Heat Template implementation in OpenStack

[1] **D. Srinivasa Rao**

[1]Assosiate Professor, *Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad-500090, Telangana, India,*

Abstract: is a free and open-source software platform for cloud computing, mostly deployed as an infrastructure-as-a-service (IaaS). The software platform consists of interrelated components that control hardware pools of processing, storage, and networking resources throughout a data centre. Users either manage it through a web-based dashboard, through command-line tools, or through a RESTful API. The OpenStack scheme is an open source cloud computing platform for all types of clouds, which aims to be simple to implement, massively scalable, and feature rich. Developers and cloud computing technologists from around the world create the OpenStack scheme.OpenStack provides an Infrastructure-as-a-Service (*IaaS*) solution through a set of interrelated services. Each service offers an application programming interface (*API*) that facilitates this integration. Depending on your needs, you can install some or all services.

Heat is the main scheme in the OpenStack Orchestration program. It implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code. A native Heat template format is evolving, but Heat also endeavours to provide compatibility with the AWS Cloud Formation template format, so that many existing Cloud Formation templates can be launched on OpenStack. Heat provides both an OpenStack-native ReST API and a Cloud Formation-compatible Query API

IndexTerms **– OpenStack, heat, cloud computing**

1. INTRODUCTION

1.1 Cloud Computing:

Cloud computing is a recently developing paradigm of distributed computing. The birth of cloud computing is very recent phenomena although its root belongs to some old ideas with new business, technical and social perspectives. Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. Cloud computing, or in simpler shorthand "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. From the architectural point of view cloud is naturally build on an existing grid based architecture and uses the grid services and adds some technologies like virtualization and some business models.

2. Literature Survey

2.1 Openstack:

OpenStack is a free and open-source software platform for cloud computing, mostly deployed as an infrastructure-as-a-service (IaaS). The software platform consists of interrelated components that control hardware pools of processing, storage, and networking resources throughout a data centre. OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds. Backed by some of the biggest companies in software development and hosting, as well as thousands of individual community members, many think that OpenStack is the future of cloud computing.

3. Openstack Architecture

The openstack scheme is an open source cloud computing policy that chains all types of cloud surroundings. The development goals for simple application, huge scalability, and a rich set of structures. Cloud computing specialists from around the world subsidise to the scheme.

OpenStack delivers an Infrastructure-as-a-Service (IaaS) clarification through a multiplicity of perfected services. Each facility proposals an application programming interface (API) that enables this combination.

3.1 Conceptual Architecture

Launching a virtual machine or instance includes many interfaces among a number of services. The succeeding diagram delivers the theoretical construction of a typical OpenStack environment.

