

The Study on the Construction of the Computing Platform Based on OpenStack

Chunyu Li

Kunming Institute of Physical Science and Technology, Kunming University

Email:125185139@qq.com

Key words:cloud computing; OpenStack; nova-network ;VLAN

Abstract:When the information network of an enterprise grows to a certain size, higher requirements will be set to the security and the use of network resources maximum effectiveness. In the current internal network of an enterprise, various needs of IT applications of the enterprise can be met by commonly using VLAN in conjunction with other techniques. After OpenStack nova-network is deployed into VLAN mode, it can be well mixed with the enterprise network, providing the security of the entire cloud system and the efficiency of resources.

Introduction

With the development of information technology, many companies have a variety of management information systems, which gradually become the indispensable condition for its daily work. For an enterprise, it has become a major problem of information construction to improve the utilization of resources, to play the maximum efficiency of resources and to improve its network security under the conditions of the surge of the amount of data, the diversity and the complexity of information resource construction. The cloud computing ,which developed from grid computing, distributed computing and parallel computing, provides users with computing, storage, information and other aspects of efficient services mainly based on computing resource pool. Moreover, It can centrally manage and allocate system resources, making it possible to effectively solve the problems in the construction of enterprise information. As a fully open source cloud computing system, Openstack has focused more and more users' attention for its unique open source and low cost. It is known as the most classic and important realization of open source cloud software of all the "cloud computing" products. This paper study the construction of college OpenStack cloud computing platform, so it has a certain reference value for promoting the development of enterprise information.

The overview of OpenStack

OpenStack is an open source cloud software , which is an open source project jointly developed by NASA and Rackspace . Its goal is to provide a fully open, large-scale deployment of cloud computing platform [1]. OpenStack supports almost all types of cloud environments, whose goal is to provide a rich and uniform cloud computing management platform which is simple to implement and massive to extend. OpenStack provides infrastructure solutions (IaaS) through a variety of complementary as well as API for each service to be integrated [2].

As a cloud platform management project, OpenStack covers networks, various aspects of virtualization, operating systems, servers, etc. Its main core project (OpenStack services) includes Compute, Object Storage, Image Service, Identity Service, Network, Block Storage, Dashboard and other services. Its service relationship is shown as below:

