

Introduction to Cloud and Establishment of Infrastructure as a Service

J. Bala Savitha¹, S. Suhasini², M. Manju Sarma³ and V. Manikumar⁴

ABSTRACT

Remote Sensing is the science of acquiring information about the earth's surface without actually being in contact with it. With the development of this imaging technology, the satellite information includes abundant data. To manage that data a powerful computing storage platform is needed, this can be provided by cloud. So, the main concern is to establish the cloud using appropriate open source tools and to integrate acquired images for data dissemination and exchange. The present study deals with building Infrastructure as a Service and launching a static instance in it.

Keywords: Remote Sensing, cloud, open source tools, Infrastructure as a Service, instance.

1 INTRODUCTION

Remote sensing (RS), also called earth observation, refers to obtaining information about objects or areas at the earth's surface without being in direct contact with the object or area. With the development of remote sensing imaging technology, the satellite information includes abundant data. At present the satellite which has been launched in the polar orbit is producing images four times a day. If the satellite launches in geospatial orbit it produces 24/7 continuous images. To process those satellite images a powerful computing storage platform is needed which can be provided by cloud.

Cloud computing is a model which provides access for shared pool of computing resources like networks, servers, storage, applications, and services that can be rapidly provisioned. The very first concept for the cloud computing was given by the John McCarthy in 1960s. With the arrival of the Virtual private networks this concept which was modernized in 1966 took implementation shape. In year 2007 the most famous

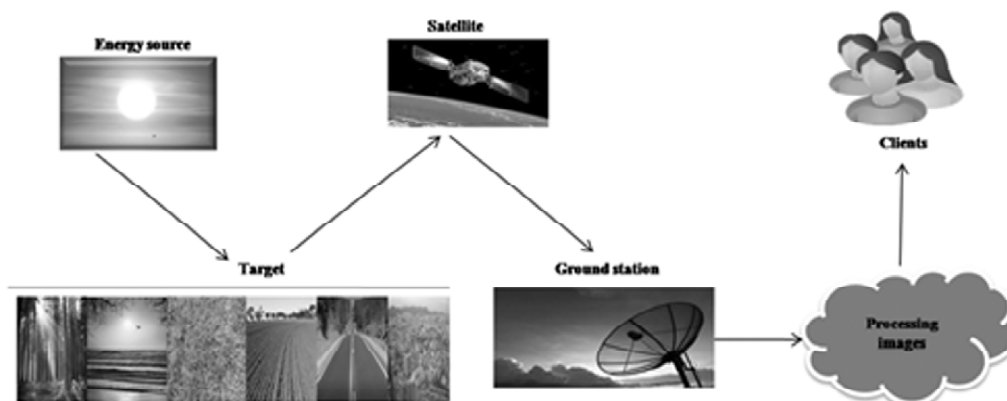


Figure 1: Remote Sensing Process.

¹ PG-Student-Department of Information Technology-VRSEC-Vijayawada, Email: savitha1203@gmail.com
² Associate Professor- Department of Information Technology-VRSEC-Vijayawada
³ Group Head, Software Group, National Remote Sensing Center-ISRO
⁴ SE, Software Group, National Remote Sensing Center-ISRO

organization such as Google, IBM and many universities worldwide started research program on cloud computing and finally in 2008 the first cloud computing system was introduced.

The (Fig.1.) shows the process of remote sensing where the energy source (sun) when falls on the targets like forests, bare land and soon the satellite takes the image of that target and sends it to the ground station for processing. The processing of the image will be taken care by cloud and it also provides the access to the clients.

1.1. Cloud Computing Architecture

A cloud computing architecture consists of a front end and a back end (Fig.2.). They connect to each other through a network. The front end side is the computer user, or client. The back end is the “cloud” section of the system [1].

The front end of the cloud computing system has clients who use the applications which are needed for accessing the cloud computing system. The interfaces may be varied from user to user according to cloud computing system. Web services like electronic mail programs use some existing web browsers such as Firefox or Microsoft’s internet explorer. In the same way other types of systems may have their own applications which provide network access to its clients. The back end has physical peripherals devices. In cloud computing, the back end is cloud itself which may be various computer machines, data storage systems and servers.

