ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

Review on OpenStack for Cloud Computing

Rajeswar P. V. N
Research scholar, Andhra University,
Visakhapatnam, AP, India

Anvesh P
Assistant Professor, Training Dept,
Chirala Engg College, Chirala., AP, India

A*bstract*--- **Now a days from Small, Medium industries to big enterprises, from students to computer professionals and to smart phone users everyone is bind with using cloud computing services directly or indirectly. Cloud Computing has quickly imposed itself on the IT scenery, rapidly flooding the market with new appealing services There are tremendous advantages with Cloud Computing in terms of service availability, scalability, use anywhere, metered use, etc. Moreover, cloud computing fulfilled a long-held dream of turning computing as a utility. Since 1960s [1] tremendous research has been carried out by various research practitioners in various fields majorly in business and academia; and also by developing many solutions to put cloud computing in implementation. As a result there are lot of cloud computing solutions these days provided by various agencies both commercial and non-commercial; Openstack is one from those communities of cloud enthusiastic. OpenStack is an open source cloud computing platform for public and private clouds. A series of interrelated projects deliver a cloud infrastructure solution. This paper analyses architecture and characteristics of an open source cloud computing platform Openstack being open source is of tremendous consideration.**

Keywords: CloudComputing, Openstack, Architectures, Open Source Cloud Computing

I INTRODUCTION

Cloud computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services on the Internet in a remotely accessible fashion. An infrastructure setup based on the cloud computing model is generally referred to as the "cloud". The following are the broad categories of services available on the cloud: [2]

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)
- Backup as a Service (BaaS)

There are many cloud computing offerings today. However, there are certain challenges in the adoption of any particular offering. The cloud offerings available in the market are for the most part, more proprietary and often are not interoperable. It has become cumbersome for customers to move from one cloud to another as it required data alteration or huge migration costs. So, many customers opt to stay with a provider that does not meet their needs, just to avoid tediousness or due to lack of knowledge about other cloud offerings - this concept has been termed "vendor lock-in".

With this limitation of proprietary cloud services it was the need of the hour to have open specifications and APIs for cloud offerings. Open Cloud Computing Interface (OCCI) emerged as standard to try and provide a solution for this by defining interoperable, portable and integration standards. OpenStack was launched as an independent implementation of OCCI and offers a flexible and accommodating cloud service. OpenStack has got more momentum considering big names are a part of its project (AT&T, IBM, HP, REDHAT, Cisco, Dell, etc).

The OpenStack cloud operating system[3] turns all the sets of hypervisors within a data centre or across multiple data centres into pools of resources. These pools of resources can be consumed and managed from a single place which is the dashboard of Openstack. Both the administrators and users can go to its dashboard to perform their tasks in easy and fast manner. One can create virtual machines, configure networks and manage volumes all from a single place.

II OPENSTACK

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter. It is all managed through a dashboard called Horizon, that gives administrators control while empowering their users to provision resources through a web interface.[4]

OpenStack is a global collaboration of developers and cloud computing technologists, producing the ubiquitous open source cloud computing platform for public and private clouds. The project aims to deliver solutions for all types of clouds by being

- Simple to implement
- Massively scalable
- Feature rich

Openstack was first introduced in June 2010, born with its initial code from NASA's Nebula platform and Rackspace's Cloud Files platform. Openstack mission according to **[5]** is "To produce the ubiquitous open source cloud computing platform that will meet the needs of public and private cloud providers regardless of size, by being simple to implement and massively scalable". Since its very first introduction in June 2010 openstack is accelerating and gaining popularity every day. And now openstack is joined by scores of developers and support team backend by some large houses like Canonical, Rackspace, etc. Currently openstack is joined by more than 180 business houses providing to support for openstack, more than 15996 people working over it and active involvement of people from more than 137 nations. Although openstack is portable software but many Linux distributions provide it as an operating system also like Ubuntu Canonical Openstack is written in python and all of the code for Openstack is freely available under the Apache 2.0 license.