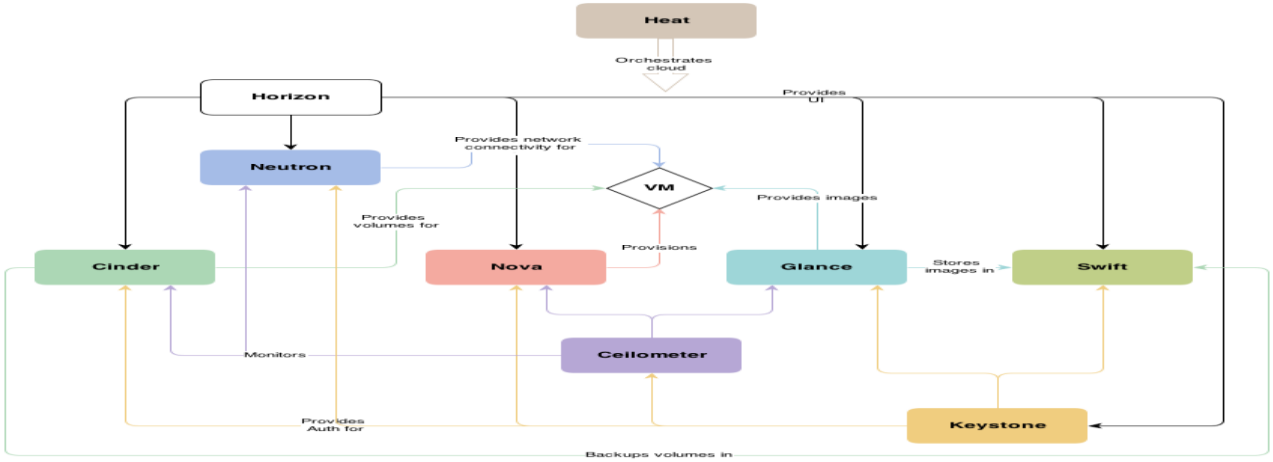


Figure 3.1: Conceptual architecture of OpenStack

3.2 Heat

Heat is a service to orchestrate composite cloud applications using a declarative template format through an OpenStack-native REST API. Orchestration is for infrastructure essentially what configuration management is for software applications. Instead of manipulating elements of your (virtual) infrastructure by hand, or with scripts, Heat allows you to work with a declarative model that represents the infrastructure resources and the relationships between them directly. Heat itself then works out the correct sequence of actions to perform to bring reality in line with your model. The model takes the form of a Heat template, and the resulting collection of infrastructure resources is known as a stack. Orchestration enables you to treat your infrastructure like code—you can store the template in a version control system to track changes to it. When you do make changes, just update the stack with the new template and Heat performs the necessary actions. The main interface to Heat is an OpenStack-native Rest API. As shown in Figure, Heat sits between the user and the APIs of the core OpenStack services in much the same way that the Dashboard (Horizon) does. If you prefer, you can also interact with Heat through the Dashboard.

Increasingly, Heat resource types are native representations of the underlying OpenStack Rest APIs, so you can access the full power and flexibility of OpenStack. However Heat also provides a fair degree of compatibility with CloudFormation, its functional equivalent in Amazon Web Services.1 If you have workloads running on AWS that are defined with Cloud Formation templates, it may be possible to move them to OpenStack with few or no changes in many cases. Heat even has a Cloud Formation compatibility API.

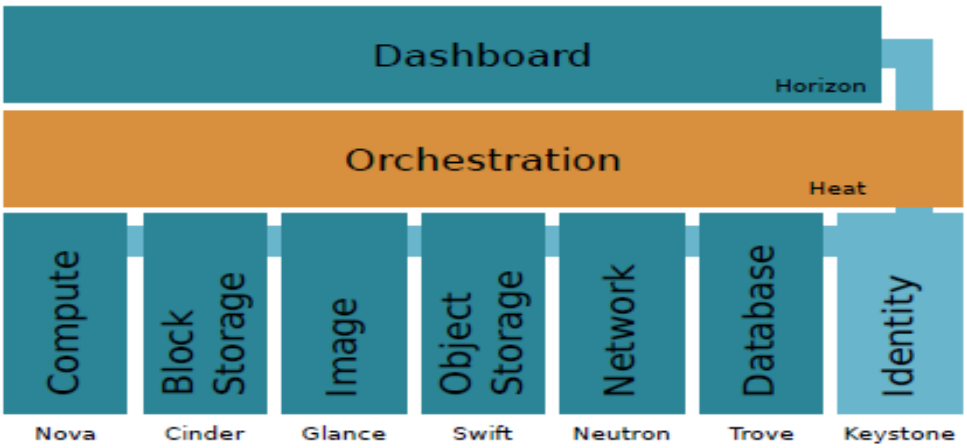


Figure 3.2: OpenStack architecture from a user's perspective, showing how Orchestration interacts with other services.

3.3 Heat Architecture

Heat is a facility to orchestrate multiple composite cloud applications using the AWS Cloud Formation prototype set-up, through both an OpenStack-native REST API and a Cloud Formation-compatible Query API. Heat delivers an AWS Cloud Formation application for OpenStack that orchestrates an AWS Cloud Formation template relating a cloud submission by performing suitable OpenStack API calls to generate running cloud applications.

