High Availability Cloud Maps for Cloud System using HPE Serviceguard

[1]Varsha Aiyappa C, [2] Dr. Ramakanth Kumar P, [3]Mr. Gururaja Grandhi,[4]Mr. Srinivas Krishnappa Shapur

[1,2]Dept. of ISE , R V College of Engineering ,Bengaluru, India

[3,4]Hewlett Packard Enterprise ,Bengaluru, India

c.varsha.a@gmail.com , ramakanthkp@rvce.edu.in, gururaja.grandhi@hpe.com, srinivas-s.k@hpe.com

Abstract:***Sustaining High Availability of IAAS (Infrastructure as a Service) services at a Cluster level with judicious cost is a provoking task that received deliberation due to the enormous popularity of Cloud Computing as a reasonable means of IT outsourcing. However, this comes with an additional cost of creating instances in the virtualized environment. Hewlett Packard Enterprise (HPE) has an innovative solution with the introduction of Cloud Maps. Cloud maps- a scalable and parallel virtual machine spawning application that enables High Availability hallmark at a VM level in a data center. Cloud Maps are implemented and will be used in all Cloud platforms based on OpenStack technology as a high availability facet for efficient failover in zillions of data centers with gobs of servers, VMs and an abundant number of disks. The introduced Cloud Maps will reduce the time required for the creation of nodes in the data centers. This paper also focuses on how Cloud Maps helps one to provision clusters with the High Availability feature enabled in them and how these cluster nodes helps us to maintain continuity in our business services in case of a failure***.

Keywords: Cloud Computing, High Availability, Cloud Maps

I. INTRODUCTION

In the early years, cloud computing research was focused on building robust cloud infrastructure, but now Cloud computing is paving its way into the enterprise and has become a technology that has a greater impact on everyone's day to day lives. From a software engineering point of view, Cloud is a fancy term for the word "Internet", and this Cloud consists of the following: interrelated services constituting a package, then the individual units called tasks with the respective configuration files containing the necessary data. All the tasks can be scheduled on multiple instances spawned and they are accessible through application program interfaces (API's). It is observed that people evolve to cloud infrastructure at the pace which is comfortable to one's business but always with the goal of hosting more instances. It is also noticed that no matter what the speed of progress to cloud is, once we reach the position of resource pool readiness, transition from physical to cloud infrastructure to deliver applications as services has to be accomplished very quickly.

Since Cloud Computing promises agility with on demand provisioning of resources, pay per use, and scalability features, the hardship of maintaining or managing the actual resources is shifted drastically in this technological approach. Location, configuration and machine details of the instances provisioned in the cloud are no longer needed by the end-users, as long as their Service Level Agreements (SLA's) are met.

When contemplating Cloud computing, it is necessary for the end users to use cluster of nodes in the cloud. But there is always a question at the backend which asks "Why would anyone need a Cluster of Computers?", If the decision of the end user is to make use of clusters, how are they even provisioned with minimal time is another hard core inquest. Here is an explanation of why do we need clusters. "A computer cluster merelyprovides you extra processing power than you have with one computer alone. For quintessential applications, one computer can simply provide all the resources which is capable enough of sufficing the work. But for people working with mission critical applications, having cluster of instances with the same functionality and Infrastructural resources along with the High Availability feature is of utter most importance, where the availability and continuity of services are needed irrespective of hardware or software failures. So that in case of a failure, services running on the failing primary node can move to other active nodes in the cluster.

When deploying instances in the cloud and creating cluster out of the spawned instances, one can get wrapped up to see how quickly one can build and provision these pieces of work. But what really enumerates here, is how the fore services are built to deliver an Infrastructural resource. Since any deployment process is error prone, in order to eliminate the dispensable human intervention and to provide a flourished experience along with end to end services for customers, routine scenarios are automated utilizing the recipes, mainly found for deployment, and these highly optimized automated solutions are Cloud Maps. Cloud Maps avail customers outline, build and amble the best cloud environments suited to ones need-predicated on the requisite of the industries and workloads. Cloud Maps runs on a cloud system- A virtual private cloud at the box which contains all the software stack units, servers, networking and storage to get commenced. Now utilizing Cloud Maps, mission is simplified and it just takes few seconds to deploy a typical application.

II. EXISTING SYSTEM

Today creating cluster of nodes in the cloud and run applications services on the cluster nodes is a trivial process – it is long, cumbersome and complex, a manual process that is error prone and non-repeatable. An end to end process to be made available in the cloud - private or public can take up to 8 weeks to complete, this is clearly misaligned with the cloud's promise of agility. Many are unprepared for the difficulties and time delays they encounter when they try to provision their IT infrastructure in the process of designing and deploying clusters in the cloud. There are challenges awaiting in maintaining the network configurations, configuring storage, multiple IT planning meetings and coordination across teams in the virtualized environment. Once the clusters get deployed, in case of a node failure, all the resources and services running on the failed node will fall into disuse, as these nodes do not procure the High Availability feature enabled in them.

