International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
National Conference "NCPCI-2016", 19 March 2016
Available online at www.ijrat.org

Openstack Manila Sharedriver for Samba

Manasi Kulkarni[1], Quinston Pimenta[2], Akshay Pawar[3], Rohidas Pathare[4]

Computer Department[1,2,3,4], Vishwakarma Institute of Information Technology,Savitribai Phule Pune University, Pune[1,2,3,4]

Email: kmanasi95@gmail.com[1] , quinstonpimenta@outlook.com[2] , pawars.a.42@gmail.com[3] , rohidaspathare@gmail.com[4]

Abstract - OpenStack is a free and open-source cloud-computing system software. It is primarily deployed as Infrastructure as a Service(IaaS), which basically means that the companies who own data centers use it to manage all their hardware resources and run large-scale applications for clients. The mission of OpenStack is to provide open source cloud computing software platform that will meet the needs of public and private clouds, regardless of size. It is massively scalable. The technology which makes OpenStack actually possible is the inter-related open-source projects which deal with the controlling of the pools of computing, networking and storage. The users manage their OpenStack instances, as they are called, through a command line or a dashboard or in some cases a Restful API. Each aspect of the entire project is its own separate project. One of which is actually, Manila. Manila or OpenStack shared file system service provides an Open API to manage shares in a vendor agnostic framework. It works with Linux NFS and Samba server. Standard primitives include ability to create, delete and provide/deny access to a share and can be used in independent or in a variety of different network environments. The purpose of this project is to establish a shared file system service for OpenStack. We are trying to dynamically create a mechanism to share files between multiple virtual machine instances on the fly.

Index Terms - OpenStack, Manila, NFS, Samba server.

1. INTRODUCTION

OpenStack means a collection of open source software projects that the service providers can use to setup their cloud compute and storage infrastructure.
Initial contributors are Rackspace and NASA. Rackspace contributed the object storage part, whereas NASA contributed the compute part.
Main OpenStack services are - Nova, Swift, Keystone, Glance, Horizon and Manila.

1. Nova is the computing controller. It handles all activities for the lifecycle of instances . Various components of nova are: nova-api, rabbit-mq server, nova-compute, nova-network, nova-volume and nova-scheduler.
2. Swift provides object store for OpenStack. It can store billions of objects distributed across nodes. It is scalable in terms of size and capacity.
3. Keystone provides identity service for all components and allows authentication and authorization for all the OpenStack components.
4. Glance is the OpenStack Imaging service. It is used for lookup and retrieval for virtual machine images. There are two components of glance: glance-control and glance-registry.
5. Horizon is the web-based dashboard. It is used to administer OpenStack services.
6. Manila is the OpenStack shared file system service. It provides an API to manage shares. It has ability to create, delete or provide/deny access to a share.

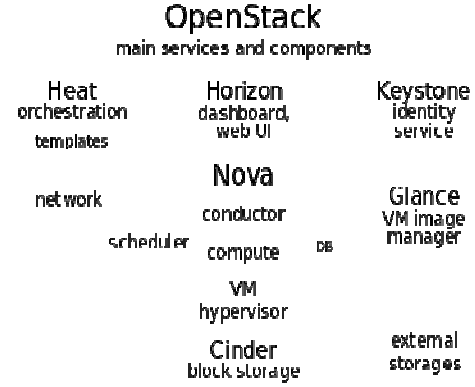

Fig 1: OpenStack and its components[4]

2. LITERATURE SURVEY

Understanding cloud computing, basics of OpenStack and the services provides by OpenStack is discussed by Atul Jha[1] and Anuj Sehgal[2]. The architecture of OpenStack according to [1] is as follows:

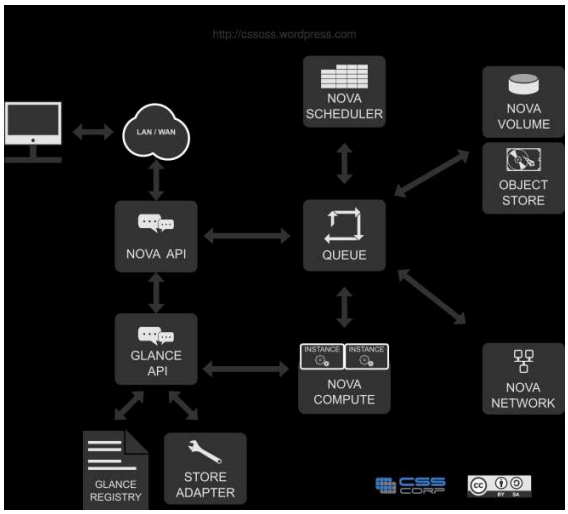

Fig 2: OpenStack Architecture[1]