Today whole of the openstack project is managed by Openstack Foundation established in September, 2012. Openstack Foundation is an independent body providing shared resources to help achieve openstack mission by empowering, protecting and promoting openstack software and the community around it including users, developers and the entire ecosystem. Openstack brought jointly by NASA and Rackspace as a new open source cloud initiative freely available under Apache License. Openstack controls large pools of compute and storage all managed either through APIs or dashboard. Dashboard provided by openstack is a web interface for provisioning to and releasing of resources from end users. Openstack delivers a scalable cloud's infrastructure as a service. Openstack community is having around six-month release cycle and till now openstack has released nine major openstack releases from Austin in October 2010 to Icehouse in April 2014 .

Some OpenStack users include: [6]

- PayPal / eBay
- NASA, CERN
- Yahoo!
- Rackspace Cloud
- HP Public Cloud
- MercadoLibre.com
- AT&T
- KT (formerly Korea Telecom)
- Deutsche Telekom
- Wikimedia Labs
- Hostalia of Telef nica Group
- SUSE Cloud solution
- Red Hat OpenShift PaaS solution
- Zadara Storage
- Mint Services
- GridCentric

III OPENSTACK ARCHITECTURE

Openstack is organized around three main modules i.e.

- Compute
- Storage &
- Networking.

Along with these three, dashboard become an important component in providing interface to administrators and users for provisioning and release of resources. These components and their interaction with user's application and underlying hardware over which other openstack services do run can be represented as shown in **figure 1**. Openstack compute is designed for provisioning of virtual machines providing scalable cloud computing platform. Openstack storage provides objects storage to be used for storing necessary images to run virtual machines or virtual instances. Openstack network provides necessary services which are used for communication with in virtual machine i.e. inter-VM and external to virtual machines. All these modules along with other services running underneath works in a close interaction with each other which may or may not be running on single server (test environment) or on a multiple servers (production environment) jointly fulfilling the common purpose of Openstack for providing a feature rich, scalable platform for infrastructure as a service cloud platform.

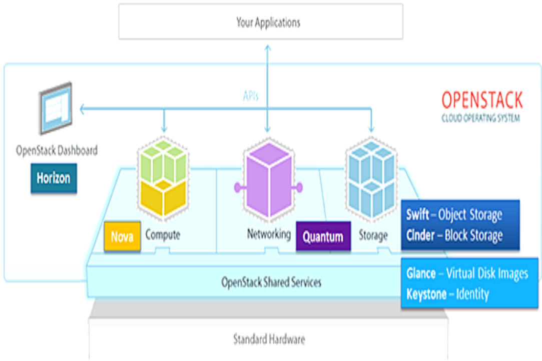

Fig 1 : Architecture for OpenStack with core components

IV OPENSTACK SERVICES/COMPONENTS

Figure 2 shows the main components of Openstack.

A storage-focused design might require the ability to use Orchestration to launch instances with Block Storage volumes to perform storage-intensive processing.

A storage-focused OpenStack design architecture typically uses the following components: [7]

A. OpenStack Identity (**keystone**)
B. OpenStack dashboard (**horizon**)
C. OpenStack Compute (**nova**) (including the use of multiple hypervisor drivers)
D. OpenStack Object Storage (**swift**) (or another object storage solution)
E. OpenStack Block Storage (**cinder**)
F. OpenStack Image service (**glance**)
G. OpenStack Networking (**neutron**) or legacy networking (nova-network)
H. OpenStack Telemetry Service (**Ceilometer**)
I. **Orchestration Heat**