1.2. Types of Cloud Computing

Based on the location of accessing there are mainly three basic types of clouds [3]. They are:

- Private Cloud
- Public Cloud
- Hybrid Cloud

With the variation of the basic types of cloud computing there are two other types named as:

- Virtual Private Cloud
- Community Cloud

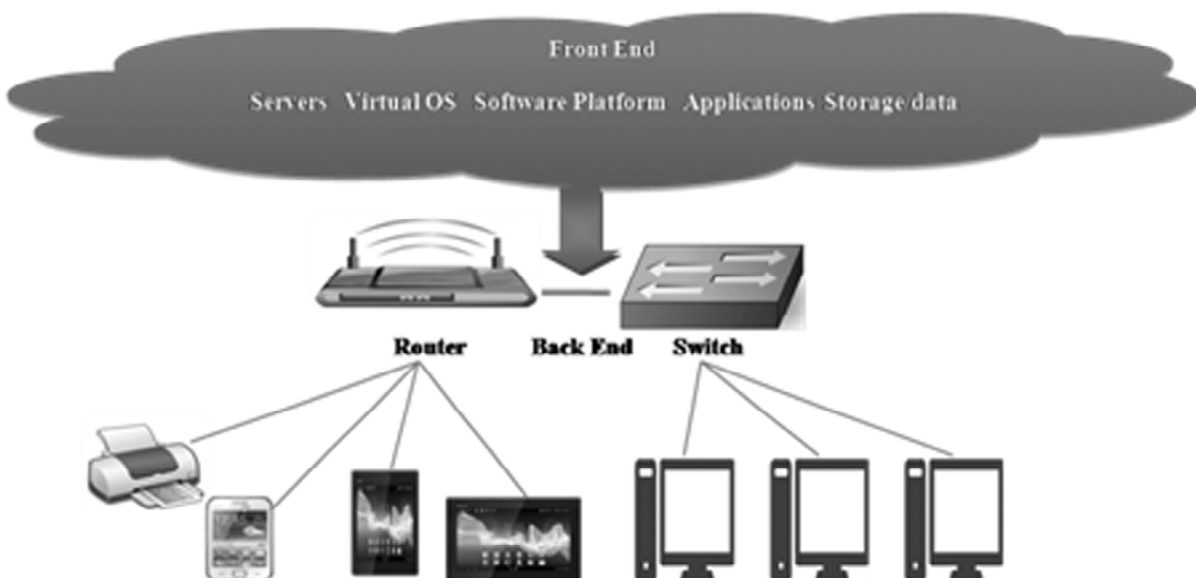


Figure 2: Front and Back End of Cloud.

1.2.1. Private Cloud

A Private Cloud is implemented within an organization. It uses a dedicated data centre infrastructure of hardware and software that is used privately by an organization. The data centre can be on-premises or off-premises but is not shared with any another organization.

1.2.2. Virtual Private Cloud

A private cloud in which data centre is shared is a Virtual Private Cloud. The data centre can be mostly off-premises. The lower IaaS is shared in a Virtual Private Cloud where PaaS and SaaS are dedicated to the organization.

1.2.3. Public Cloud

A Public Cloud is implemented by sharing with multiple organizations. It uses a shared data centre infrastructure of hardware and software which is off-premises.

1.2.4. Hybrid Cloud

A Hybrid Cloud is a combination of any two clouds where the resource sharing is done. It could be a Private Cloud and one or more Public Clouds. Similarly it could be a Virtual Private Cloud and one or more Public Clouds.

1.2.5. Community Cloud

A Community Cloud can be any one of the four clouds where it meets the needs of a community. Such communities involve people or organization that has shared interests. This includes industrial groups, research groups, standards groups, and so on.