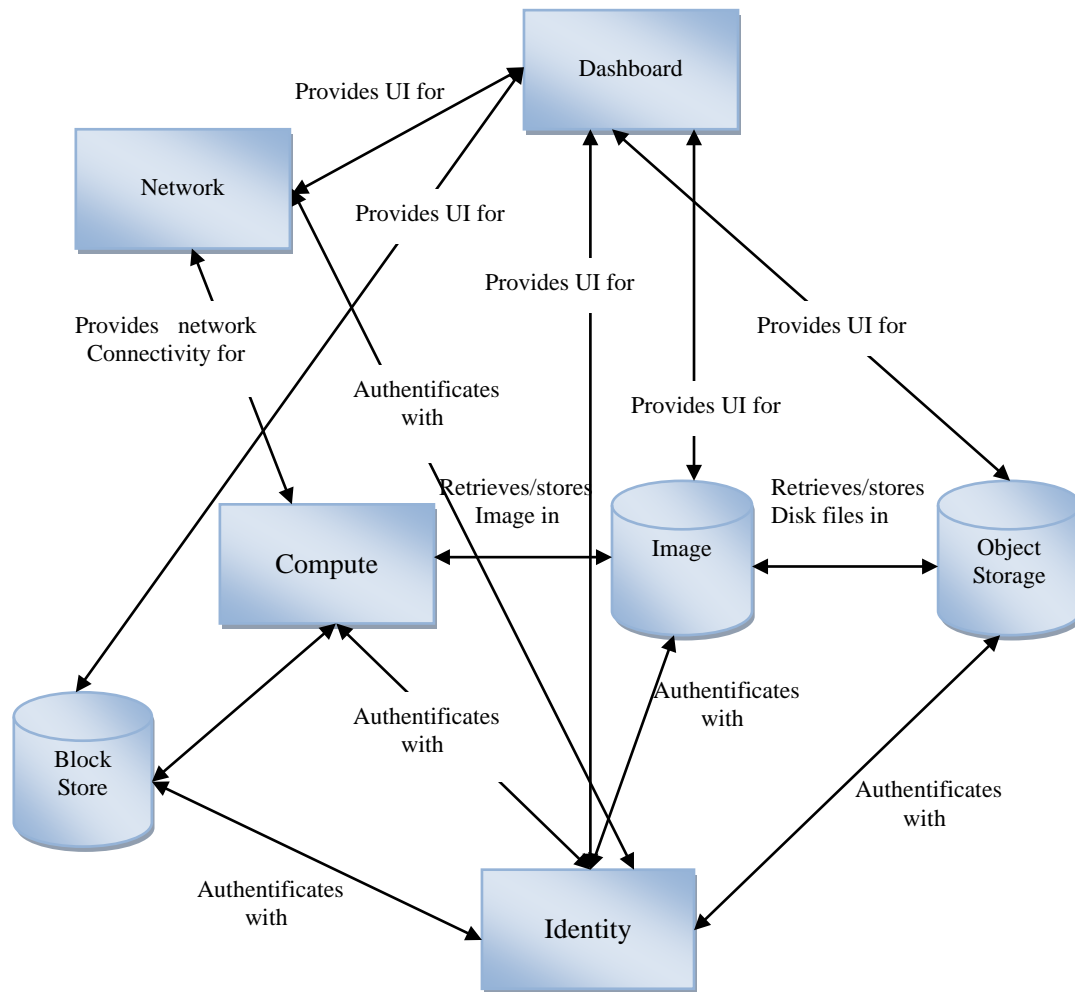


Fig:2-1 OpenStack The service relationship

(1) Compute (codename Nova) The service provides a framework for deploying and managing large number of virtual machine instances and is the core component of the cloud computing. It mainly manages the lifecycle of virtual machine instances, provides virtual services according to user needs. Besides, it is responsible for creation, startup, shutdown, suspend, pause, adjust, migrate, restart, destruction operation, configuration CPU, memory and other information on the specifications for the virtual machine. It is the central controller of the cloud platform, including nova-api, nova-cert, nova-consoleauth, nova-novncproxy, nova-scheduler and other components.

(2) Object Storage (codename Swift) It provides system object storage service. It is a system to realize object storage via the built-in redundancy and high fault tolerance mechanisms in massive extensible system, which allows for storage or retrieval of documents. It can both provide Glance with image storage and Cinder with volume backup service. Under normal circumstances, The object storage service can have a lot of applications that focus on processing, including data replication and archiving, video, image integration services, various types of data storage, new integration, the creation of cloud storage based on the Web's applications and so on [3].

(3) Image Service (codename Glance) It is a finding and retrieval system of a virtual machine. It supports a variety of virtual machine image format, which has the functions to create upload images, remove the mirror, edit the mirror basic information. Images can be managed, resulting in simplifying the processes and operations of mirroring management.

(4) Identity Service (codename Keystone) The primary responsibility of Keystone is to provide OpenStack with authentication and service rules and service token, managing Domains, Projects, Users and Groups, Roles. Keystone is equivalent to a service bus or the registry of the entire OpenStack platform framework. Meanwhile it is the exit point which can register services via other

services of certification service components. Communication with each other and call between two services must be authenticated by the service components of Keystone. Only in this way can Endpoint of target service be obtained. The target service invocation can be found according to it.

(5) Network (codename Neutron) The network provides cloud computing virtualization technology and other services of OpenStack with network connection services. It also provides API a user-defined and access network, the plug-in architecture which supports many mainstream manufacturers and network technology.

