International Journal of Research in Advent Technology, Special Issue, NCRCEST 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Setting Up A Private Cloud Using Free Open Source Software

Mr. A. Nageswara Rao[1] Mr. S Pratap Singh[2]
[1,2] *Associate Professor,*
[1]*CMRIT, Hyderabad*
[2] *HITAM, Hyderabad*
[1]*Email:naguanr.rao@gmail.com , pratap.singh.s@gmail.com*

Abstract- There is rapid development in computer technologies both in software and hardware is observed. New hardware device are launched in the market and there is huge demand for the new technologies as they are very user friendly, reduces the computation cost and one time investment. These demand increases the use of new technologies that are less expensive, flexible and effective ways of utilization of resources. In this paper we present how to setup a Private Cloud using Free Opens Source Software (FOSS). The Private Cloud is setup using free open source software called Opens Stack. Creating virtual machines and sharing of files from one virtual machine to another and creating the floating IP which makes the private network of the virtual machine to public network so that it can be accessed by anyone who knows the IP.

Index Terms- Private Cloud, Free Open Source Software (FOSS), Virtual Machine, IaaS, PaaS, SaaS.

1. INTRODUCTION

The concept of Cloud Computing started from 2006-2007 through smooth historical events. It started from the concept of multiprogramming then virtualization, grid computing, SaaS, utility computing and then finally it came to today IT world. The word Cloud Computing can be defined as " Cloud computing is the on demand availability of computer system resources, especially data storage and computing power, without direct active management by the user"[1].there are many different definitions given by many people as per their requirement. Before the concept of cloud computing traditional infrastructure for core IT had to hectic job. The client used to manage code, data, runtime, middleware, operating system and little later virtualization, servers, storage and networking. The client or organization used to take weeks to setup and deploy the required setup. It was also difficult job to maintain or upgrade the software and hardware for client. But with the invention of Cloud Computing it solved many problems step by step. Cloud Computing provides the following services Infrastructure as a Service(IaaS), Platform as a Service(PaaS), and Software as a Service(SaaS). In Infrastructure as a Service(IaaS), the client require to manage code, Data, Runtime, Middleware, and Operating System where as the vendor manages Virtualization, servers, Storage and Networking. In Platform as a Service(PaaS), the clients need to manage code and data whereas the vendor needs to manage runtime, Middleware, Operating System, Virtualization, servers, Storage and Networking. In Software as a Service(SaaS), the clients need not to manage any service whereas the vendor manages all the services like code, data, Runtime, Middle ware, Operating System, Virtualization, servers, Storage and Networking. So the cloud computing provides the following benefits like the client needs to simply sign up and start using business application, client can access all the application and data from any connected endpoints, no data is going to be lost if the computers crashes, there is no need to install, update, and maintain the softwares, and can extend the service dynamically upto usage needs[2]. Now a days the demand of cloud computing is increase rapidly. There are many clouds available in the market. Many of the clouds are providing the paid services as per the utilization of the services on the cloud. But there is other community of clients or organizations who are working for free open source software. Still there is lot of confusion which cloud computing is best suitable for their organization.

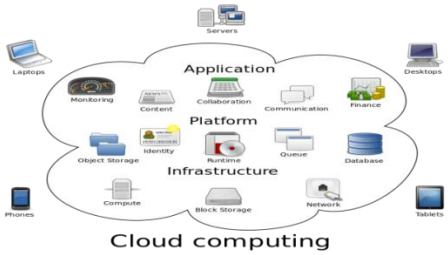

Fig: 1 Cloud Computing architecture

2. LITERATURE SURVEY

The authors Babar and et.al [3] builds a secure and scalable private Cloud Infrastructure with Open Stack, presents the tutorial and discussion of architectural and technological challenges with solution. The authors also presented the technical strength of Open Stack cloud and its related tools. They could setup a private cloud using Open Stack.

The authors Cash, S and et.al [4] discuss how OpenStack software is used to deliver a managed private cloud. They build a Cloud that provides necessary levels of controls that can focus on application and services rather than infrastructure. They also discussed about the design challenges of the customers and address the challenges using open source technologies.

The authors Mullerikkal, J and et.al [5] provides a comparative study of two middle ware technologies (like Open Stack and CloudStack) that are used to create a private cloud. They compares with respect to complexity, stability and performance. Their comparative study of experimental results shows that OpenStack performs better than Cloud Stack.