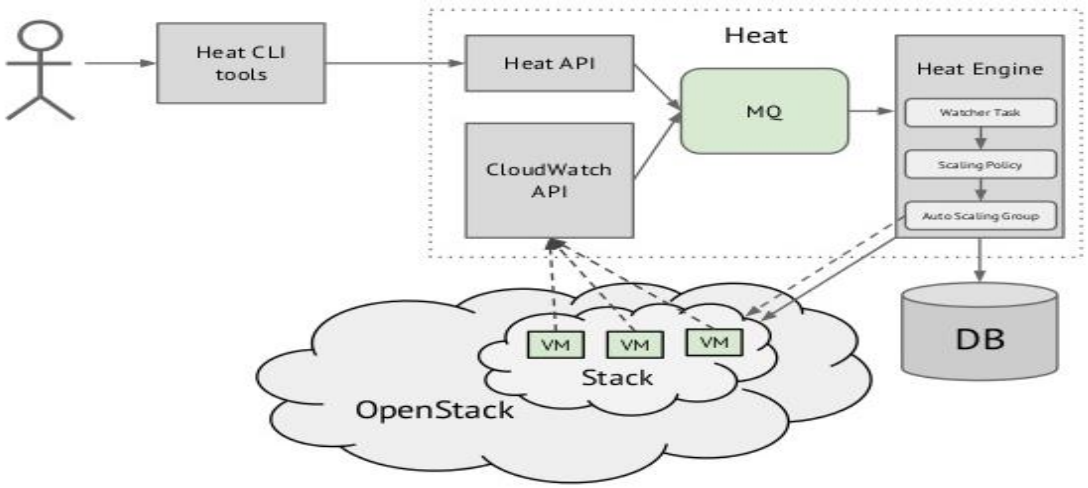


Figure 3.3: Heat Architecture

4. Implementation And Results

4.1 Code For Local Configuration File

```
[[local|localrc]]
disable_all_services
# Core compute (glance+keystone+nova+vnc)
enable_service g-api g-reg key n-api n-crt n-obj n-cpu n-cond n-sch n-novnc n-xvnc n-cauth # n-net
#enable_service c-sch,c-api,c-vol
#enable heat services
ENABLED_SERVICES+=,heat,h-api,h-api-cfn,h-api-cw,h-eng
# Enable the ceilometer metering services
enable_service ceilometer-a compute ceilometer-a central ceilometer-a notification ceilometer-collector
# Enable the ceilometer alarming services
enable_service ceilometer-alarm-evaluator, ceilometer-alarm-notifier
# Enable the ceilometer api services
enable_service ceilometer-api
#enable_service horizon
disable_service tempest
#enable_service cinder c-api c-sch c-vol
#enable_service s-proxy s-object s-container s-account
# dashboard
enable_service horizon
#network
disable_service n-net
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service neutron
# additional services
enable_service mysql rabbit
HOST_IP=127.0.0.1
SERVICE_HOST=$HOST_IP
IMAGE_HOST=$HOST_IP
IDENTITY_HOST=$HOST_IP
#SCHEDULER=nova.scheduler.simple.SimpleScheduler
MULTI_HOST=1
export no_proxy="localhost,127.0.0.1"
#MYSQL_HOST=$SERVICE_HOST
#RABBIT_HOST=$SERVICE_HOST
#GLANCE_HOSTPORT=$SERVICE_HOST:9292
#KEYSTONE_AUTH_HOST=$SERVICE_HOST
#KEYSTONE_SERVICE_HOST=$SERVICE_HOST

MYSQL_PASSWORD=openstack
RABBIT_PASSWORD=openstack
```

```
SERVICE_TOKEN=openstack
SERVICE_PASSWORD=openstack
ADMIN_PASSWORD=openstack

GLANCE_BRANCH=stable/kilo
HORIZON_BRANCH=stable/kilo
KEYSTONE_BRANCH=stable/kilo
NOVA_BRANCH=stable/kilo
NEUTRON_BRANCH=stable/kilo
SWIFT_BRANCH=stable/kilo
CLIFF_BRANCH=stable/kilo
TEMPEST_BRANCH=stable/kilo
CINDER_BRANCH=stable/kilo
HEAT_BRANCH=stable/kilo
TROVE_BRANCH=stable/kilo
CEILOMETER_BRANCH=stable/kilo
```