1.3. Services of Cloud Computing

A cloud service is any resource that is provided which ranges from full applications and development platforms, to servers, storage and virtual devices. The main services of cloud computing are:

- Infrastructure as a Service(IaaS)
- Platform as a Service(PaaS)

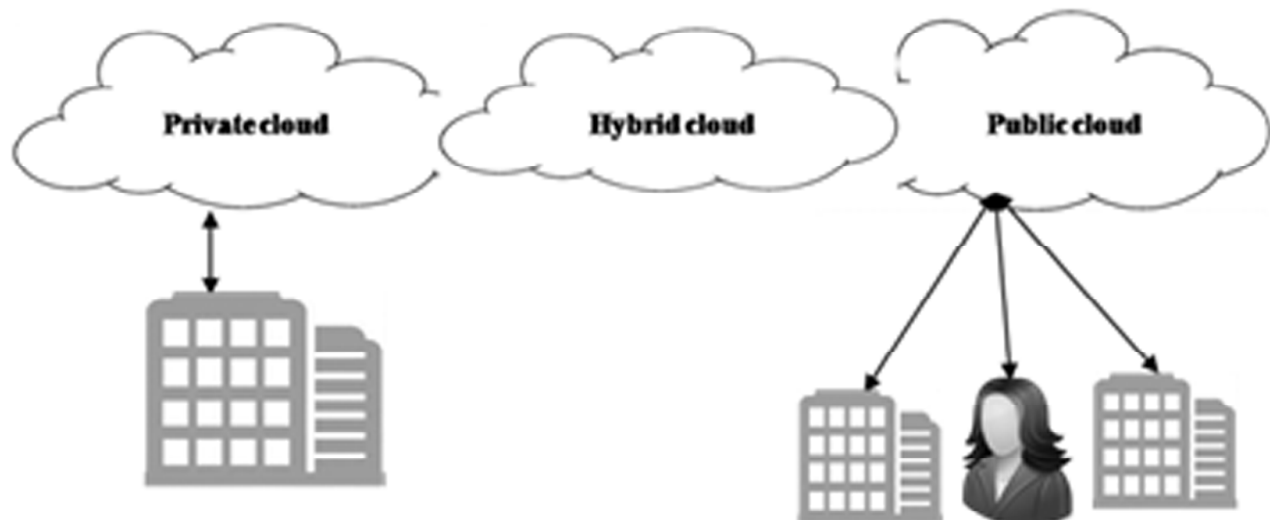


Figure 3: Types of Cloud Computing.

- Software as a Service(SaaS)
- Infrastructure as a Service (IaaS).

IaaS delivers the cloud computing infrastructure such as servers, storage, network and operating system. IaaS can be a private or public or combination of both. It generally includes multiple users on single piece of hardware. Infrastructure as a Service provides you with the highest level of flexibility and management control. Leading vendors that provide Infrastructure as a service are Amazon EC2, Amazon S3, and Rackspace Cloud.

1.3.1. Platform as a Service (PaaS)

PaaS provides the computing platform where the web applications can be created quickly, simply and without any complexity of maintaining the software. Platform as a Service is mostly used in development process as it provides a framework which the developers can build upon to develop or customize the applications. Some of the examples of PaaS include Google App Engine, Microsoft Azure, and Salesforce.com.

1.3.2. Software as a Service (SaaS)

SaaS provides a complete product that has been managed by the service provider. It represents the largest cloud market as it can also provide the applications to run directly on web browsers without any installations or downloads. The integration between different pieces of software can be provided by Application Programming Interfaces (APIs). Most SaaS offerings are email, customer relationship management, and healthcare-related applications.

(Fig. 4.) depicts the service architecture of the cloud along with examples at each level of the service.

Examples of all three services with respect to Google’s offerings are:

- IaaS - Google Compute Engine
- PaaS - Google App Engine
- SaaS - Gmail, Google plus.

