open nebula, open stack

ARTICLE INFO

Article History

Received : 15th June 2015

Received in revised form :

17th June , 2015

Accepted : 21th June, 2015

Published online :

26th June 2015

I. INTRODUCTION

OpenNebula is the open source tool that is used for the cloud computing based for the creation of the private cloud. Its open, architecture, interfaces and components provide the flexibility and extensibility. These features also facilitate its integration with any product and service in the cloud and virtualization ecosystem, and management tool in the data centre. OpenNebula provides an abstraction layer independent from underlying services for security, virtualization, networking and storage, avoiding vendor lock-in and enabling interoperability. This open and flexible approach for cloud management ensures widest possible market and user acceptability, and simplifies adaptation to different environments. OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. Most advanced and innovative enterprise for management of virtualized centers to build private and hybrid clouds. OpenNebula is not a feature or performance limited edition of an Enterprise version, OpenNebula is truly open-source code, not open core, distributed under Apache license.

OpenNebula is not a feature or performance limited edition of an Enterprise version, OpenNebula is truly open-source code, not open core, distributed under Apache license.

II. LITERATURE SURVEY

Vinay Khedekar[6] introduces the cloud computing technology for making the private cloud for college and its basic concepts for installation. The popularity of the private cloud for college like virtual lab as well as deployment of private cloud is explained. The paper describes the open nebula open source software's to deploy private clouds. A setup for open nebula open source software is including the basic installation steps to configure the cloud with the help of open nebula. Cloud implementation, programming language, database compatibility, OS compatibility of open nebula is given further.

Girish L S[1] propose a process framework for supporting of the cloud computing and it is proposed that PaaS clouds would have long term commitments because of unavailability of alternative solutions. To meet the quality of private cloud for virtual lab effectively as well as efficiently open nebula is used to implement the cloud platform of the

open nebula.

III. PROPOSED SYSTEM ARCHITECTURE

Architecture of open nebula:

During the 2nd and 3rd quarter of 2012 from the OpenNebula open source cloud management platform community with regards to the type of cloud deployment and its main architectural components. The aim is to acquire information in order to improve the support for the most demanded infrastructure platforms and deployments. OpenNebula has been downloaded more than 80,000 times from the project site. However, the number of downloads is much higher because the code can be also downloaded from the software repository and from the official repository of the main Linux distributions. Moreover, several commercial and open-source distributions use OpenNebula as an embedded cloud management platform.

Architectural Overview:

OpenNebula assumes that your physical infrastructure adopts a classical cluster-like architecture with a front-end, and a set of hosts where Virtual Machines (VM) will be executed. There is at least one physical network joining all the hosts with the front-end.

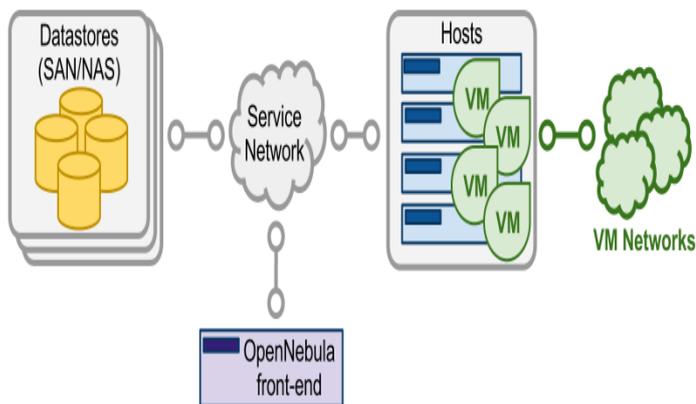


Fig 1: Architectural Overview

The basic components of an OpenNebula system are:

- **Front-end** that executes the OpenNebula services.
- Hypervisor-enabled **hosts** that provide the resources needed by the VMs.
- **Datastores** that hold the base images of the VMs.
- Physical **networks** used to support basic services such as interconnection of the storage servers and OpenNebula control operations, and VLANs for the VMs.

OpenNebula presents a highly modular architecture that offers broad support for commodity and enterprise-grade hypervisor, monitoring, storage, networking and user management services. This guide briefly describes the different choices that you can make for the management of the different subsystems. If your specific services are not

supported we recommend to check the drivers available in the Add-on Catalog. We also provide information and support about how to develop new drivers.

Virtualization Hosts: The hosts are the physical machines that will run the VMs. There are several certified platforms to act as nodes for each version of OpenNebula. The Virtualization Subsystem is the component in charge of talking with the hypervisor installed in the hosts and taking the actions needed for each step in the VM lifecycle.

OpenNebula natively supports three hypervisors:

- Xen
- KVM
- VMware

IV. IMPLEMENTATION

An OpenNebula is one of the easiest cloud platform that the systems to install and configure the private cloud to a small enterprise to make the installation of the private cloud to the extension. Figure 1 shows the layout of the OpenNebula deployment