4.2 Code For Heat Template:

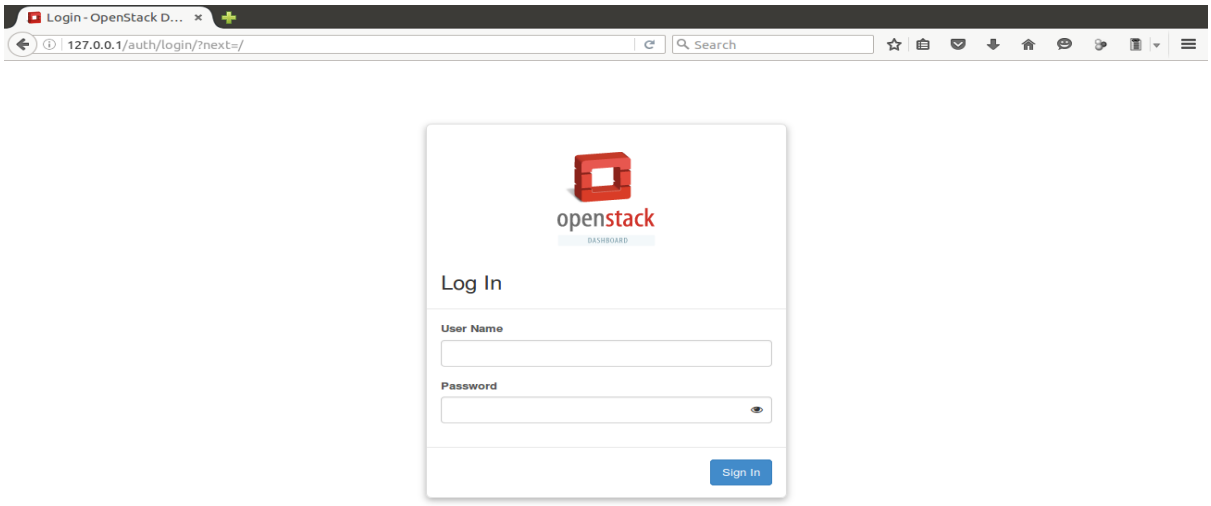
```
heat_template_version: 2015-04-30
description: Simple template to deploy a single compute instance
resources:
        new_net:
                type: OS::Neutron::Net
         new_subnet:
                type: OS::Neutron::Subnet
                properties:
                        network_id: { get_resource: new_net }
                        cidr: "10.8.1.0/24"
                        ip_version: 4
         my_instance:
                 type: OS::Nova::Server
                properties:
                        key_name: heat_key
                        image: cirros-0.3.4-x86_64-uec
                        flavor: m1.small
                        networks:
                                - network: { get_resource: new_net }
 outputs:
        instance_ip:
                description: IP address of the instance
                value: { get_attr: [my_instance, first_address] }
```

4.3 Code For Userdata

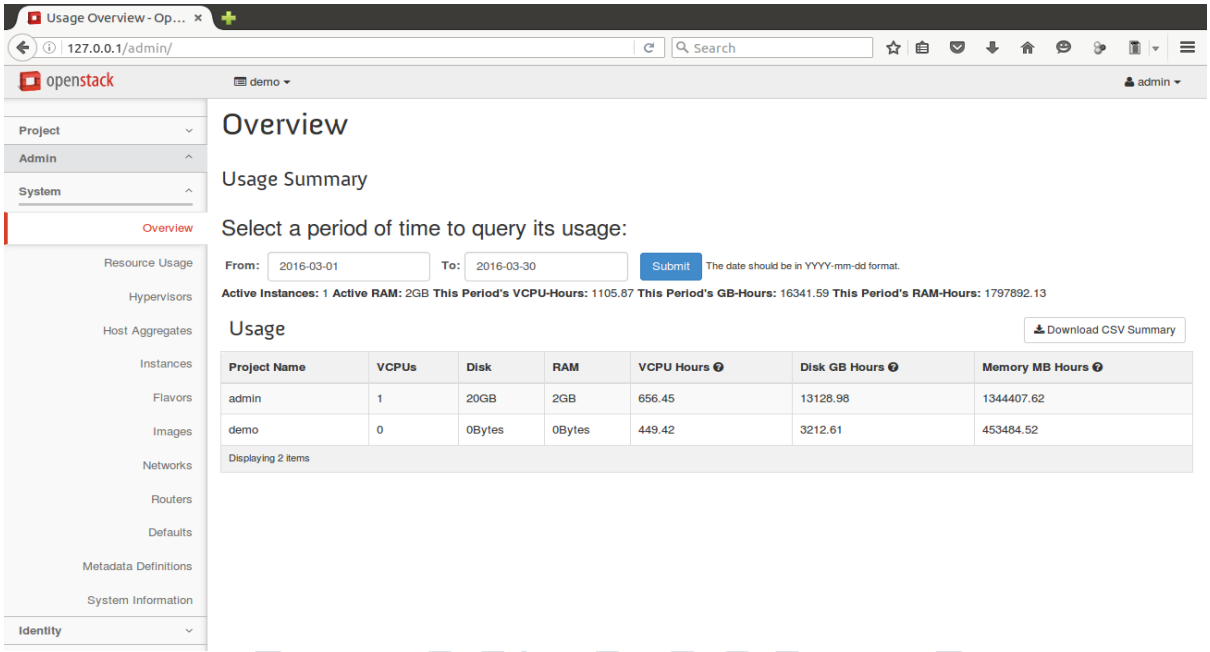
```
user_data:   |
        #!/bin/sh
        mkdir test
        echo "hello, world!!"
        user_data_format:  RAM
```