The authors Vogel, A and et.al [6] provides a contribution to analysis of IaaS domain by mapping new insights and discussing the challenges for improving cloud services. The author compared cloud tools like Open Nebula, OpenStack and Cloud Stack. The authors compared these tools with respect to support for flexibility and resiliency. They deployed tools using mutual hypervisor (KVM). Their experimental results show that OpenStack is the most resilient and CloudStack is most flexible for deployment.

The authors Kang, M.,and et.al [7] provides a comparative study of systems performance using Amazon EC2, which is very large public cloud and OpenStack which is most popular Cloud. Their experimental parameters are CPU, Memory, and Storage and network utilization. They carried their comparison on different virtual machines.

3. PROPOSED SYSTEM

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. The system architecture of OpenStack is as depicted in Fig.2. Its services can be accessed via APIs. The important components of OpenStack are Nova, Swift, Keystone and Glance, Keystone and Horizon[8].

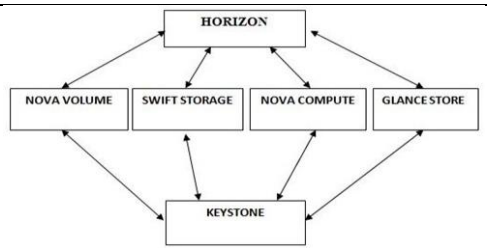

Fig.2.OpenStack System Architecture

- ✓ 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.
- ✓ 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.
- ✓ Swift: Swift can store billions of virtual object distributed across the nodes. The swift offers built- in redundancy, failover management, archiving and media streaming. Swift plays an important role in scalability.
- ✓ Keystone: Keystone provides identity and access policy services for all components in the OpenStack family. All components of OpenStack including Swift, Glance, and Nova are authenticated and authorized by Keystone.
- ✓ Horizon: Horizon can be used to manage

instances and images, create key pairs, attach volumes to instances, manipulate Swift containers etc.

4. IMPLEMENTATION

The proposed system is implemented using open source software called Openstack and Ubuntu operating system[9]. 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.3 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.

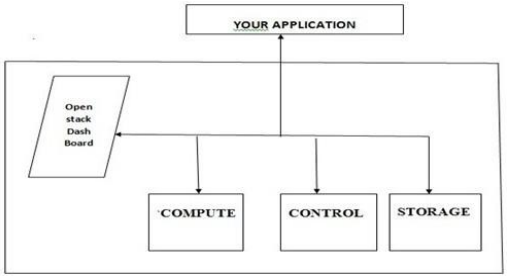

Fig.3 Open Stack Implementation Architecture

Module 1: Compute Node

The installation of nova packages is carried out by downloading the nova packages which install the most the packages like (nova-api, nova-compute, nova-network etc.) to work nova on the open stack.

Module 2: Control Node

- ✓ The installation of glance packages is carried out by downloading the glance which installs the packages (like glance-api, nova- registry etc.) to work on the openstack.
- ✓ The installation of keystone packages is carried out by downloading the keystone packages that adds most of the packages (like python-keystone, python-keystone etc.) to work with keystone on the open stack.
- ✓ The installation of horizon packages is carried out by downloading the horizon packages that adds most of the packages that expected to work dashboard on the OpenStack.

Module 3: Storage Node

The installation of swift packages is carried out by downloading the swift packages that install the added packages like (swift-proxy, swift- account, swift-container etc.) to work swift on the OpenStack.

4.1 Installation of OpenStack

Now the next step is installation of OpenStack, ✓ which involves the following steps

Step1: we have to install git software.

Step2: Installation of devstack, which is useful for cloning into devstack.

Step3: The third step is inclusion of Block storage CEPH and it is included by copying a localrc file in the devstack folder.

Step4: Finally the devstack is ready and now by using the below command we can start deploying the openstack along with its required layers.

After the installation of openstack is finished. Its time to log into into localhost IP and launching Virtual Machines in same and different network using CLI.

Our proposed system is about the creating virtual machimes on same network and different networks (case 1)

CASE 1 (a): CREATING VIRTUAL MACHINES ON SAME NETWORK

Steps are as follows

- ✓ Log in into the openstack
- ✓ Launching of VM.
- ✓ Log in into VM
- ✓ Writing the files on VM
- ✓ Taking snapshot of the Running VM: here we take the snapshot of the running instance as shown in below screenshot fig:

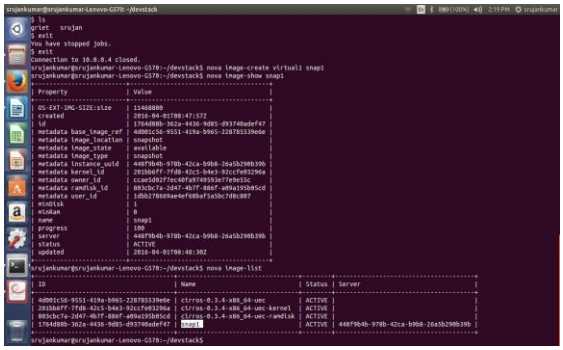

Fig.4. Creating the snapshot of the running instance

✓ Launching another VM with the snapshot.

