



Volume: 2, Issue: 7, 148-151
July 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Divya Shree R
M. Tech scholar, Department
of computer science,
Visvesvaraya Institute of
Advanced Technology,
Muddenahalli, Karnataka

Decentralized Access Management with Anonymous Authentication of data stored on OpenStack and Hyper-V cloud

Divya Shree R

Abstract

Recently Cloud Computing has evolved into a big buzz word and building your own private cloud can be interesting. This paper demonstrates how a private cloud such as OpenStack can be implemented using Ubuntu OS. OpenStack is an open source platform which provides IAAS and PAAS services. Infrastructure includes the storage, servers, virtualization, compute and network services and platform as a service includes the operating system, middleware and runtime environment. Hyper-V is a native hypervisor developed by Microsoft. Decentralized Access management is the software application that will be run on either of the cloud servers. This software can be any of the software developed to be run on cloud.

Keywords: OpenStack, private cloud, nova, neutron, cinder, keystone, glance, horizon, Hyper-V.

1. Introduction

Cloud Computing allows the users to access a shared pool of computing resources like networks, servers, storage, applications, and services that can be rapidly provisioned based on the customer requirements. Essential Characteristics of Cloud Computing are on-demand, broad network access, resource pooling, measured service and rapid elasticity etc. Cloud computing has five deployment models as private cloud, public cloud, hybrid cloud, virtual private cloud and community cloud. Cloud computing has three service models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). This paper proposes the system architecture for building a private cloud which is capable of providing IaaS and PaaS as a service over the internet. The proposed system is developed using OpenStack. OpenStack is open source software which the project developers and cloud computing technologist can use to setup and run the cloud. Its services can be accessed via APIs. The important components of OpenStack are Nova, Cinder, Keystone, Glance and Horizon.

Section II gives related work and proposed system architecture is explained in Section III. The implementation details of the proposed system are described in Section IV. The experiments and results of the proposed system are presented in Section V. Conclusion is given in Section VI.

2. Literature Survey

Sonali Yadav ^[1] introduces the cloud computing technology and its basic concepts. The popularity of the private cloud and its deployment is explained. The paper describes various open source software's to deploy private clouds such as Eucalyptus, Open Stack and Open Nebula. A comparative study is made about these three open sources of tware's based on their architectures. Cloud implementation, programming language, database compatibility, OS compatibility of these software's are mentioned. Muhammad *et al.* ^[2] propose a process framework for supporting migration to cloud computing and it is proposed that PaaS clouds would have long term commitments because of unavailability of alternative solutions. To meet the quality effectively and efficiently, cloud should support for defined SLA and this paper discusses Hackystat systems for migration. Wei Haoet. *et al.* ^[3] considered service and migration as a problem and proposed a frame work which facilitates service migration. They also designed a cost model and the decision algorithm to determine the trade-offs on service migration and selection. The paper also focuses on the placement of migrated services by considering the migration, maintenance and communication costs. Scott *et al.* ^[4] propose SMART decision frame work for migrating software testing frame work to the cloud and it is

Correspondence:
Divya Shree R
M. Tech scholar, Department
of computer science,
Visvesvaraya Institute of
Advanced Technology,
Muddenahalli, Karnataka

based on the SOA migration, Adoption and Reuse technique (SMART). This SMART helps organization to identify their current testing process and also describe the requirements of the target cloud computing environment for performing software testing. Bram *et al.* [5] presents a test method to determine the viability and value of deployment in the cloud. The essential characters of the cloud discussed are on-demand, self-service, broad network access, resource pooling, rapid elasticity and measured service.