4.4 Sample Screenshots Of Dashboard

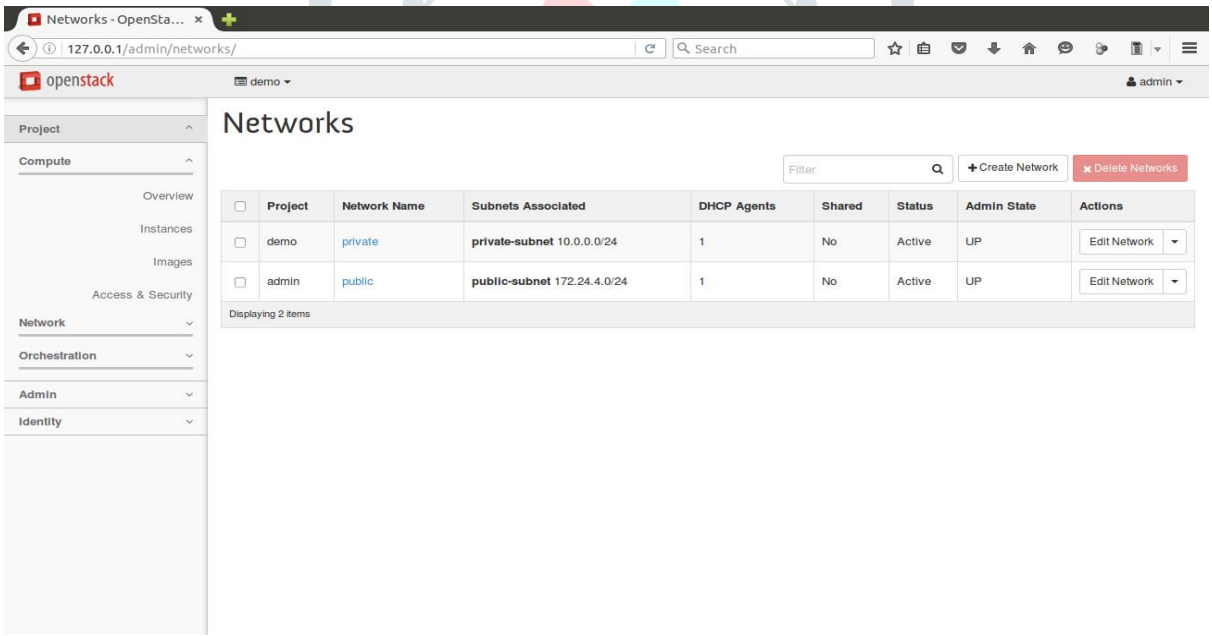
Login Page:



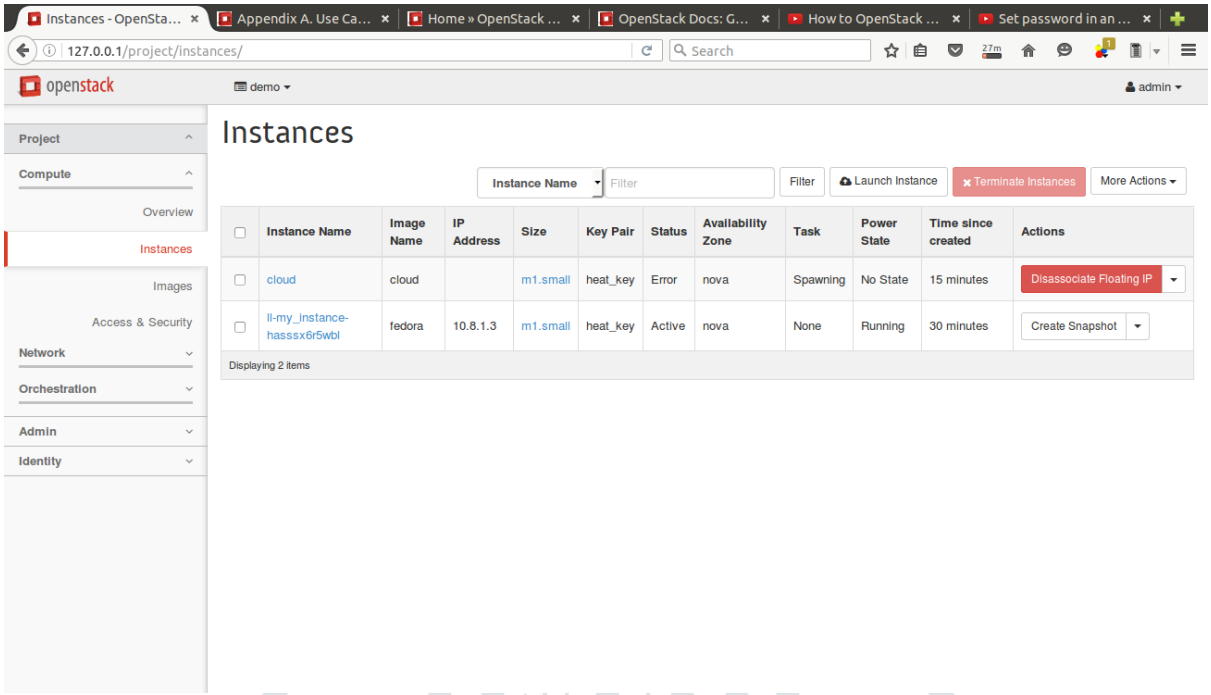
Dashboard Overview:



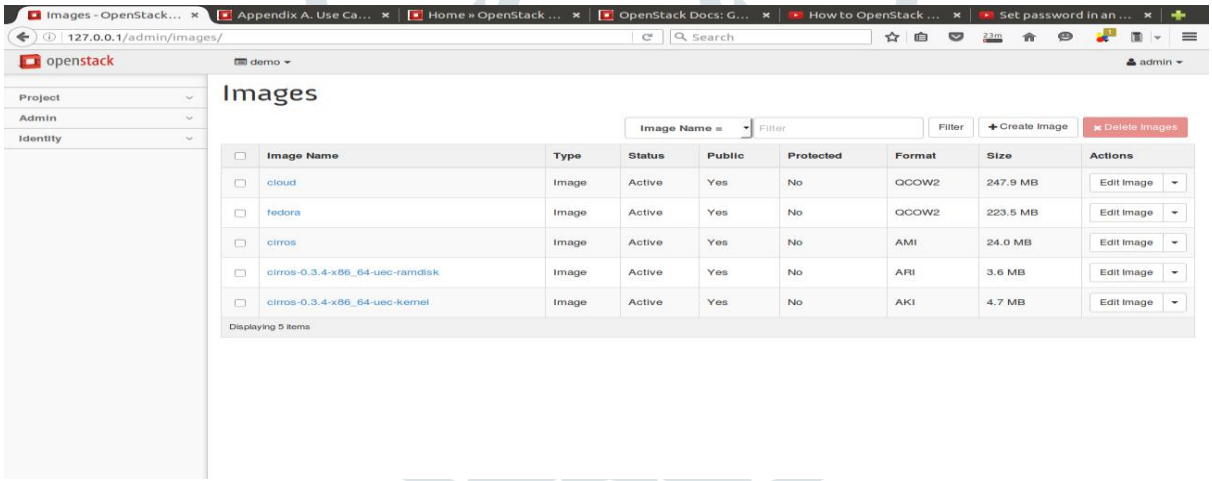
Network Creation:



Instance Creation:



Launching Images:



4.5 Sample Screenshots Of Command Line Interface:

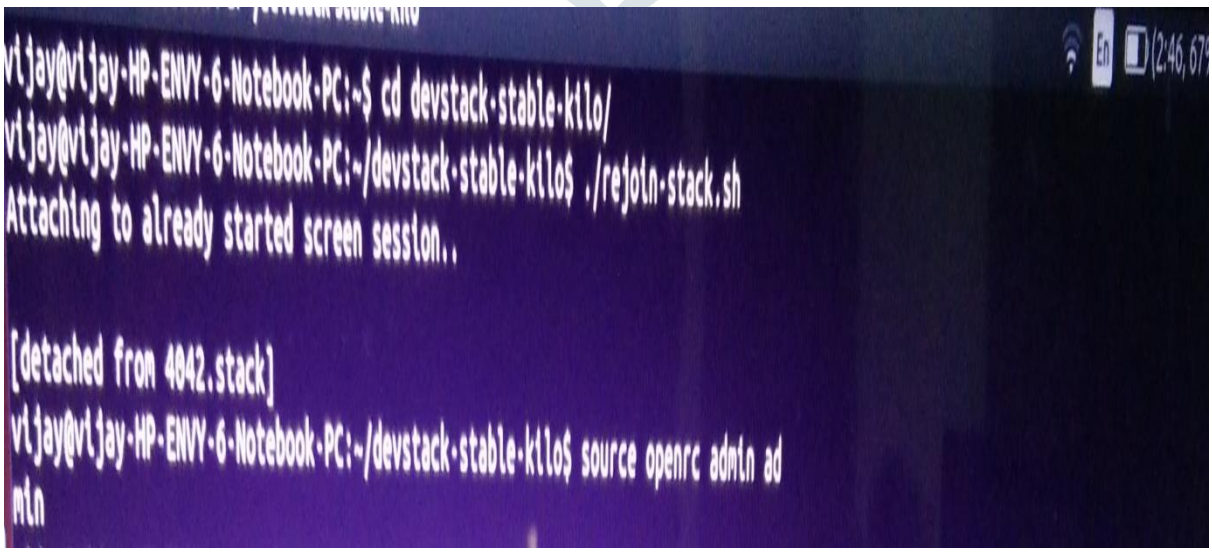


Fig 4.1: Authentication of user

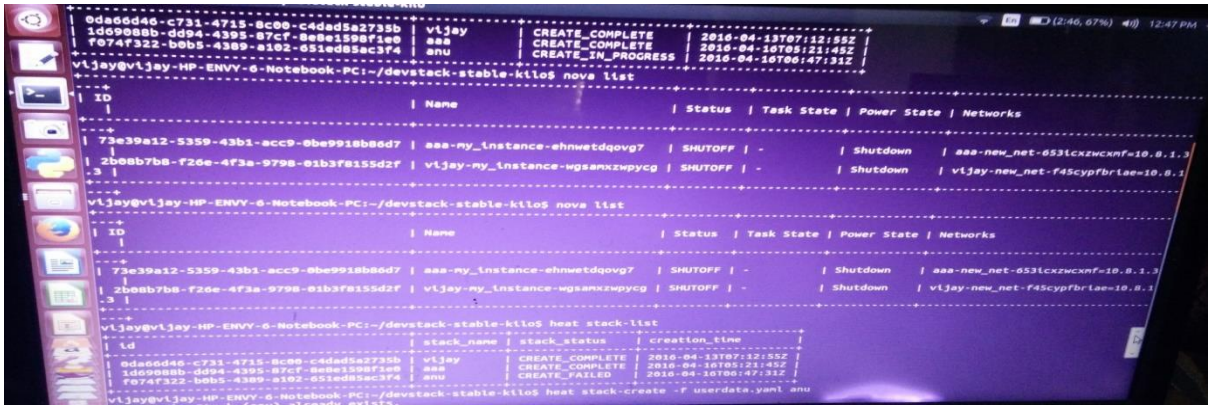


Fig 4.2: Status of virtual machine

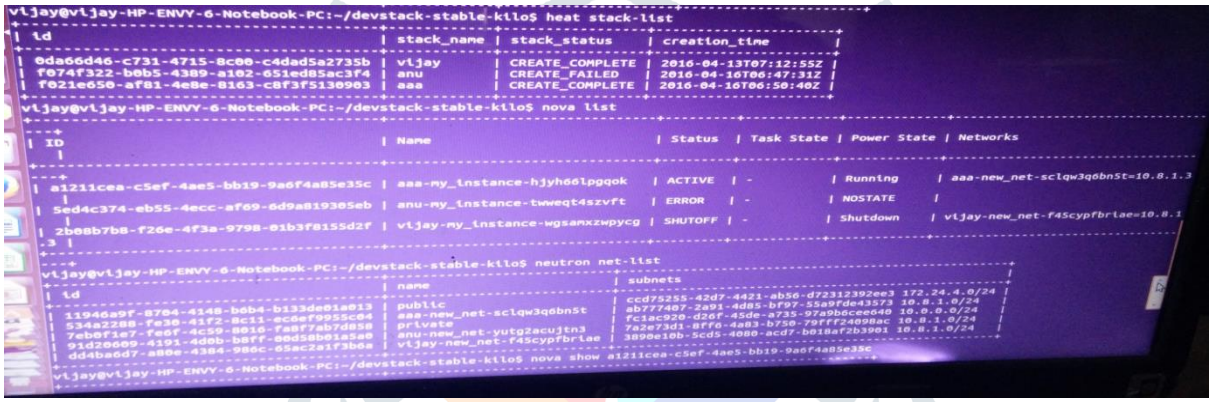


Fig 4.3: List of networks

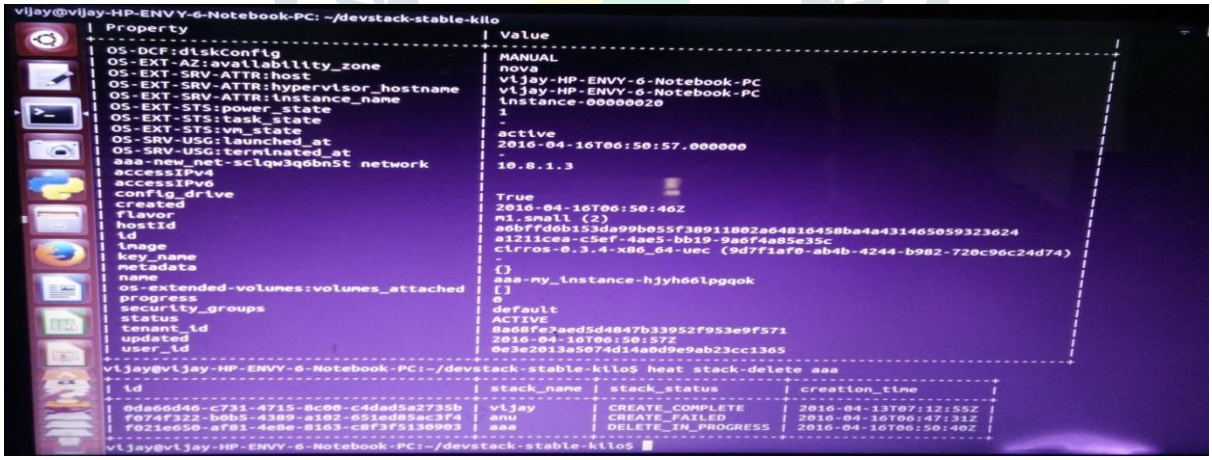


Fig 4.4: Characteristics of virtual machine

5. Conclusion:

One of the primary yields competing for a place at the cloud computing table is OpenStack, a system that has formed a group of tools for restructuring the construction of Infrastructure as Service platforms. Amid the plethora of cloud software arrangements contending for consideration, emerging an understanding of the determination and comparative assets of each is not modest. The potential that IaaS and public and private clouds fetch business has led to the formation of multiple resolutions for the positioning and organisation of cloud substructure. OpenStack users shouldn't need to worry about the details of the underlying hardware; the virtualization layer manages the physical resources and makes them available at a higher level of abstraction. Essentially, users can just ask for a virtual machine and not have to worry about what's going on beneath the virtualization layer.

OpenStack's APIs are intended to be companionable with Amazon Web Services, the utmost current open cloud platform. For businesses, that means the procedure of porting IaaS client presentations from AWS to OpenStack-based IaaS providers needs minimal exertion. Lastly, by doing this arrangement we attain the directory structure within the VM with the help of user-data which can be viewed after logging into VM.

6. REFERENCES:

https://github.com/openstack/heat
https://github.com/openstack/heat-templates
http://docs.openstack.org/developer/heat
http://wiki.openstack.org/Heat/
http://openstack.redhat.com
https://opensource.com/resources/what-is-openstack
https://www.rackspace.com/cloud/openstack
https://access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/7/architecture-guide/
http://docs.openstack.org/liberty/install-guide-ubuntu/
http://www.openstack.org/software/