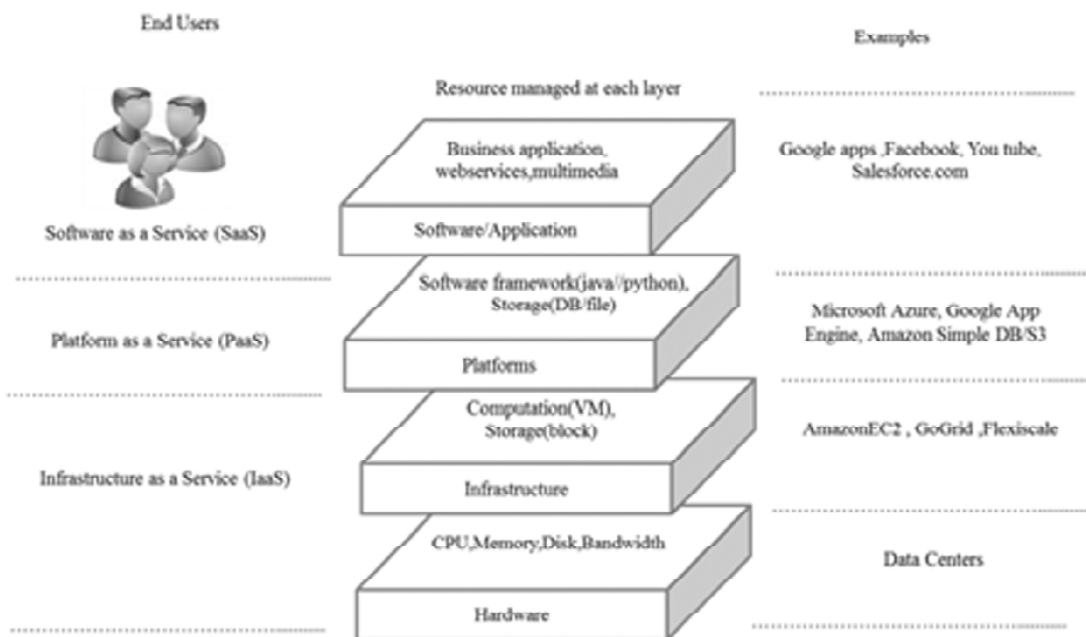


Figure 4: Levels of Services

Essential characteristics of cloud computing are on-demand, broad network access, resource pooling, measured service, rapid elasticity, etc [5].

On-demand self service. Without any interaction with humans the web based self service portals will provision the cloud computing resources for users. On demand self service providers are Amazon Web Services (AWS), Microsoft, Google, etc.

Broad Network Access. Cloud provides the capability over the network and accessed through standard mechanisms that supports heterogeneous thin or thick client platforms such as mobile phones, laptops and PDAs.

Resource Pooling. With same or different physical and virtual resources dynamically the providers computing resources are being pooled to multiple consumers. The resources include storage, processing, network and bandwidth.

Measured Service. Cloud systems automatically control and optimize resources by a metering capability appropriate to type of service (storage, processing, bandwidth). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

Rapid Elasticity. Cloud services can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in.

In section two, background study will be discussed. In section three the proposed architecture with the OpenStack services and functionality is specified.

2. BACKGROUND STUDY

Vikas Goyal. 2012 discusses how cloud computing architecture have provided the services i.e. web services through layers of cloud computing architecture and development model of cloud computing architecture [1]. Rashmi et al. 2012 discuss about the Cloud offers like dynamic scaling, high availability, multi-tenancy and effective resource allocation and the challenges for migration include business factors, technical factors and ease of implementation. A Five Phased Waterfall Model is proposed which is based on the iterative waterfall model from software development life cycle [2]. Rahul Bhojar, Nitin Chopde 2013 explores about different service models (IaaS, PaaS, and SaaS), cloud computing types (public, private, community) and advantage and usage of cloud. They also discuss the cloud database including deployment model, characteristics and identifies some technological and legal issues in cloud computing [3].

Sonali Yadav 2013 introduces the cloud computing technology and its basic concepts. The popularity of the private cloud and its deployment is explained. The author describes various open source software's to deploy private clouds such as Eucalyptus, OpenStack and OpenNebula. A comparative study is made about these three open source software's based on their architectures. Cloud implementation, programming language, and database compatibility, OS compatibility of these software's are mentioned [4].

Girish L S1, Dr. H S Guruprasad 2014 discuss about implementing a private cloud using open source software and operating system. 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, Swift, Keystone and Glance, Keystone and Horizon [6]. So to establish the IaaS, open source tools are used [5].

2.1. Comparative Study of Open Source Tools for IaaS

The comparative study of the open source tools [4] for IaaS is listed in [Table 1]. OpenStack is more advantageous by comparing the open source tools of IaaS. OpenStack has backing of 159 companies and more than 2800 people actively contributing to its code base.