III. PROPOSED SYSTEM

Cloud Maps are considered to be the superlative solutions, prepackaged with the necessary logic that can kick off the entire IT process fairly automatically. These solutions are also termed as heat maps, which gets deployed on the Cloud platform hinged on the OpenStack technology. These Cloud Maps are consumed and processed by the Orchestration Application Program Interface of the Cloud platform, which in turn allows one to automate the creation and enablement of informative infrastructure with provisioning of resources at minimum time. For our proposed solution, these Cloud Maps will help the end user spawn instances with the product offering High Availability feature integrated in them. Cloud Maps will help one to provision cluster of nodes in minimum clicks avoiding the situation of going through entire IT process.

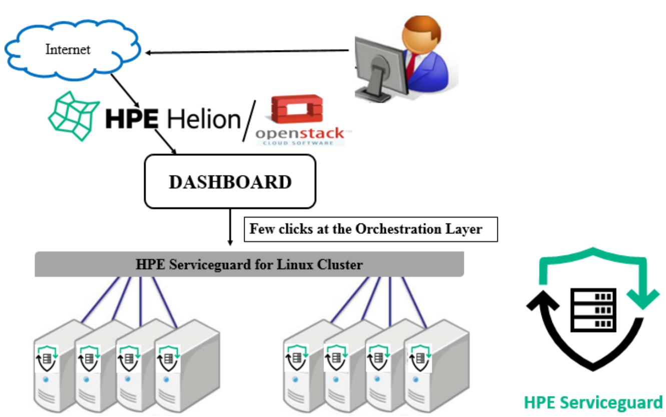

Figure 1: User view of the proposed system

IV. PREREQUISITES

A. Cloud platforms leveraging the OpenStack technology

OpenStack cloud software avails abstract and standardizes the involutions of managing heterogeneous cloud infrastructure. Many of the industry bellwethers have contributed code to OpenStack to enable their contrivances to work felicitously when called on by OpenStack. This makes it more facile to provision a virtual server, associate certain resources such assecurity and network configurations, a part of storage to that virtual server regardless of the underlying, vendor categorical infrastructure. Since Orchestration service is the core component of Cloud Maps, OpenStack Icehouse and succeeding flavors of OpenStack is recommended for the deployment of Cloud Maps.

To provide supremacy of Cloud Computing to customers, Hewlett Packard Enterprise (HPE) has introduced a Cloud platform designated as "HPE Helion" that suctions the thumbprints of OpenStack and Cloud Foundry. This Cloud platform provides a collective Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) offering for Cloud-native workloads with flexibility in software and hardware. Application development and deployment in an accelerated fashion, portability of services, with no vendor lock-in are the major offerings provided by Helion platform.

When setting up a Helion cloud environment, one has to enable the Orchestration Service. Orchestration Service also entitled as Heat, encloses a Heat engine- core part of the Orchestration layer that will help one to automate the creation of composite infrastructural applications based on Cloud Maps.

B. HPE Serviceguard

One of the major contributions of Hewlett Packard Enterprise is a product that provides High Availability for the various components of the instances and this product is coined as HPE Serviceguard. Serviceguard monitors the health of several constituents within each instance and quickly responds to failures in a way that removes or diminishes application downtime. Serviceguard is able to distinguish and retort to failures in: System Processing Unit (SPU), System memory, System and application processes, Local Area Network media and adapters, Storage subsystem and adapters.

Serviceguard cluster implementations attempt to build redundancy into clusters to eliminate single points of failure, including multiple network connections and data storage which is ceaselessly connected via storage area networks. Application services such as individual Linux processes are grouped together in packages; in the event of a breakdown or disruption of the working service, node, network linkage, or other resource, another active node in the cluster will start up with all the services that were primarily functioning on the failing node and these transfer is done by Serviceguard with the negligible intervention. Free trial version of the product is available for customers to flourish the experience of having continuity in their business services in case of any failure.

V. IMPLEMENTAION

A. Initial Implementation

In the initial implementation, cloud set up was required, we set up OpenStack cloud environment – the foundation of

HPE Helion private cloud-system. Private and a Public network was created by embedding the necessary security protocols in them. As an initial check virtual machines were spawned with the "**nova boot**" command to see the virtual machine had the necessary prerequisite's to carry out the elementary complex operations.