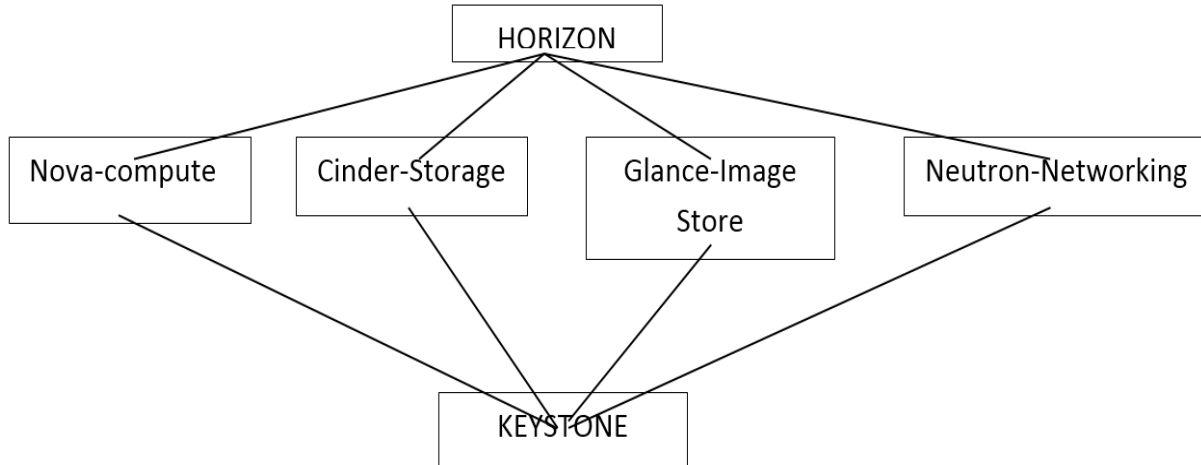


Fig 1: Components of OpenStack

1. **Nova:** Nova is the Computing Fabric controller for the OpenStack Cloud. The necessary activities for the life cycle of instances within the OpenStack cloud are handled by Nova. This characteristic makes Nova a Management Platform to manage various compute resources, networking, authorization, and scalability needs of the OpenStack cloud.
2. **Glance:** Glance is a standalone service which provides a catalog service for storing and querying virtual disk images. Nova and Glance together provides an end-to-end solution for cloud disk image management.
3. **Cinder:** Cinder is a Block Storage service for OpenStack. It's designed to allow the use of either a reference implementation (LVM) to present storage resources to end users that can be consumed by the OpenStack Compute Project (Nova). It virtualizes pools of block storage devices and provides end users with a self-service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device.
4. **Keystone:** Keystone provides identity and access policy services for all components in the OpenStack family. All components of OpenStack including Cinder, Glance, and Nova are authenticated and authorized by Keystone.
5. **Neutron:** Neutron is an OpenStack project to provide "networking as a service" between interface devices (e.g., vNICs) managed by other Openstack services (e.g., nova). Neutron is a core and supported part of the OpenStack platform
6. **Horizon:** Horizon can be used to manage instances and images, create key pairs, attach volumes to instances, manipulate Swift containers etc.

3.2 Microsoft's Client Hyper-V

Hyper-V, codenamed Viridian and formerly known as Windows Server Virtualization, is a native hypervisor.

3. Proposed System Architecture

3.1 Openstack Cloud Server

The proposed system aims to build private cloud using open source software OpenStack. The system architecture of OpenStack is as depicted in Fig.1. The proposed system consists of various modules such as Horizon, Nova, Glance, Cinder and Keystone

Microsoft released the finalized version of Hyper-V with certain x86-64 editions of Windows Server 2008 on June 26, 2008. Microsoft provides Hyper-V through two channels:

1. Part of Windows: Hyper-V is an optional component of Windows Server 2008 and later. It is also available in x64 SKUs of Pro and Enterprise editions of Windows8 and Windows 8.1
2. Hyper-V Server: It is a freeware edition of Windows Server with limited functionality and Hyper-V component Client Hyper-V is the name for the virtualization technology that was introduced in Windows 8

Step 1: Prerequisites

1. Windows 8 Pro or Enterprise 64 bit Operating System
2. 64 bit processor with Second Level Address Translation (SLAT)
3. 4GB system RAM at minimum
4. BIOS-level Hardware Virtualization support.