(6) Block Storage (codename Cinder) It provides a stable data block storage service for running instances. Its plug-driven architecture is conducive to the creation and management of block devices, such as creating a volume, deleting the volume, mounting and unmounting Vol at the instance.

(7) Dashboard (codename Horizon) It is the Web management portal of various OpenStack services . It is used to simplify the user's operation of the services, such as starting instances, assigning IP addresses, configuring access control and so on. It is integrated into the project from Essex version.

Constructing the cloud computing platform based on OpenStack

The cloud computing platform adopts nova-network VLAN deployment model based on a multi-host OpenStack, whose benefit of multi-host installation is that nova-network service deployment can be dispersed. Thus the problem that the entire cloud system communication service will stop once a single point of nova-network service disrupts in the process of the standard installation can be solved. Its deployment pattern is shown in Figure 3-1:

The cloud computing platform is established based on OpenStack, so it is a key component as OpenStack support while it works. The service and the running state of the cloud computing platform is monitored and started through the dashboard (Horizon components). The specific computing task of the cloud computing platform is scheduled and executed by the calculating components(Nova-API component) . The virtual image storage and retrieval service of the is achieved through the mirror (Glance-API component). The storage of data and information of the cloud computing platform is achieved via the object storage (Object-API component). The users' information and access management are authenticated (Keystone components) [4, 5].

Basic environment. (1)The number of servers: Use three Dell PowerEdge R730 servers, 256GB memory, 6TB hard drive, 2-way 12-core processor

(2) Control node (Host A): Since the basic control services of OpenStack consists of MySQL, RabbitMQ, Glance, Keystone, API + Scheduler and Horizon, a CentOS operating system should be installed into host A .Then six virtual machines should be deployed to the controlling server.

(3) Network services: It mainly means to installa nova-network services, using muti-host installation, which is installed along with the nova-compute point to host B and host C.

(4) Computing nodes: Install Ubuntu 12.04 operating system by computing nodes. The created virtual machines run on this computer.