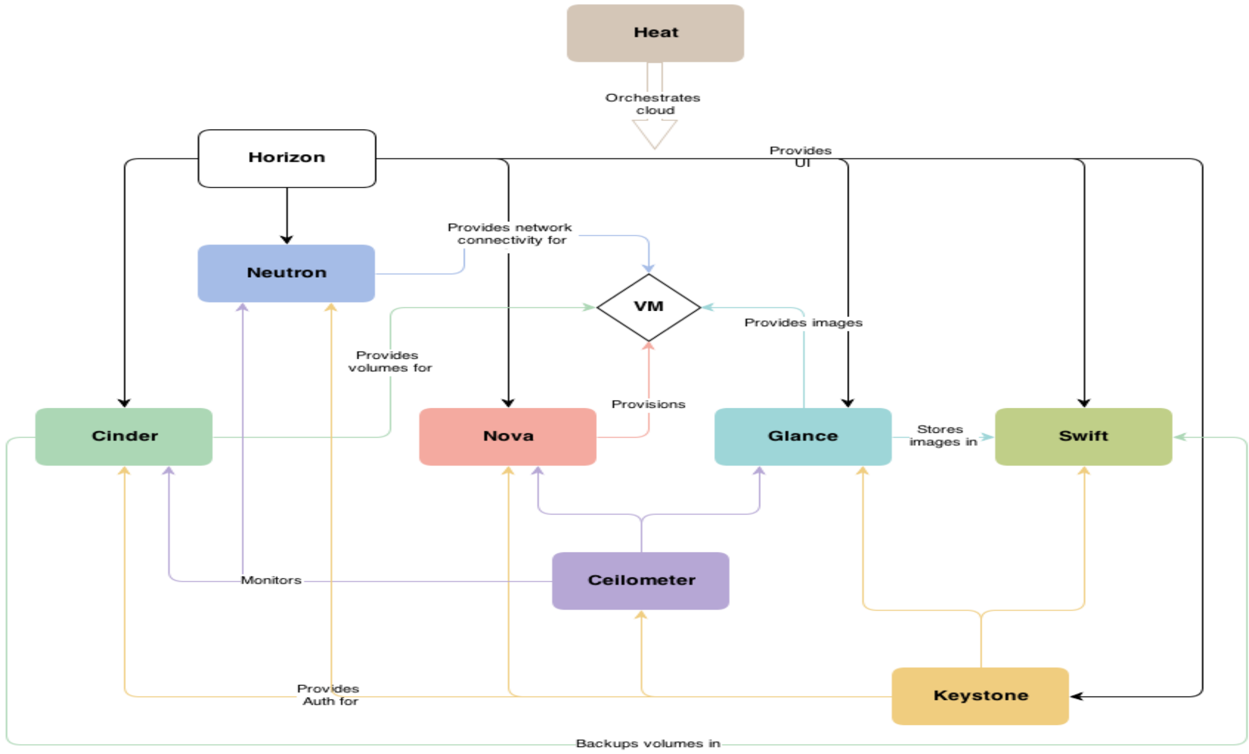

Fig 2: Components/Services of a OpenStack

A. OpenStack Identity (keystone)

This provides a central directory of users mapped to the OpenStack services. It is used to provide an authentication and authorization service for other OpenStack services.

B. OpenStack dashboard (horizon)

This component provides a web-based portal to interact with all the underlying OpenStack services, such as NOVA, Neutron, etc.

C. OpenStack Compute (nova)

OpenStack compute (codename: Nova) is the component which allows the user to create and manage virtual servers using the machine images. It is the brain of the Cloud. OpenStack compute provisions and manages large networks of virtual machines.

D. OpenStack Object Storage (swift)

This component stores and retrieves unstructured data objects through the HTTP based APIs. Further, it is also fault tolerant due to its data replication and scale out architecture.

E. OpenStack Block Storage (cinder)

This component provides persistent block storage to running instances. The flexible architecture makes creating and managing block storage devices very easy

F. OpenStack Image service(Glance)

This provides the discovery, registration and delivery services for the disk and server images. It stores and retrieves the virtual machine disk image.

G. OpenStack Networking (neutron)

It is a pluggable, scalable and API-driven system for managing networks. OpenStack networking is useful for VLAN management, management of IP addresses to different VMs and management of firewalls using these components.

H. OpenStack Telemetry Service (Ceilometer)

It monitors the usage of the Cloud services and decides the billing accordingly. This component is also used to decide the scalability and obtain the statistics regarding the usage.

I. Orchestration Heat

ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 5, May 2016

This component manages multiple Cloud applications through an OpenStack-native REST API and a CloudFormation-compatible Query API.

