**Imperial Journal of Interdisciplinary Research (IJIR)**
**Vol-2, Issue-5, 2016**
**ISSN: 2454-1362,** http://www.onlinejournal.in

# Automation in Openstack using Heat Templates

Shreyansh Jarged[1], Bhuvnesh Purohit[2] & Tushar Sharma[3]
Computer Science and Engineering, Poornima Group of Institutions, Jaipur, Rajasthan

***Abstract:*** *Cloud computing seems to be newfangled technology. Almost all the major IT companies has moved onto cloud because of its numerous advantages downtime. As cloud promises the minimal or zero down the productivity of an organization. time to the users hence the need of automation is inevitable. The aim of this paper is to endow readers to introduce with the functionality of Heat component and how heat templates could be helpful in increasing over standard IT infrastructure. One among the varied advantages of cloud is it guarantees a minimal*

***Keywords:*** *OpenStack, Keystone, Glance, Heat, Neutron, Cinder, swift*

## 1. Introduction

Cloud computing using Openstack is a relatively newborn field of endeavor in Computer science which deals with all major aspects of computers. Cloud computing technologies such as visualization, Cloud Storage, networking .The aim for which the cloud has been created is the optimal utilization of resources. Openstack is a platform which is used to deploy the cloud both public and private. Its a joint venture of NASA and Rackspace which contributed to compute and storage services respectively.

Openstack is a ubiquitous Open Source Cloud Computing platform which is actually collection of software tools which are required to deploy cloud according to the need of an organization. It is managed by foundation, which is non-profit corporate entity established in September 2012.More than 300+ Companies like IBM, Intel, Cisco, GoDaddy, HP, Dell, VMware, Oracle.

Openstack community's first official release, namely Austin, appeared in October 2010. Then Bexar, Cactus, Diab lo, Essex, Folsom, Grizzly, Havana, Icehouse ,Juno and Kilo were released in succession. Openstack uses different components for the different services it offers. Though at the initiation of this project only Iaas (Infrastructure as a Service ) can only be offered to the clients .They can create as well as manage large group of virtual servers.

Components of Openstack :

- Keystone
- Glance
- Cinder
- Nova
- Neutron
- Horizon
- Heat
- Swift

1. **Keystone**:

It is used for authentication, authorization and catalog management. It's maintained a database for all the users and provides the catalog of services after authentication and validation of a user. It uses message broker like qpid to enhance the security of

### 2. Glance:

Image services include discovering, registering, and retrieving virtual machine images. Glance has a Restful API that allows querying of Virtual Machines.

If Images is in shared state it can be used by more than one tenants.

### 3. Cinder:

It serves as a block storage can be considered as a raw disk which can be used by client accordingly.

### 4. Nova:

Nova is an OpenStack project designed to provide power massively scalable, on demand, self-service access to compute resources.

It is also responsible for proper scheduling of instances on compute nodes.

### 5. Neutron:

Neutron is responsible for networking. Networking include creating networks both public and private , subnets ,association of floating Ip addresses, creating routers ,adding interfaces ,etc.