(5) Installation mode: nova-network multi-host Vlan

(6) Switches: The reliable switches which support two or three functions

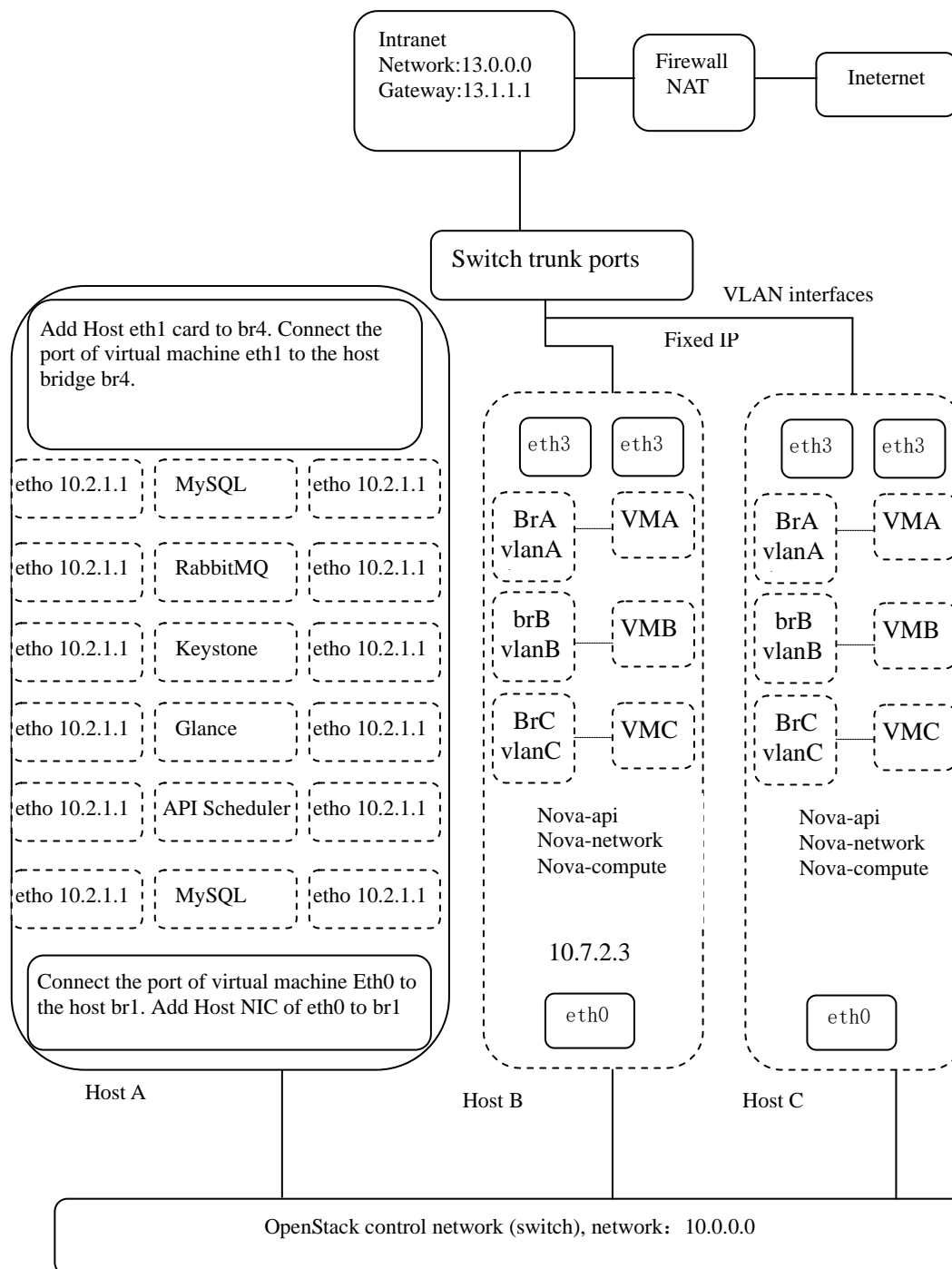


Figure 3-1 OpenStack multi-host deployment pattern nova-network Vlan

Nova-network VLAN installation environment of multi-host . 1) CentOS6.2 operating system is installed to Host A. Six virtual machines based on Ubuntu12.04 operating system run on it, which are installed MySQL, RabbitMQ, Glance, Keystone, API + Scheduler, Horizon and other Openstack control services.

(2) A host creates br1 and br4 bridge on CentOS, respectively including em1 and em4 physical NIC; each virtual machine creates two network interfaces of eth0 and eth1, which are attached to br1 and br4.

(3) Install Host Ubuntu12.04 operating system to host B and hostC, each running nova-api, nova-network, nova-compute services.

(4) Connect the network cards of Host A, host Band host C of em1 / eth0 to the first switch, forming OpenStack control network; connect em3 / eth3 to the second switch; connected eth2 of host B and host C to the third switch.

(5) The Internet gateway of the system 13.1.1.1

(6) Eth0 segment exists independently as the controlling network segment of cloud system

(7) eth1 segmen works as a VLAN interface and virtual machines are allocated to the fixed IP on this segment.

(8) Connect eth1 port to the switches inside the school. The school's internal and external network access work is completed by using a single fixed IP address of the virtual machine , while the gateway points directly to the school network switches. Nova-network is only responsible for the assigned work of IP the virtual machine's addresses. The participation of nova-network is no longer needed for the outside flow of the virtual machine, avoiding the problem that the virtual machine can not access internal and external network because of the disruption of nova-network.

VLAN component installation of computing nodes. Install VLAN related packages and load the VLAN kernel modules, as follows:

```
#apt-get install VLAN
```

```
#modprobe 8021q
```

In order to start 801.1q next time, it should be directly written to / etc / modules in the configuration file, making sure that 802.1q entry is included.