Table 1
Open Source Tools-IaaS.

	OpenStack	Cloud Stack	Open Nebula	Eucalyptus
Established	2010	2010	2008	2008
Code base	Python	Java	Java, Ruby, C++	Java, C, python
Architecture	Integration of controller, compute, network	Hierarchical Management server, pod, zone, nodes	Centralized 3 components minimum 2 servers	Hierarchical 5 components minimum 2 servers
Licensing	Apache License V2	Apache License V2	Apache License V2	BSD License
OS	Linux, Windows	Depends on hypervisor and h/w	Few Linux distributions	Some Linux distributions
Used by	NASA	Open minds	Cumulus Project	UEC
Hypervisors	KVM, Xen, ESX, VMware, QEMU, etc	VMware, KVM, Xen, Oracle VM	KVM, Xen, ESXi, VMware ESX	KVM, Xen, VMware
Cloud types	Public, private, hybrid	Public, private, hybrid	private, hybrid	private, hybrid
UI	Web interface (Dashboard), CLI	Web interface, CLI	Web interface, CLI	Euca2ools (CLI)
Security	API, nova and keystone protects from DOS attacks and provides identity	Contains Cloud Stack security group 5	Authentication is provided by passwords, RSA key code pairs	Cloud controller provides public/private key pair for user authentication
Error Robustness	Reproduction	Reproduction	Database backend (permanent storage)	Separate clusters reduce correlated errors
Load Balancing	Cloud Controller	TCP Load Balancer	Nginx	Cloud Controller

The document support, ease of use is more in OpenStack compared with other tools. The present release of OpenStack is OpenStack-liberty.

3. OPENSTACK ARCHITECTURE

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a data centre, managed through dashboard [6]. OpenStack architecture mainly consists of

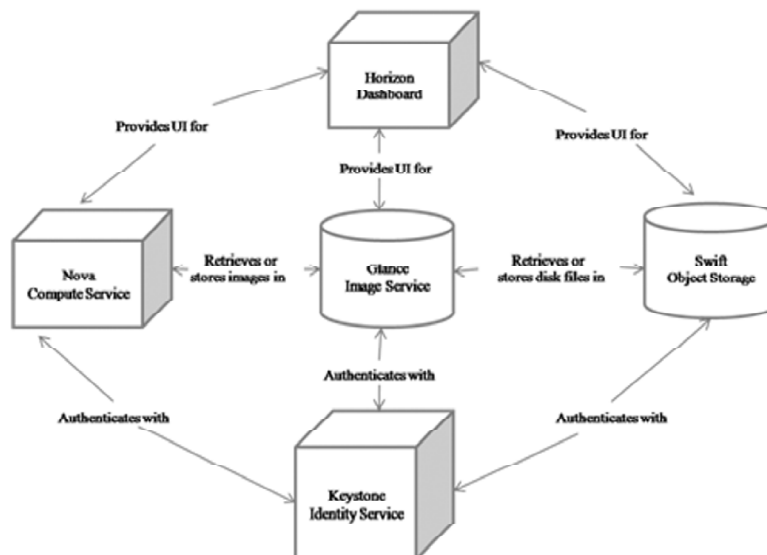


Figure 5: OpenStack architecture

Table 2
Components of OpenStack Liberty

Service	Code Name	Usage
Identity Service	Keystone	Provides Authentication
Compute Service	Nova	Provides Virtual Machine Management
Image Service	Glance	Manages virtual image
Dashboard	Horizon	Provides GUI
Object Storage	Swift	Provides cloud storage
Block Storage	Cinder	Storage management for virtual machine
Network Service	Neutron	Management of virtual Networking
Orchestration Service	Heat	orchestration function for virtual machine
Metering Service	Ceilometer	Usage measurement for accounting
Database Service	Trove	Management of database resource
Data Processing Service	Sahara	Provides data processing service
Bare Metal Provisioning	Ironic	Provides bare metal provisioning function
Messaging Service	Zaqar	Provides Messaging service function
Shared File System	Manila	Provides file sharing service
DNS Service	Designate	Provides DNS server service
Key Manager Service	Barbican	Provides key management service