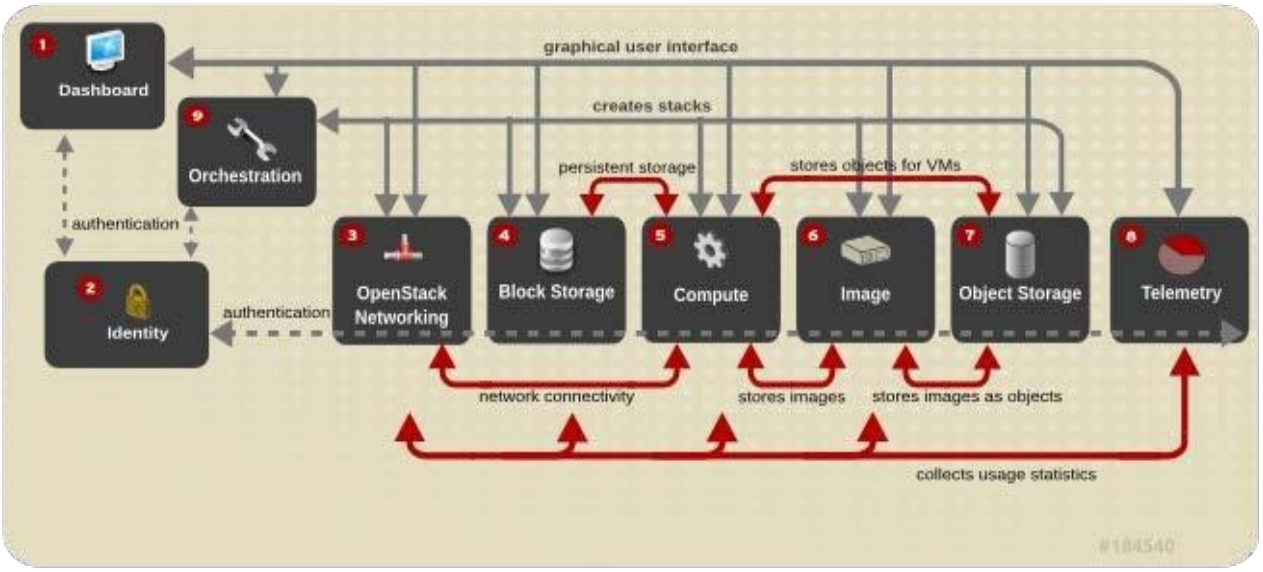
All of the above mentioned tasks are basically created virtually and hence called SDN i.e. Software Defined Networking.

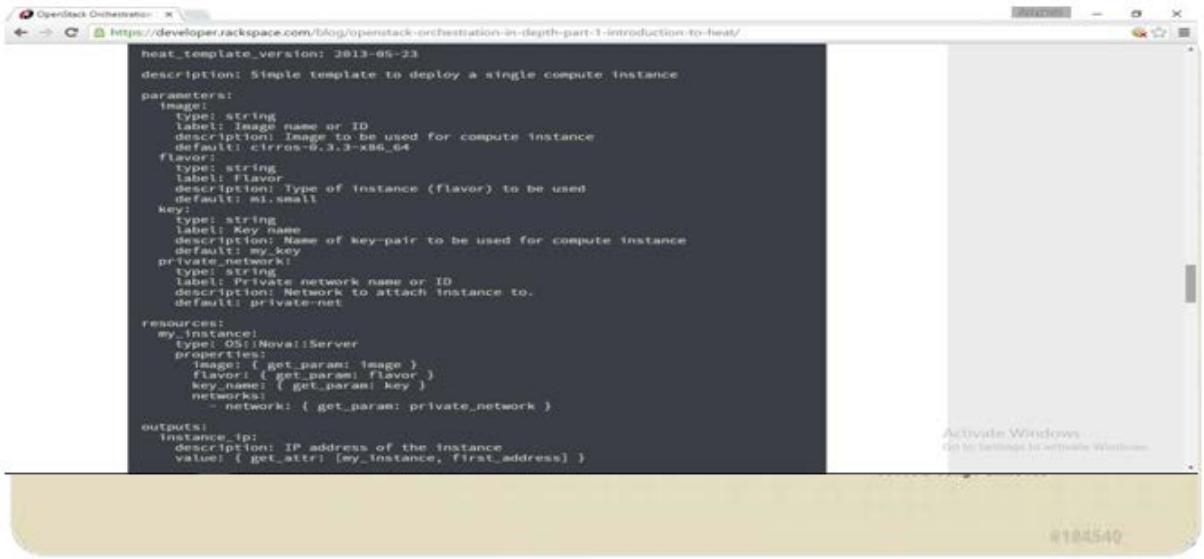
### 6. Horizon:

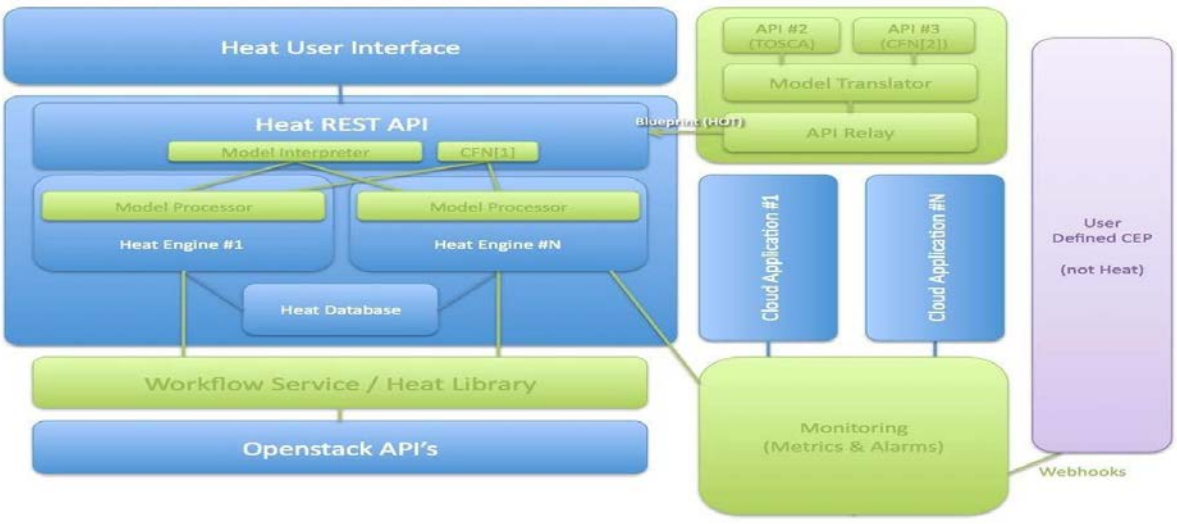
The heat tool is a command line interface which communicates with the heat-api to execute AWS Cloud Formation APIs. Heat is the main project in the Openstack Orchestration program.

### 8. Swift:

Provides objecr storage in openstack clouds.It's built for scale and optimized for durability, availability, and concurrency across the entire data set


graphical user interface
creates stacks
persistent storage
stores objects for VMs
Dashboard
Orchestration
authentication
Identity
authentication
OpenStack Networking
Block Storage
Compute
Image
Object Storage
Telemetry
network connectivity
stores images
stores images as objects
collects usage statistics




Heat User Interface
Heat REST API
Model Interpreter
CFN[1]
Model Processor
Heat Engine #1
Model Processor
Heat Engine #N
Heat Database
Workflow Service / Heat Library
Openstack API's
API #2 (TOSCA)
API #3 (CFN[2])
Model Translator
API Relay
Cloud Application #1
Cloud Application #N
User Defined CEP
(not Heat)
Monitoring (Metrics & Alarms)
Webhooks

## OpenStack Architecture

## Components of Heat

1. Heat engine :-It does the main work of orchestrating the launch of templates and management of events.
2. heat-api:-component provides an native API that processes API requests by sending them to the heat-engine over remote procedure call.
3. heat-api-cfn:-component provides an AWS-style Query API that is compatible with AWS Cloud Formation and processes API requests by sending them to heat engine Writing Heat Orchestration Template (HOT) Writing HOT is a very demanding task and changes as the scenario of an enterprise changes. We can understand various different segments of HOT using a snapshot.

### Sections in Heat Templates

1. Version: The mandatory part of template usually depicts the version in which this yaml format file is written.
2. Description: It's an optional and usually display the functionality of a particular template.
3. Parameter: Declaration of input parameter.
4. Outputs: defines what attributes of the stack to export after it is deployed.

### Resources:

This section defines different components that are used while deploying the template - In given example it is
OS: Nova::Server, which is the type of a Nova compute instance.
Stack: A group of connected cloud resources. Example:

Stacks are created from Templates. Integrated with chef and Puppet.
HOT: Short for Heat Orchestration Template, HOT is one of two template formats used by Heat. HOT is not backwards-compatible with AWS CloudFormation templates and can only be used with OpenStack. Templates in HOT format are typically—but not necessarily required to be—expressed as YAML.