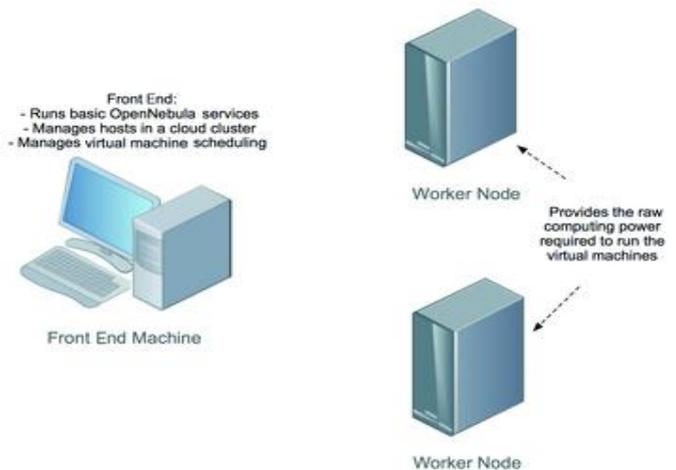


Fig 2: The OpenNebula deployment

An OpenNebula deployment is modeled after the classic cluster architecture of the cloud deployment of the server. A single gateway, sometimes also called the master node of the cloud can be performed with the help of the open nebula front end, queues jobs scheduling, and submits jobs to the machines in cluster to perform next action. The other machines (other than server node) in the cluster, known as "worker nodes", provide the computing environment to the cluster node, jobs submitted to the

cluster. The worker nodes in an OpenNebula cluster are machines that deploy a virtualization hypervisor, such as VMware. The master node is also known as the front end machine and it runs on the several

OpenNebula services like VMware. If you migrate from a private cloud to a hybrid cloud, the front end will enable you to send virtual machines to Amazon EC2.

Installation

Installation of open nebula server:

Step 1:

One machine will serve as the front end, called cloud server (nebula cloud server). Two machines will be acting as a worker nodes deploying KVM. The worker nodes are open-nebula 1 and open nebula 2.

To install the front end, you need to install a single package:

```
sudo apt-get install opennebula
```

This package installs the daemon oned that manages all the type of OpenNebula services. The installation also creates a new user "oneadmin," which OpenNebula retains for its own use. The installation of this package also produces a new pair of SSH RSA keys. OpenNebula uses SSH to communicate with other machines in a cluster of the open nebula, so a public key is used to identify the front end to recognise the user to authenticate to give the facility to make a cluster.

Step 2:

After the opennebula package has been installed, the next step is to the nodes of the open nebula, add worker nodes is the further work. Before you add worker nodes, you need to install the worker node OpenNebula package on the worker node machines. At both worker nodes, need to install the opennebula (cluster) node package :

```
sudo apt-get install opennebula-node
```

This package installs the necessary worker node packages and creates the oneadmin user.

Step 3:

Worker Nodes

One host command need to be installed. The syntax of the command is: onehost [options] command [parameters]

The hostname argument is the hostname or the IP of the machine, the next three parameters relate to the hypervisor and the transfer manager service. OpenNebula uses different drivers to access different hypervisors like VMware, XEN. If anyone wants to add the the cluster to the open nebula server like want ot add a host deploying KVM:

```
onehost add open-nebula-wn im_kvm vmm_kvm tm_ssh
```

The final arguments relate to the way virtual machines will be transferred to the node. If anyone were using NFS, you would specify tm_nfs instead of tm_ssh. sudo -u oneadmin ssh opennebula1 and verify the host's authenticity. On opennebula1, run:

```
sudo apt-get install opennebula-node
```

```
sudo tee /var/lib/one/.ssh/authorized_keys << EOT
```

onehost add open nebula 2

```
sudo -u oneadmin ssh opennebula1
```

and verify the host's authenticity. On opennebula2, run

```
sudo apt-get install opennebula-node
```

```
sudo tee /var/lib/one/.ssh/authorized_keys << EOT
```

The onehost create command will prompt for the front end's key to the authorized key file, which OpenNebula uses internally. So, you have to go back to the worker nodes and add the SSH key to the worker nodes. If everything goes well, your can check the status of the nodes with the one host command.

V. CONCLUSION

In this way we have seen the methods for configuring private cloud using the open nebula, which is freely available for the private cloud. With the help of open nebula, open source platform one can easily form the hybrid cloud and private cloud.

VI.ACKNOWLEDGEMENT

It gives me great pleasure in expressing thanks and gratitude to my guide Prof. Shyamrao V. Gumaste, Head of the Computer Engineering Department, for his valuable guidance and encouragement throughout the Project work. I am heartily thankful for his time to time suggestion and the clarity of the concepts of the topic that helped me a lot during this work. Throughout the work there was great help and support given by my husband Prof. Sachin S Nehe, HOD, Mechanical Dept, SRCOE. Also I am giving gratitude to my all my family members.

REFERENCES

- [1] d"Building Private Cloud using Open Stack", Girish L S, Dr. H S Guruprasad., *IJETTCS*, Volume 3, Issue 3, May – June 2014
- [2] SonaliYadav, "Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, Openstack and Opennebula", *International Journal Of Engineering And Science*, Vol.3, Issue 10 (October 2013), pp 51-54, ISSN (e): 2278-4721, ISSN (p):2319-6483.
- [3] Muhammad AfeefChauhan, Muhammad Ali Babar, "Migrating Service-Oriented System to Cloud Computing: An Experience Report", 4thInternational Conference on Cloud Computing, Washington, DC,
- [4] Wei Hao, I-Ling Yen, BhavaniThuraisingham, "Dynamic Service and Data Migration in the Clouds", 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA,20-24 July 2009, pp 134-139,Print ISBN: 978-0- 7695-3726-9, DOI: 10.1109/COMPSAC.2009.127.
- [5] Scott Tilley,TauhidaParveen, "Migrating Software Testing to the Cloud", IEEE International Conference on Software Maintenance (ICSM), Timisoara, 12-18

Sept. 2010, pp 1,Print ISBN: 978-1-4244-8630-4,
DOI:10.1109/ICSM.2010.5610422.

- [6] Study of Cloud Setup for College Campus,
International Journal of Advanced Research ,
Computer Science and Software Engineering, Vol2,
Issue 10, October 2012