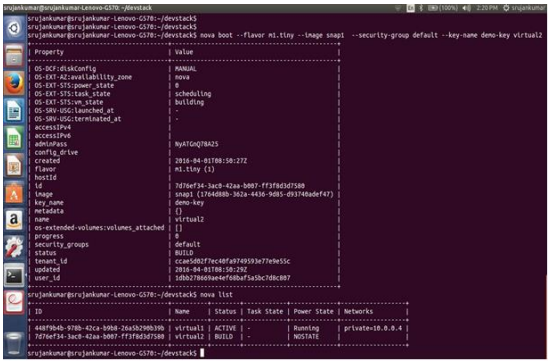

Fig.5.Launching another instance using the snapshot

✓ List the virtual machine created on the same network.

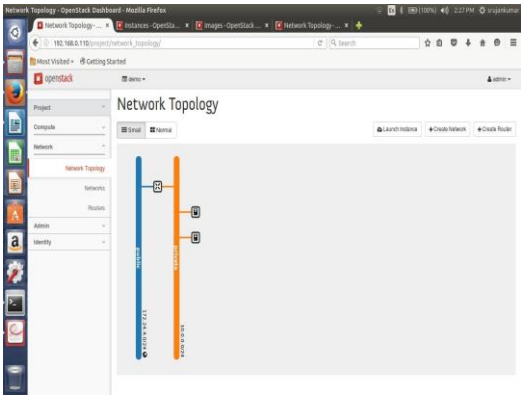

Fig.6.VM'S created on the same network

✓ Now login in to VM2 and checking for the file created on VM1

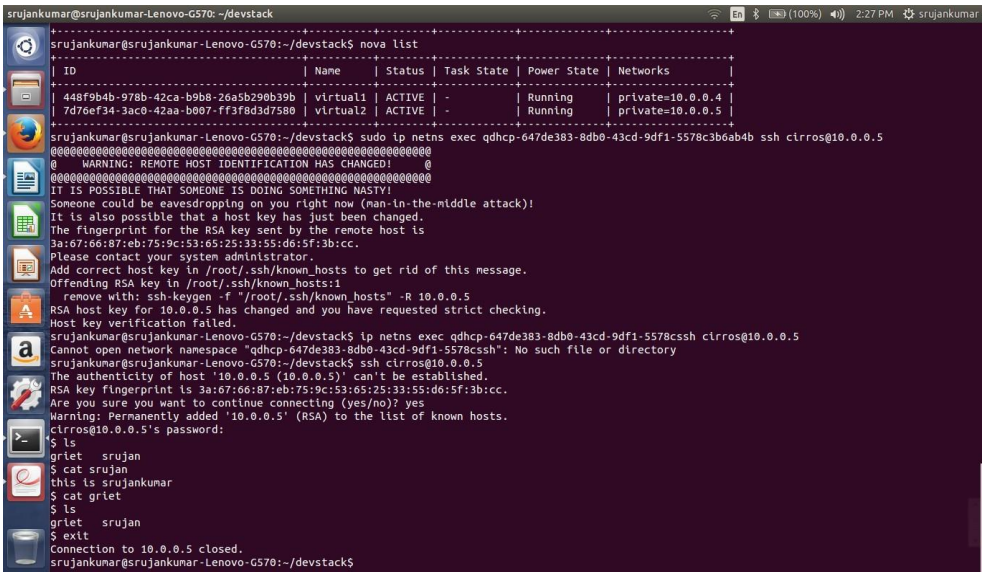

Fig.7:.Log in into VM2, checking the files created on VM1 and closing the connection of the VM2

CASE 1 (b): CREATING VIRTUAL MACHINES ON DIFFERENT NETWORKS

✓ Creating Networks in OpenStack:

Since we have to launch VM'S in different networks we have to create two different networks first.

Similarly we have to create another network named as created first one. After creation of network we need to check whether the networks are created or not.