compute, glance, and object as shown in [Fig. 5.]. Each component is authenticated using identity service (keystone) and the dashboard (horizon) provides the user interface. Compute (nova) controls the infrastructure and it allows managing large networks of virtual machines. It includes instance management for servers, networks and access control. Nova consists of API which acts as the web front end service control of the hypervisor. Image service (glance) provides the storage service, recording and distribution of images to virtual machine disks. Object storage (Swift) is used to create long term storage space for large volumes. It uses a distributed architecture with multiple access points to avoid single point of failure. The other components of the OpenStack-liberty are listed in the [Table 2.] along with the services [7] that specific component provides.

3.1. OpenStack Functionality

The main intention of the OpenStack is to provide scalable resources to the users. For providing the resources like storage, networking and servers Infrastructure as a Service should be established. So to establish IaaS following steps should be followed

- Install the latest version of OpenStack (liberty) along with required dependencies.
- Creating basic environment (controller, compute, network)
- Installation of OpenStack components (Keystone, glance, nova, etc)
- Providing authentication for projects and users

Launching static instances (running VM's).

The request flow of launching instance is depicted in [Fig.6.]. The following are the steps involved in it.

1. Dashboard or CLI gets the user credential and does the REST call to Keystone for authentication.
2. Keystone authenticates the credentials and generates and sends back auth-token which will be used for sending request to other Components through REST-call.

- Before launching instance create an image in glance which is used while launching instance.
- Choose the availability zone, some may have default zones and for some cases nova is the availability one.
- Assign a name for instance [virtual machine].
- Choose the appropriate flavor which are available or create a new flavor according to requirement [size of the instance].
- Choose the instance count to be launched at a time. By default it is 1.
- Choose the boot source of the instance (boot from image, snapshot, volume). At the first instance generally boot from image is selected.
- Based on the requirement provide key pair and security group
- Select the type of the network; if no networks are available create a network for the instance.
- Launch the instance

From the next step the instance starts to build. Different tasks in building state are:

- Scheduling task
- Networking task
- Block device mapping
- Spawing task

3.1.1. Scheduling task

3. Dashboard or CLI convert the new instance request specified in ‘launch instance’ or ‘nova-boot’ form to REST API request and send it to nova-api.
4. Nova-api receives the request and sends the request for validation of auth-token and access permission from keystone.
5. Keystone validates the token and sends updated auth headers with roles and permissions.
6. Nova-api interacts with nova-database.
7. Creates initial database entry for new instance.
8. Nova-api sends the *rpc.call* request to nova-scheduler expecting to get updated instance entry with host ID specified.
9. Nova-scheduler picks the request from the queue. [Message Queue-which handles the internal communication with OpenStack components such as nova, neutron].
10. Nova-scheduler interacts with nova-database to find an appropriate host via filtering and weighing.
11. Returns the updated instance entry with appropriate host ID after filtering and weighing.
12. Nova-scheduler sends the *rpc.cast* request to nova-compute for ‘launching instance’ on appropriate host.
13. Nova-compute picks the request from the queue.
14. Nova-compute send the *rpc.call* request to nova-conductor to fetch the instance information such as host ID and flavor (Ram, CPU, Disk).