It is shown as follows

```
802.1q
```

MySQL server deployment. In OpenStack control system, MySQL not only provides the basic services for the OpenStack control system, but also provides database services for the entire system. In MySQL database, a data store database corresponding to nodes is needed to create for storing the data of controlling nodes.

(1) Configure MySQL server's IP address. The following files (/ etc / network / interfaces)are shown as follows :

```
#The loopback network interface
```

```
Auto lo  Iface lo inet loopback
```

```
#The primary network interface
```

```
auto eh0    auto eth1    iface eth0 inet static    address 10.7.1.2
```

```
netmask 255.0.0.0    iface eth1 inet static    address 13.7.1.2
```

```
netmask 255.0.0.0    gateway 13.1.1.1    dns-nameservers 61.139.2.69
```

(2) Install packages related to MySQL:

```
Apt-get install -y mysql-server python-mysqldb
```

(3) Configure MySQL to listen on all network ports, and reboot:

```
sed -i 's/127.0.0.1/0.0.0.0/g'/etc/mysql/my.cnf
```

```
service mysql restart
```

(4) Create Keystone, Glance, Nova database and give permission

The MySQL database is required for the individual components of Openstack to create a corresponding database for individual services. Its creation process is as follows:

```
mysql -u root -p mysqlpassword
```

```
CREATE DATABASE keystone;
```

```
GRANT ALL ON keystone.*TO 'KEYSTONEuSER'@'%' IDENTIFIED BY 'keystonePass';
```

```
CREATE DATABASE glance;
```

```
GRANT ALL on glance.*TO 'glanceUser'@'%' IDENTIFIED BY 'glancePass';
```

```
CREATE DATABASE nova;
```

```
GRANT ALL ON nova.* TO 'novaUser'@'%' IDETIFIED BY 'novaPass';
```

```
Quit;
```

RabbitMQ Deployment. In OpenStack control system, RabbitMQ provides message queuing services with the entire OpenStack control system. As MySQL database service, it is a part of basic

services of the control system. The IP address configuration content file (etc / network / interfaces) of RabbitMQ server is shown as follows:

```
#The loopback network interface      auto lo    iface lo inet loopback
#The primary network interface
auto eth0      auto eth1      iface eth inet static      address 13.7.1.3
netmask 255.0.0.0      gateway 13.1.1.1      dns-nameservers 222.172.200.68
After IP address configuration is completed, install packages related to RabbitMQ
apt-get install -y rabbitmq-server
```

Keystone server deployment. Keystone is responsible for providing OpenStack control system with validation of users and various public services . When Keystone relevant packages are installed, the code is shown as follows:

```
apt-get install -y keystone python-keystone python-keystoneclient python-mysqldb
```

Modify the configuration file (/etc/keystone/keystone.conf) . Modify admin_token into required modification and the database into the database for local use.

Glance server deployment. Install Glance related package. Its code is shown as follows:

```
apt-get install -y glance glance-api glance-common glance-registry python-glance
python-keystone python-mysqldb python-glanceclient
```

Glance-api.conf, glance-registry.conf, glance-api-paste.ini and glance-registry-paste.ini several configuration files need to be adjusted by using a Glance services. Thus Glance database can be created.

Nova API node deployment. Nova API service is the core computing service in OpenStack control system. The entire OpenStack system is managed and controlled via it. Furthermore, virtual machines are also scheduled and managed through it. In Nova API installation, install nova-api service at the same time to receive and respond to external requests. It supports OpenStack API, Amazon EC2 API and so on.. Nova-cert service, which is primarily for Nova certification management services, provides authentication for EC2 service; nova-scheduler service is primarily for scheduling work. It allocates computing nodes when a virtual machine is created; nova-conductor service is mainly used to compute an intermediate node to access the database, and it avoids the information inconsistency when computing nodes nova-compute service directly accesses the database .