B. Parallel Restart

The requirement for the Redhat OpenStack image was to enclose cloud packages, remote package managers as needed by Serviceguard and several network configuration files. For this Virtual machine was spawned on the physical host with Redhat as an Operating system using lib tools, all the necessary packages were installed and the required configuration changes were done. Using different tools, compression and conversion of an iso image to the fully developed Redhat cloud image in qcow2 format – most suitable format to run on Kernel based Virtual machine (KVM) hypervisor was performed. The image was uploaded to one of the image repositories of the cloud environment.

C. Parallel work on the Cloud Maps

Cloud Maps are written in the YAML (YAML aint a markup language) – data serialization language enclosing the features of C,C++ and Perl to pull out the required network, Operating system image and all the required parameters from the respective services of OpenStack cloud environment. Validity of Cloud Maps are verified using the JSON validator. Cloud Maps are createdaccording to the end user requirements taking into account the number of nodes required in the cluster, an image to be deployed in the nodes, network on which the instances will reside and so on.

Shell script containing the logic to enable the High Availability feature is ingrained in the Cloud Map. Shell scripts automates the logic to pull out the High Availability ISO from the image repository to the newly spawned instances.The logic to install the High Availability product HPE Serviceguard on the instances getting spawned is written in PERL with all the necessary core functions. Cloud package deep-sealed in the Redhat image will enable the execution of this shell script during the boot time of the instance.

C. Cloud Map Structure

Following depicts the structure of Cloud Maps:

heat_template_version: 2013-05-23
description: Purpose of the Cloud Map
parameters:
image:
type: string
label: Name of an image or ID assigned to it
description: Image to be used for launching a VM
default: Redhat
key:
type: string
description: Name of the key pair to assign to instances
default: heat_key
flavor:
type:string
label: Flavor
description: Type of flavor to be used
default: m1.medium
private_network:
type: string
label: Private network name or ID
description: Network to be attached to the instances
default: private_network
resources:
<Resource_name>
<type of the heat resource to be ued>
properties:
name:<Name of an instance>
image: { get_param: image }
flavor: { get_param: flavor }
key_name:{ get_param: key }
networks:
-network:{ get_param: private_network }
user_data: |
#Shell script containing the core logic of the work to be executed at the boot time of the instance creation
user_data_format: RAW

Parameters section specifies the input parameters that have to be supplied by the user while instantiating Cloud Map. *Resource section* explains the actual resources such as instances, network and storage volumes that makes up the stack derived from the Cloud Map. *Output section* specifies the parameters that has to be displayed to the end-user at the end of the deployment of stack. When cloud Maps are executed using certain commands, these code files are fed into the Orchestration engine, where the actual processing is done. The working process is depicted in the following diagram:-

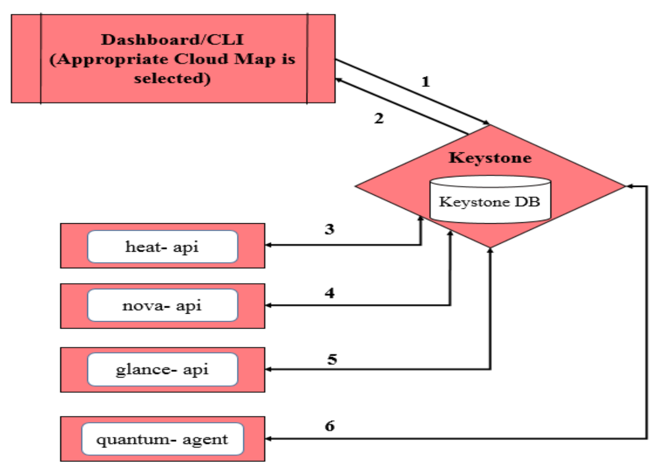

Figure 2: Workflow of the proposed system

User credentials are received at the user interface termed dashboard or at the command line interface, in the first step the credentials received are passed to the authentication service keystone which verifies if the user is authenticated. If the user is found to be a valid user, authentication token is sent back which is used to call various other services required for the implementation of the project. User selects the Cloud Map based on the requirements, Cloud Map will be first received by the heat-api which processes and parses the code file and makes a REST call to all the other services to create instances, retrieve the image from the image repository and the network to be attached to the compute instance. In the next step heat-api launches the stack which will contain the resources such as server, storage, flavor, network and etc. Stack topology can be viewed by clicking at the options provided at the dashboard console.

VI. INFERENCE

With the deployment of Cloud Maps, the following facts are observed:-

- Cloud Maps providing the High Availability feature reduces the time delays experienced by the end users in getting the Clusters up and running.
- Less Error prone
- Allows the end-users to provision clusters in few clicks.
- HPE Serviceguard provides High Availability for applications deployed inside these instances.