V INSTANCES IN OPENSTACK

In OpenStack, virtual machines are called *instances*, mostly because they are instances of an image which is created per request, and which is configured when launched. Figure 3 shows the example instance in OpenStack.

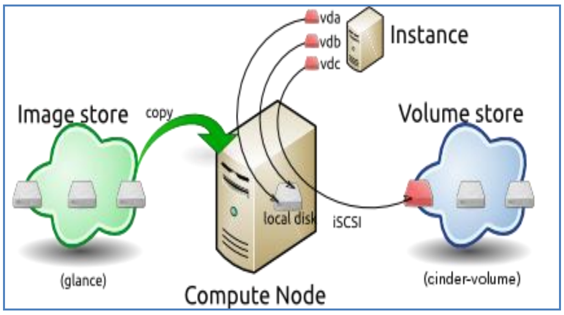

Fig 3: An instance in OpenStack

The main difference between OpenStack and traditional virtualization is the way state is stored. In traditional virtualization, the definition of the virtual machine and the virtual machine is persistent.

OpenStack can support both a persistent and ephemeral models. In the ephemeral model, an instance is launched from an image, the image is copied to the run area and once the copy completes, the instance starts running. The size and connectivity of the instance are defined at the time of launching the instance. This ephemeral model is useful to be able to scale out quickly, and maintain agility for users.

In the persistent model, the instance is launched from a volume. A volume can be any kind of persistent storage including a file, block device, LVM partition, or any other form of persistent storage.

In this case, when the instance is terminated, all the changes the user has made are kept and are present next time an instance is launched from the same volume. In the persistent case, the size and connectivity of the instance are also defined at the time the instance launches. In some sense, the persistent model in OpenStack is close to the traditional approach to virtualization.

VI STORAGE IN OPENSTACK

As already mentioned, storage used in OpenStack can be either *ephemeral* or *persistent*. Ephemeral storage is deleted when an instance is terminated, while persistent storage remains intact. Persistent storage in OpenStack is referred to as *volume*, regardless of the technology and device it is backed by. Persistent storage can either be used to launch an instance or can be connected to an instance as a secondary storage device to retain state. An example for this is a database launched as an ephemeral instance, with a volume connected to it, to save the data. Once the instance is terminated the volume retains the data and can be connected to another instance as needed.

The OpenStack Cinder service is responsible for managing the volumes and offering a framework for vendors to create plug-ins. The following figure 4 gives the block diagram for Cinder block storage service.

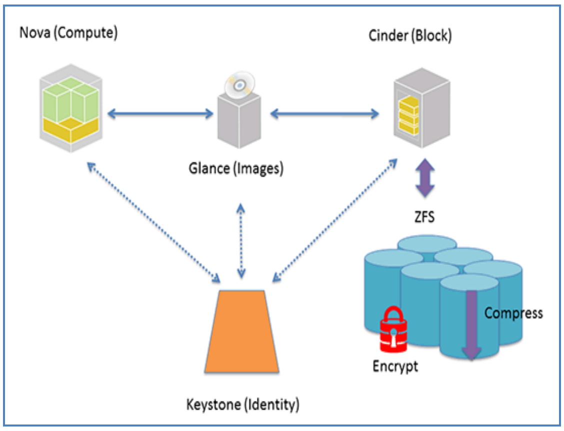

Fig 4: Cinder block storage service

If a storage vendor wants to support OpenStack deployment and allow users to create volumes on the device, the vendor must create a Cinder plug-in that allows users to use the standard calls to control the storage device.

VII NETWORKING IN OPENSTACK

This section gives an introduction to networking in OpenStack.

A. Network Services

Networking in OpenStack is one of the most powerful and sophisticated feature sets. The OpenStack networking service, Neutron, offers a complete SDN solution along with various network services, out of the box. The network services Neutron can support include: routing, firewall, DNS, DHCP, load balance, VPN, and more.