Nova-consoleauth services is primarily for authorized users to access and use the console; nova-novncproxy service mainly provides users with a VNC proxy when they access the virtual machines

In Nova API service, Nova API also launched the start of metadata service, metadata service in 8775 ports. Metadata service provides the cloud system with id, ke, IP addresses and other information of virtual machines. The information is just needed when a virtual machine creates or starts. In multi-host network architecture, each computing node also install nova-api service for metadata agent. The requests of virtual machines of the calculating node server will be forwarded to node metadata service of Nova API.

Nova API and nova conf configuration of computing nodes The configuration of the network is shown as follows

```
#Nova-network VLAN mode
Network_manager=nova.network.manager.VLANManager
public_interface=eth3      VLAN_interface=eth1
#following will be covered by mysql novadb network setting
Fixed_range=192.168.1.0/24      Network_size=250
Dhcpbridge_flagfile=/etc/nova/nova.conf
Dhcpbridge=/usr/bin/nova-dhcpbridge
Flat_injected=False      Force_dhcp_release=true
network_host=10.7.2.2
#network:for mutil node      Multi_host=True
Enabled_apis=metadata      Metadata_listen=0.0.0.0
```

Metadata_listen_port=8775 10.7.2.2 is the IP address of the computer node GCMI.

Horizon server deployment. Horizon service provides OpenStack with a Web interface. Most of the command line can be completed by using Horizon service, simplifying the management of cloud system. Horizon package installation-related code is shown as follows:

```
apt-get install -y apache2 OpenStack-dashboard memcached python-memcache python-mysqldb python-mysqldb python-keystone
```

After the installation is completed, the configuration file should be modified /etc/OpenStack-dashboard/local_settings.py, the code is shown as follows:

```
OPENSTACK_HOST="10.7.1.4"
```

```
OPENSTACK_KEYSTONE_URL="http://%s:5000/v2.0" %OPENSTACK_HOST
```

```
OPENSTACK_KEYSTONE_DEFAULT_ROLE="Member"
```

After modifying the configuration, restart Apache2 and memcached service, as is shown below:

```
service memcached restart      service apache2 restart
```

Conclusion

This paper presents a plan to build a cloud computing platform based on multi-host nova-network VLAN mode of OpenStack. The installation and deployment is relatively simple and easy to understand. More importantly, it is very reliable. It can both meet the needs of small enterprises and large enterprises. Although this program is restricted by VLAN 4096, this limitation can be broken through if VXLAN technology of Linux is used. Meanwhile, nova-network can be deployed dispersedly to the different computing nodes in the mode of multi-host nova-network, eliminating the failure that the communication service of cloud system will interrupt if a single point of fault occurs in nova-network standard mode deployment. Another advantage of deploying nova-network by multiple hosts is to disperse the metadata services of the system simultaneously. Therefore the stability of the entire cloud system operation can be enhanced.

References

- [1] Xiaolong Wen; Genqiang Gu; Qingchun Li; Yun Gao; Xuejie Zhang. Comparison of open-source cloud management platforms: OpenStack and Open_Nebula. Fuzzy Systems and Knowledge Discovery (FSKD), 2012, 9(2457—2461);
- [2] Information on http://baike.baidu.com/link?url=C5IUxTQUiErme9uqVe4cFqIxDM-fcd0S2NV2CDJybCayevoStHahS5Av1XKt1L9AfolDHJ_7qQAnx4Dcx8fyoK 2015
- [3] Katsaros, D.; Pallis, G.; Sivasubramanian, S.; Vakali, A. Cloud computing [J]. Network, IEEE, 2011, Vol.25(No.4)
- [4] T. Coutelen, H. Elbiaze, B. Jaumard. An efficient adaptive offset mechanism to reduce burst losses in OBS networks [J]. IEEE Globecom, 2009, 4:28.
- [5] Yu Liang, Haibo Yang. Resource Monitoring System Based on OpenStack [J]. Computer Systems & Applications 2014, 4:20