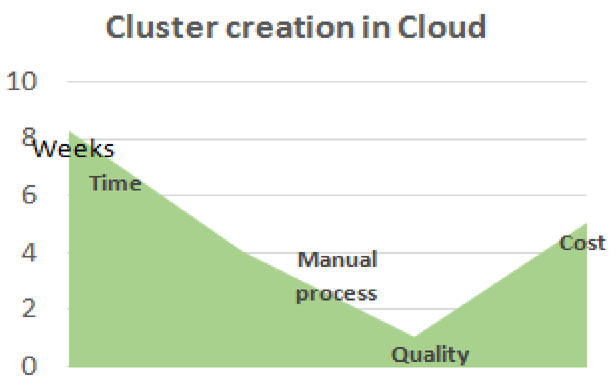

Figure 3: Manual process of Cluster Creation

Figure 3 shows the time taken for the creation of clusters in a manual process. Figure 4 shows the improvement in performance with the usage of Cloud Maps.

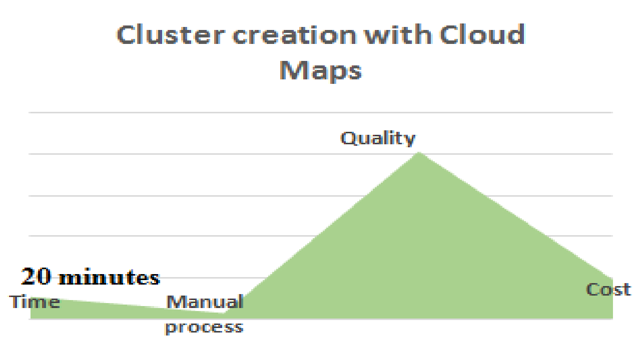

Figure 4: Cluster Creation using Cloud Maps

Table 1: Experimental analysis of the work undertaken

Metric/Feature	Traditional Approach for Creation of Clusters in the Cloud	Using Cloud Maps to automate the creation of Clusters.
Creation of Virtual Machines	Manual process	Automated using Cloud Maps.
Installation of the image (Redhat/SUSE)	Manual installation	Feature provided in the Cloud Maps.
Availability of the services when the node fails	No	Taken care by the High Availability product HPE Serviceguard
Time taken to create a Cluster	8 weeks	20 minutes
Quality	Low	High
Cost	High – as its error prone due to manual process	Pay per use
Ubiquitous	No	Yes, as the cluster instances will be hosted on the cloud platform.

VII. CONCLUSION AND FUTURE WORK

In this paper, we conferred the concept of Cloud Maps that enables one to spawn nodes in the cloud and form a cluster with a High Availability feature enabled for the services functioning in them. Cloud Maps helps to provision clusters with minimum clicks. Time taken to provision the clusters is brought down by an exponential factor. Further the Cloud Map solution provides a cost effective way to deploy nodes in the cloud in conjunction with the failback and failover policies of services running on these nodes with the help of the High Availability product termed Serviceguard embedded in the nodes. In our future work we would like to explore the OpenStack tool called “Flame” that will help one to dynamically generate Cloud Maps from the existing infrastructure and also provide end-users a copy of the Cloud Map template file which will help them get their instances up and running with the nominal time.

ACKNOWLEDGEMENT

This work was carried out at Hewlett Packard India Software Operations Pvt. Ltd., a Hewlett Packard Enterprise entity. Hewlett Packard Enterprise reserves all rights to this work. We thank all the individuals of this organization for their feedback and support and also the members of RV College of Engineering for their timely advice.

REFERENCES

i. *Kazuhide Kanenishi, Satoshi Togawa, “Private Cloud Cooperation Framework of e-learning Environment for Disaster Recovery”, IEEE International Conference on Systems, Man, and Cybernetics, 2013.*

ii. *Richard Harper, Valentina Salapura, “Remote Restart for a High Performace Virtual Machine Recovery in the Cloud”,* *IEEE 8th International Conference on Cloud Computing, 2015.*

iii. *Suguna, "Overview of Data Backup and Disaster Recovery in the Cloud", IEEE conference on Information Communication and Embedded Systems, 2014.*

iv. *Zhang Jian-hua and Zhang Nan," Cloud Computing-based Data Storage and Disaster Recovery", IEEE conference on Future Computer Science and Education, China, 2011.*

v. *Katsaros, Menzel,"Cloud Application Portability with OpenStack", IEEE World Congress on Services, 2015.*

vi. *Di Martino and Cretella,"Defining Cloud Services Workflow: A Comparison between TOSCA and OpenStack Hot",Ninth International conference on complex , intelligent and intensive systems, 2015.*

vii. *Prathiba and Sheethal Taneja," Using Heat and Ceilometer to provide Autoscaling in OpenStack" IEEE conference on Information and Communication Technology, 2014.*

viii. *Prashanth Shingale, Sujay Bothe,"Heat Template Generator in OpenStack", IEEE conference on Innovative Science, 2014.*