Neutron, like Cinder, offers a framework for vendors to write plug-ins for different services. A network vendor that would like to offer a custom load balancer, instead of the default load balancer provided by Neutron, can do so. This gives a user a powerful tool to build sophisticated network topologies with standard APIs.

B. Network Isolation – Tenant Networks

The tenant networks are the basis for Neutron's SDN capability. Neutron has full control of layer-2 isolation. This automatic management of layer-2 isolation is completely hidden from the user, providing a convenient abstraction layer required by SDN.

To perform the layer-2 separation, Neutron supports three layer-2 isolation mechanisms: VLANs; VxLANs; and GRE tunnels. The user is asked to define which one should be used, and sets up the physical topology. Neutron is responsible for allocating the resources as needed. Using VLANs, for example, the user is required to allocate a VLAN range and hand it off to Neutron. The user also sets up the nodes so that all of them have an interface connected to a VLAN trunk port on a switch. The trunk port needs to be configured to the same VLAN range allocated to Neutron.

Once such configuration is done whenever a user defines a new network, Neutron automatically allocates a VLAN and takes care of the isolation for the user. The user does not have to manage VLANs and does not need to be aware of which VLAN was assigned to the network which was just created.

VIII OPENSTACK VERSIONS /RELEASES

The Table 2 give brief information regarding the versions of OpenStack from its first release to till date.

SNO	SERIES NAME	RELEASES	DATE
1	Mitaka	Due	Not Announced
2	Liberty	Due	Oct 15, 2015
3	Kilo	2015.1.1	Jul 30, 2015
		2015.1.0	Apr 30, 2015
4	Juno	2014.2.3	Apr 13, 2015
		2014.2.2	Feb 5, 2015
		2014.2.1	Dec 5, 2014
		2014.2	Oct 16, 2014
5	Icehouse	2014.1.5	Jun 19, 2015
		2014.1.4	Mar 12, 2015
		2014.1.3	Oct 2, 2014
		2014.1.2	Aug 8, 2014
		2014.1.1	Jun 9, 2014
		2014.1	Apr 17, 2014
6	Havana	2013.2.4	Sep 22, 2014
		2013.2.3	Apr 03, 2014
		2013.2.2	Feb 13, 2014
		2013.2.1	Dec 16, 2013
		2013.2	Oct 17, 2013
7	Grizzly	2013.1.5	Mar 20, 2014
		2013.1.4	Oct 17, 2013
		2013.1.3	Aug 8, 2013
		2013.1.2	Jun 6, 2013
		2013.1.1	May 9, 2013
		2013.1	Apr 4, 2013
8	Folsom	2012.2.4	Apr 11, 2013
		2012.2.3	Jan 31, 2013
		2012.2.2	Dec 13, 2012
		2012.2.1	Nov 29, 2012
		2012.2	Sep 27, 2012

Table 2: Releases of OpenStack

IX CONCLUSION

The architecture analysis and characteristics feature of Openstack shows it is useful to deploy large-scale cloud deployments for private, public and hybrid cloud and that too economically. Today corporations, service providers, SMBs, researchers, and global data centres all looking towards or using Openstack. Openstack being used and supported by various commercial and non-commercial houses is a proof for the standardization and maturity of Openstack. Since its first release it has been evolved and besides fixing up of various bugs,

Apache 2.0 license. Anyone can run it, build on it, or submit changes back to project. Openstack removing fear of proprietary lock-in for customers and create a large ecosystem that spans cloud providers. Various contributors are contributing towards it and new features are adding up day by day. Openstack matters because of all its code is licensed under

REFERENCES

[1] PARKHILL, D. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1996.
[2] https://vivekraghuwanshi.wordpress.com/openstack-and-its-component/
[3] http://getcloudify.org/2014/07/10/what-is-openstack-tutorial.html
[4] http://www.openstack.org/
[5] Openstack. *Openstack* . [Online] www.openstack.org. Ubuntu cloud. *ubuntu*. [Online] www.ubuntu.com.
[6] http://docs.openstack.org/icehouse/training-guides/content/operator-getting-started.html
[7] http://getcloudify.org/2014/07/18/openstack-wiki-open-cloud.html.