Step 2: Setting Up Hyper-V

- Ensure that hardware virtualization support is turned on in the BIOS settings. Select and enable **Hyper-V** under **turn windows features on or off** and reboot the system.
- When you enable Hyper-V, during the reboot, it is installed underneath Windows, and the Windows Installation then becomes the parent partition/VM.
- It acts differently, as it can see the hardware and directly manage the hypervisor as such - think of it like a local, familiar GUI to the hypervisor and its drivers, configuration.

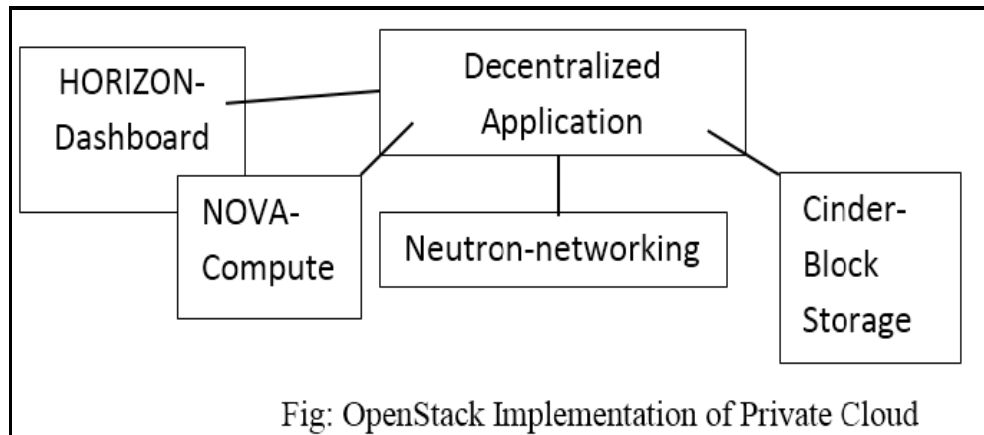
3.3 Decentralized Access Management with Anonymous Authentication of data

The new decentralized access management theme for secure knowledge storage in clouds that supports anonymous

authentication. Within the projected theme, the cloud verifies the credibility of the user while not knowing the user's identity before storing knowledge. The theme conjointly has another feature of access management during which solely valid user's area unit ready to decipher the hold on data. The theme prevents replay attacks and supports creation, modification, and reading knowledge hold on within the cloud. It tends to conjointly address user revocation. Moreover, the authentication and access management theme is decentralized and sturdy, in contrast to different access management schemes designed for clouds that are a unit centralized. The communication, computation, and storage overheads are compared to centralized approaches.

4. Implementation

The proposed system is implemented using open source software called Openstack and Ubuntu operating system. The three nodes such as Compute, Controller and Storage are installed with Ubuntu server operating system because all these nodes have to behave like servers as shown in Fig. 2. Compute node is installed with the Nova packages and services. Controller node is installed with the Glance, Keystone and Horizon packages and services. Storage node is installed with the Swift or cinder packages and the services. All three nodes are connected internally to OpenStack Dashboard with internal network. The Application which is ready to use the cloud service is connected through external network to controller node of the private cloud.



1. Module 1: Compute Node The installation of nova packages is carried out by downloading the nova packages by the following command:

- apt-get install -y nova-api nova-cert nova-conductor nova-consoleauth nova-novncproxy nova-scheduler python-novaclient nova-compute nova-console. These install lines added most of the packages that expected (nova-api, nova-compute, nova-network etc.) to work nova on the open stack.

2. Module 2: Glance Image Store The installation of glance packages is carried out by downloading the glance packages by the following command

- sudo apt-get install glance glance-api glance-client glance-common glance-registry python- glance

These install lines added most of the packages that expected (glance-api, nova-registry etc.) to work on the open stack.

3. Module 3: Keystone The installation of keystone packages is carried out by downloading the keystone packages by the following command.

- sudo apt-get install keystone python-keystone pythonkeystoneclient These install lines added most of the packages that expected (python-keystone, python-keystone etc.) to work keystone on the open stack.