1. **CFN:**

Short for AWS CloudFormation, this is the second template format that is supported by Heat. CFNformatted templates are typically expressed in JSON.
How Heat Works ?

- Heat template describes the desired cloud infrastructure in terms of human readable text file. - This infrastructure comprises of number of resource which include but not limited to servers, floating IPs,volumes, security groups, users, etc.
- In above example - The parameter section specifies the value of input that is necessary to deploy the Heat Template. - If user did not provide any value of any attribute of parameter section than the engine uses default value that has been provided by the template designer.
- Resource section according to the type mentioned will perform the series of steps in deploying templates and uses the value of parameters by "get_param" attribute.
- Output section will launch any instance on particular network with defined keypair and with providednetwork .

### Heat Orchestration Template (HOT) specification

HOT is a new template format meant to replace the Heat Cloud Formation-compatible format (CFN) as the native format supported by the Heat over time. This specification explains in detail all elements of the HOT template format. An example driven guide to writing HOT templates can be found at *Heat Orchestration Template (HOT) Guide*.
Status.
HOT is considered reliable, supported, and standardized as of our Icehouse (April 2014) release. The Heat core team may make improvements to the standard, which very likely would be backward compatible. The template format is also versioned. Since Juno release, Heat supports multiple different versions of the HOT specification.
Template structure.
HOT templates are defined in YAML and follow the structure outlined below.

```
heat_template_version: 2013-05-23

Description:
  # a description of the template

parameter_groups:
  # a declaration of input parameter groups and order

Parameters:
  # declaration of input parameters

Resources:
  # declaration of template resources

Outputs:
  # declaration of output parameters
```

heat_template_version
This key with value 2013-05-23 (or a later date) indicates that the YAML document is a HOT template of the specified version.
Description

This optional key allows for giving a description of the template, or the workload that can be deployed using the template.
parameter_groups
This section allows for specifying how the input parameters should be grouped and the order to provide the parameters in. This section is optional and can be omitted when necessary.
Parameters: This section allows for specifying input parameters that have to be provided when instantiating the template. The section is optional and can be omitted when no input is required.
Resources This section contains the declaration of the single resources of the template. This section with at least one resource should be defined in any HOT template, or the template would not really do anything when being instantiated.
Outputs : This section allows for specifying output parameters available to users once the template has been instantiated. This section is optional and can be omitted when no output values are required.

## Advantages of using Heat Templates

1. Reduces the work of system administrator, to monitor each and every instance that is being launched on a cloud which is nearly impossible. Heat Automation reduces the work of continue monitoring of instances and resources and hence reduces the work load of the administrator. Help to minimize the downtime and hence reduces loss which organization faces during downtime.
2. Enhances the reputation of the cloud hosting organization.

## Conclusion

In this paper we came across the various usage of heat templates , how to write a heat templates , how heat works , components of heat and its advantages .But its only superficial task that can be done using Heat Templates. Some organization uses heat to regulate day -to-day activities. Uses of heat templates reduces time for completion of task. Minimizes down time. Openstack is one of the open source platform to build private cloud. It is relative young as compared to other open source platform. This paper provides details of our Project Heat Template Generator and how it has an edge over traditional template writing. We also briefly discussed how it will benefit the group of users. While it certainly has few limitations- reduces redundant stuff7.

## Future Work :

With increasing trust and dependability on cloud computing in present era . The cloud platform has to be fully automated and should apply on-demand services to the user using it . Hence the template will include all the features that may be necessary in creating a fully automated cloud environment.

## References:

[1]Rackspace-https://developer.rackspace.com/ blog/openstack-orchestration-in-depth-part-

.[2] CSDN : http://blog.csdn.net/yxc135/article/

[3] details/20466333 [3] .Openstack Documentation : http://docs.openstack.org/developer/heat/ template_guide/hot_guide.html
[4] .Blog : http://blog.scottlowe.org/2014/05/01/anintroduction- to-openstack-heat/

[5] Template guide http://docs.openstack.org/developer/heat/template_guide/