Requirement of OpenStack these days is increasing. Many companies are in an attempt to deploy their cloud infrastructures. Companies using OpenStack are Bloomberg, BestBuy, Comcast, PayPal etc.[3]

Understanding Manila and to get a vision for establishing shared filesystem for OpenStack. We also understand how to use the file share service and its use as a core service to OpenStack[4]

Developer's OpenStack is known as devstack. It provides and maintains tools used for installation of OpenStack services.[5]. Installation of devstack, administering OpenStack cloud is described in [6]. We can access OpenStack services either through dashboard or by using OpenStack command-line clients. Ubuntu-specific installation of latest version of OpenStack(Liberty) is provided in [8].

Yoji Yamato et al. have proposed a fast and restoration method of virtual resources on OpenStack when physical servers or virtual machines are down.[10]. Beth Cohen[9] has suggested to plan, design and architect the cloud properly, understanding the user's need and use-cases. Ibad Kureshi et al.[11] have researched to provide a private cloud to improve High Performance Computing(HPC) research infrastructure.

3. EXISTING TECHNOLOGIES

Various types of drivers are used as the storage backend for serving file shares to manila clients.
Manila is not restricted to deployment in traditional storage arrays. Hence, there is development activity on Ceph and GlusterFS storage platforms. Also, manila is deployed on traditional systems like NFS, CIFS and HDFS.

Another type in which Manila support is implemented is without using an external storage array, instead using a generic driver provided by Manila. This driver creates a file share using Nova, with each file share creating a new VM.

4 PROPOSED WORK

We are trying to deploy Manila using Samba server. It mainly works on CIFS. Although, driver can be written for GlusterFS, HDFS etc. Main aim is making the new driver active, as the active driver is generic one. We will create a manila instance first. After installing the native client, we need to mount a volume through CIFS protocol, as Samba works on CIFS. Then we will set up Samba configuration and export the mount point of volume. These changes are to be done in smb.conf file. The difference is, while using Samba for windows, we have to export volumes manually; whereas Manila helps us to export the volumes automatically. Thus, we can create different VMs operating in the same network.

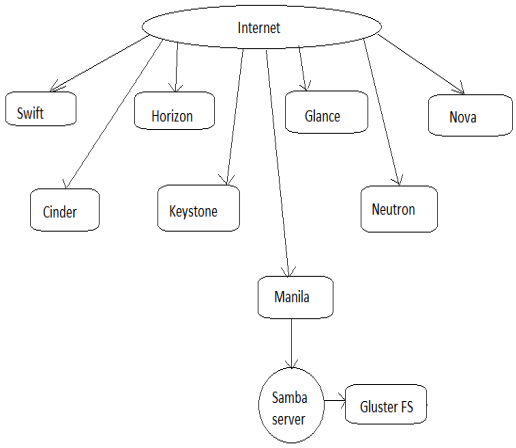

Fig 3: Proposed system

5 CONCLUSION

The outcome of this project is to implement a shared file system storage. The Manila project provides an API for management of shared file systems with support for multiple protocols and back-end implementations. Simply, the goal of Manila is to do what Cinder has done for block storage. We aim to provide a vendor neutral management interface that allows provisioning and attaching shared file systems such as NFS, CIFS, HDFS etc. We aim to mirror the architecture of Cinder, with support for a public REST API. We plan to design solutions that are compatible with the OpenStack ideas of modularity and scalability.

REFERENCES

[1] Atul Jha, "OpenStack Beginner's guide(for Ubuntu-precise) ", CSS Corp, v3.0, 2012
[2] Anuj Sehgal, "Introduction to OpenStack", Jacobs University, 2012
[3] OpenStack website, http://www.openstack.org
[4] OpenStack wikipedia website, https://wiki.openstack.org/wiki/Manila
[5] OpenStack devstack website, http://docs.openstack.org/developer/devstack
[6] Admin-user guide for OpenStack, http://docs.openstack.org/user-guide-admin/
[7] Ganesha library OpenStack, http://docs.openstack.org/developer/manila/devref/ganesha.html
[8] Ubuntu installation guide, http://docs.openstack.org/liberty/install-guide-ubuntu/
[9] Beth Cohen, "OpenStack Introduction and Architecture", Virtualization Group, Boston, 2014
[10] Yoji Yamato, "Fast and reliable restoration method of virtual resources on OpenStack", Member of IEEE, 2014
[11] Ibad Kureshi et al. , "Advancing Research Infrastructure using OpenStack", IJACSA, 2013