4. Module 4: Neutron The installation of neutron packages is carried out by downloading the neutron packages by the following command.

- apt-get install -y neutron-server neutron-plugin-openvswitch neutron-plugin-openvswitch-agent neutron-common neutron-dhcp-agent neutron-l3-agent neutron-metadata-agent openvswitch-switch

5. Module 5: Horizon Dashboard The installation of horizon packages is carried out by downloading the horizon packages by the following command.

- sudo apt-get install openstack-dashboard
- These install lines added most of the packages that expected to work dashboard on the OpenStack.

5. Experiments and Results

At the end of implementation, the private cloud openstack must be able to access through the horizon dashboard and all its components must be able to work correctly. The application Decentralized Access Management must be able to run on the windows instance. Using Hyper-V, the application must be working properly on the created new instance of windows 8.1 virtual machine.

6. Applications

- The above implementation can be used for any applications which needs cloud server.
- The implementation can be extended by adding up the storage of more than one hard disk for more space requirements.
- You can create your own testing environment with a single machine
- Developers can also use Client Hyper-V to test their software on multiple operating systems.
- You can use Client Hyper-V to troubleshoot. You can export a virtual machine from your production environment, open it on your desktop with Client Hyper-V, perform your required troubleshooting, and then export it back into the production environment.
- You can use Hyper-V on a laptop for demonstrating older versions of Windows or non-Windows operating systems.

7. Conclusion

In this paper, a new and powerful infrastructure for building private cloud is introduced which is capable of serving the

user request for Infrastructure and Platform services. Services include launching different flavours of images, instances and services etc. The future work can be the implementation of Software as a Service with the help of OpenStack and Ubuntu. Microsoft's Hyper-V is the native hypervisor which can be used in any Windows 8 systems to use cloud. Any software application such as Decentralized Access Management with Anonymous Authentication can be run on any of the two above mentioned cloud servers.

8. References

1. 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.
2. Bram Rongen, "Making the case for migration of information systems to the cloud", 16th Student Conference on IT, Enschede, The Netherlands, Jan 27 2012, Copyright 2011, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
3. Samer Al-Kiswany, Dinesh Subhraveti, PrasenjitSarkar, MateiRipeanu, "VMFlock: Virtual Machine Co-Migration for the Cloud", 20th International Symposium on high performance distributed computing, New York USA, 2011, pp 159-170, Print ISBN: 978-1-4503-0552-5, DOI:10.1145/1996130.1996153.
4. Rashmi, Dr. Shabana, ehfuz, Dr.G.Sahoo, "A fivephased approach for the cloud migration", International Journal of Emerging Technology and Advanced Engineering, April 2012, ISSN 2250-2459.
5. Timothy Wood, PrashantShenoy, K.K. Ramakrishnan Jacobus, V an der Merwe, "CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines", 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments Newport Beach, California, USA, March 9-11 2011, pp121-132, Print ISBN: 978- 1-4503-0687-4, DOI:10.1145/1952682.1952699.
6. Johnson B, YanzhenQu, "A Holistic Model for Making Cloud Migration Decision", 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, Leganes, 10-13 July 2012, pp 435-441, Print ISBN: 978-1-4673-1631- 6, DOI: 10.1109/ISPA.2012.63.
7. Parastoo Mohagheghi, ThorSaether, Software Engineering Challenges for Migration to the Service Cloud Paradigm", 2011 IEEE World Congress on Services, Washington, DC, 4-9 July 2011, pp 507-514, Print ISBN: 978-1-4577-0879-4, DOI:10.1109/SERVICES.2011.26.
8. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, USA 2009.
9. M. Armbrust, et al., "A View of Cloud Computing," Communication of ACM, vol. 53, pp. 50-58, 2010.
10. Q. Zhang, et al., "Cloud computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, vol. 1, pp. 7-18, 2010 [19] Ken Pepple, "Deploying OpenStack" July 2011, First Edition, ISBN: 978-1-449-31105-6.
11. <http://docs.openstack.org/essex/openstackcompute/starter/content>.