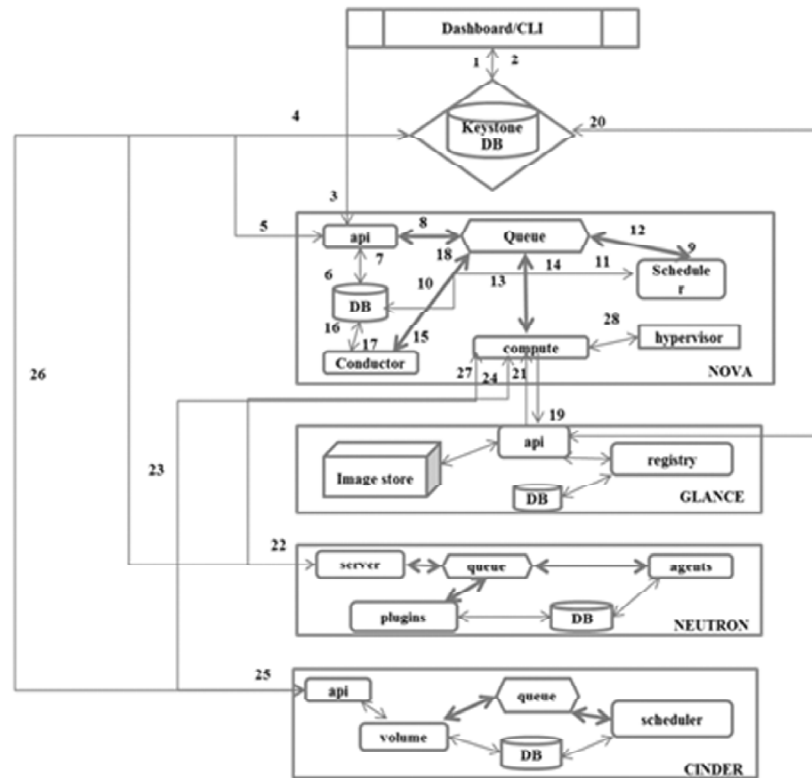


Figure 6: Flow Diagram to Create Instance

15. Nova-conductor picks the request from the queue.
16. Nova-conductor interacts with nova-database.
17. Return the instance information.
18. Nova-compute picks the instance information from the queue.
19. Nova-compute does the REST call by passing auth-token to glance-api to get the Image URI by Image ID from glance and upload image from image storage.
20. Glance-api validates the auth-token with keystone.
21. Nova-compute gets the image metadata.

3.1.2. Networking Task

22. Nova-compute does the REST-call by passing auth-token to Network API to allocate and configure the network such that instance gets the IP address.
23. Neutron-server validates the auth-token with keystone.
24. Nova-compute gets the network info.

3.1.3. Block Device Mapping

25. Nova-compute does the REST call by passing auth-token to Volume API to attach volumes to instance.
26. Cinder-api validates the auth-token with keystone.
27. Nova-compute gets the block storage info.

3.1.4. Spawing Task

28. Nova-compute generates data for hypervisor driver and executes request on hypervisor (via libvirt or API).
 - After completing all the tasks the instance comes to active state from build state.
 - Now the instance is accessed using the console and accordingly applications can be deployed.

4 CONCLUSION

This paper is focused towards provision of IaaS service. Open stack tool was used for configuring cloud infrastructure. Static instance was created for clients, which provides an environment to use the resources. Comparative analysis also done on open source tools to manage the framework, required for data tasks. It is observed that open stack provides live migration in load balancing that happens from virtual machines when there are more requests for resources. Moreover the users while operating open stack environment are authenticated. This paper can be extendable by means of dynamic instances with IaaS.

REFERENCES

- [1] Vikas Goyal: Layered architecture of Cloud Computing. International Journal of Computing & Business Research, ISSN 2229-6166 (2012).
- [2] Rashmi, Dr. Shabana, Mehruz, and Dr. G. Sahoo: A five phased approach for the cloud migration. International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459 (2012).
- [3] Rahul Bhoyar, Nitin Chopde: Cloud Computing: Service models, Types, Database and issues. International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 3, (2013).
- [4] Sonali Yadav: Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, OpenStack and OpenNebula. International Journal of Engineering and Science, ISSN (e): 2278-4721, ISSN (p):2319-6483 Volume.3, Issue 10 pp. 51-54, (2013).
- [5] Aniruddha S. Rumale, D.N. Chaudhari: Cloud Computing: Infrastructure as a Service. International Journal of Inventive Engineering and Sciences ISSN: 2319-9598, Volume-1, Issue-3, (2013).
- [6] Girish LS1, Dr. H. S. Guruprasad: Building Private Cloud using OpenStack. International Journal of Emerging Trends & Technology in Computer Science ISSN 2278-6856 Volume 3, Issue 3, (2014)
- [7] Venkatesan Balu, Dr. L. Josephine Mary: A Model of Security Architecture on Private Cloud Using OpenStack. International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 2 pp.587 – 590, (2014)