The below figure shows the two different networks created through user interface

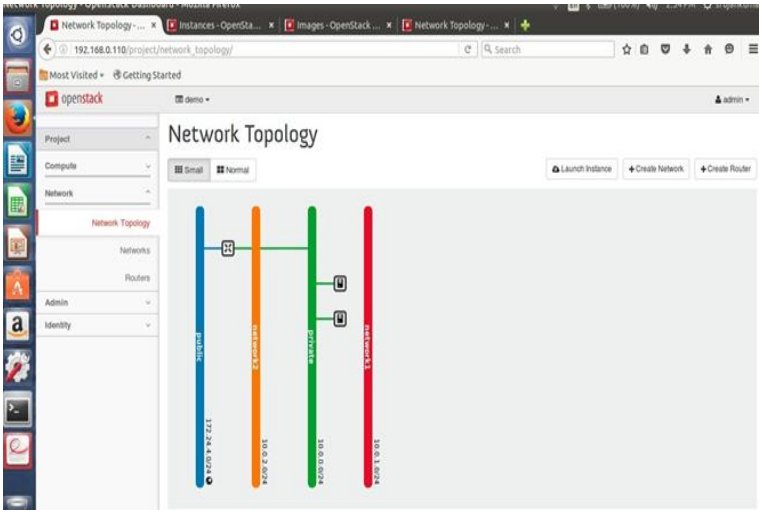

Fig.8.Two different Networks Created

✓ Creating the router and interfacing networks to the router

In openstack the neutron is configured in such a way that the two networks or isolated from each other and cannot communicate without any medium. So we have to create a virtual router so that the networks can interfaced to the router and communicate with each other. The below figure shows the router interfacing the two networks

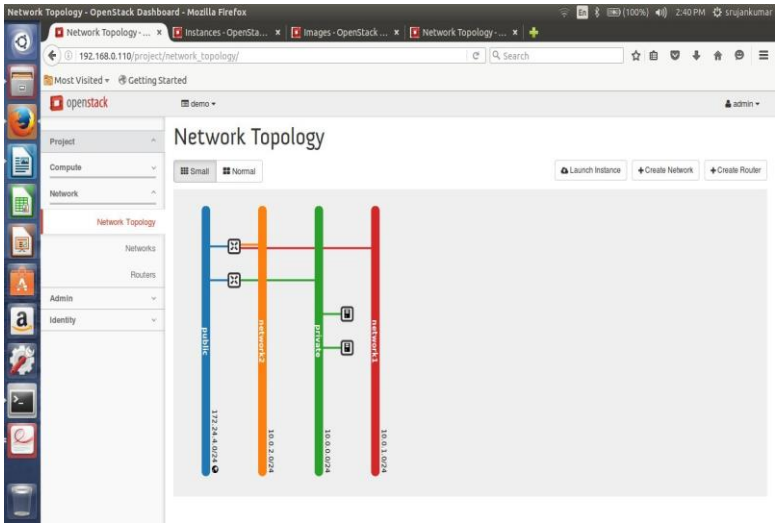

Fig.9.Router interfacing the two different networks

- ✓ Adding rules to the default security group

To login into the VM'S launched in a specified network we have to add rules to the default. The rules we have to add are TCP and ICMP. The tcp rule is added on port no. 22 and icmp does'nt have any port no.

- ✓ **Launching the virtual machines in a specified network**

Previously in same networks we have discussed how to launch a vm but when u want to launch a vm when the multiple no. of networks are present you have to be more specific in command. Now we have to mention the network ID in which u want to launch the vm.

CASE 2: Creating the floating IP'S

- ✓ Actually the floating IP is given to the VM so that it can be made into a public network from private network so that it can accessed by anyone who knows IP. If floating IP is not assigned to the VM launched in networks we have to know the network ID also which is accessible by the persons present in it. By doing this we can access VM very easily. We can access it like we do in same networks.
- ✓ Next logging into virtual machine:Now we have created a VM in a network so that the logging of the VM has to be more specific in the command to login into the VM. We have to specify the network id in the command to login into the VM.
- ✓ Taking snapshot of the VM :here we take the snapshot of the VM and check whether images had created or not. If the image is appeared we are successful in creating the snapshot image.
- ✓ Launching another VM with the snapshot in another network, Now we have to launch another VM using the snapshot image we have created using the same command we have used before. After successfully launching the VM login into the VM as described before. Now if we type the command "ls" it has to display the files we have saved in the first VM we created. We can check the VM's that has created in two different networks as shown in below screen shot fig.11.

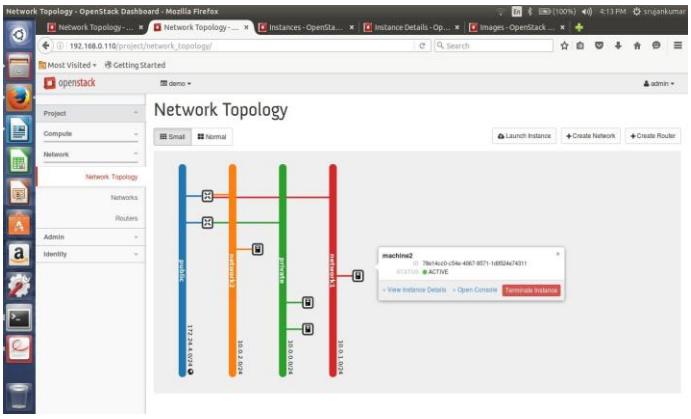

Fig.10.VM'S created on the different networks

- ✓ Next login in to VM2 and checking for the file created on VM1
- ✓ Ping and ssh the other virtual machine, we have to check whether it can ping the VM present in other network because the main concept of this use case is to connect the two networks to communicate with each other via virtual machines. The below screen shot shows that pinging in other network.

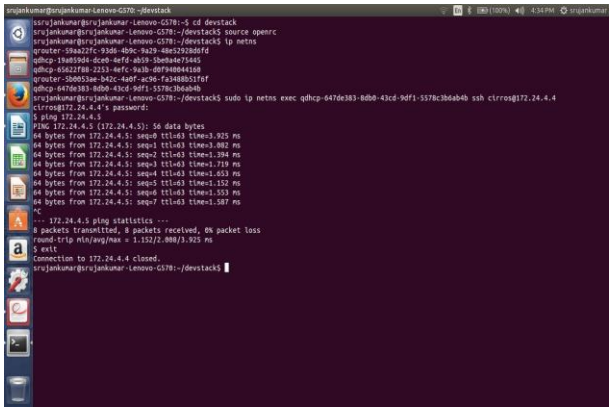

Fig.11.Result of different networks

5. CONCLUSION

We have installed Ubuntu software in the PC and installed OpenStack software by using some commands in the terminal. Next we have entered into the OpenStack and created a network & subnet in the OpenStack and attached some rules for the default security group. Then we had launched virtual machine and created a volume and attached it to the virtual machine. Next we had get login into the virtual machine and created a file and written some data and logged out of the virtual machine, and created a snapshot and attached the rules to default security group and launched another virtual machine through this snapshot and attached volume to the virtual machine. Now we had logged into the virtual machine launched through the snapshot and checked whether the files of VM1 are present in the snapshot VM are not, then our use case1 has been completed.

Now in different networks we had created a network & subnet in the OpenStack and next we had created a router and interfaced two networks to the router and had set a gateway to the router. Next we had to launched another virtual machine and pinged the first virtual machine which gives a communication between them. We had created the floating IP which makes the private network of the virtual machine into public network so that it can accessed by anyone who knows IP. Thus the use case2 has also been completed.

REFERENCE

[1] https://en.wikipedia.org/wiki/Cloud_computing

[2] https://www.ibm.com/cloud-computing/

[3] Babar, A., & Ramsey, B. (2015). Tutorial: Building Secure and Scalable Private Cloud Infrastructure with Open Stack. https://doi.org/10.1109/edocw.2015.23.

[4] Cash, S., Jain, V., Jiang, L., Karve, A., Kidambi, J., Lyons, M., … Patel, N. (2016). Managed infrastructure with IBM Cloud OpenStack Services. *IBM Journal of Research and Development*. https://doi.org/10.1147/jrd.2016.2515418.

[5] Mullerikkal, J. P., & Sastri, Y. (2016). A Comparative Study of OpenStack and CloudStack. In *Proceedings - 2015 5th International Conference on Advances in Computing and Communications, ICACC 2015*. https://doi.org/10.1109/ICACC.2015.110.

[6] Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., & Fernandes, L. G. (2016). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *Proceedings - 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2016*. https://doi.org/10.1109/PDP.2016.75.

[7] Kang, M., Kang, D. I., Walters, J. P., & Crago, S. P. (2017). A Comparison of System Performance on a Private OpenStack Cloud and Amazon EC2. In *IEEE*

International Conference on Cloud Computing, CLOUD. https://doi.org/10.1109/CLOUD.2017.47.

[8] "ReleaseNotes/Liberty - OpenStack". Wiki.openstack.org. Retrieved 2015-10-19.

[9] "ReleaseAnnouncement/Liberty - OpenStack". www.openstack.org. Retrieved 2015-10-19.