Cloud Service Orchestration

Orchestrates the VM Deployed in the cloud

[1]Abhishek Urs C J, [2]Ajay S, [3]Jeevan R, [4]Dhananjaya H M, [5]Siddharth B K

[1]Student, [2]Student, [3]Student, [4]Student, [5]Asst. Professor
[1]Computer Science Engineering,
[1]BGS Institute of Technology, Mandya, India

Abstract: There is a need for a service orchestrator which will Manage the Life cycle of the application/network services, Perform Day 0, Day 1 and Day 2 operations on these services, cater to the need of the hybrid cloud Deployment, Monitor the services and heal the services dynamically. The current work will provide Design and Implementation of a Cloud service orchestrator, which will help the application/network service providers to seamlessly deploy their application/network on a desired Cloud (Private or Public)

***IndexTerms* - orchestrator, cater.**

I. INTRODUCTION

The current lifestyle depends on the services provided by the applications. For E.g., Media Service (For E.g., Netflix), Education Service (Byju's, Vedanta, MS TEAMs, Google Meet), E-commerce Service (For E.g., Amazon), Commute Service (Uber, Ola), Hospitality Service (Oyo, MakeMyTrip), Communication Service (4G/5G mobile), Medical Service, Banking Transactions, IoT service. These Service providers often face difficulty in catering to the dynamic surge of the customers. The main issue is with the scale of the customers. For E.g., Covid situation has drastically affected the following services i) Netflix suddenly sees a surge in the demand of customers due to the ban on open hall theatres ii) Demand for online education sees the surge in the use of Online Tutorials and Online discussion. Few of the cloud service providers that are available are AWS, GCP, Azure, Openstack. However, the role of these cloud providers is to maintain the infrastructure or they provide infrastructure as a service to the application/network service providers.

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server. Clouds may be limited to a single organization (enterprise clouds), or be available to multiple organizations (public cloud). Cloud computing relies on sharing of resources to achieve coherence and economies of scale. Advocates of public and hybrid clouds note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand, providing the burst computing capability: high computing power at certain periods of peak demand. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models.

Cloud orchestration is a new buzzword in the industry; it was defined differently by researchers according to the context of its implementation. Indeed, the orchestration concept has been widely studied in the context of Web services and it was extended to handle other domains like Cloud Computing. In this domain, orchestration involves different types of services (computation, storage and network) and it can be realized at different layers of the Cloud services model. We can distinguish two types of Cloud orchestration approaches: orchestration of software services (or SaaSO) and orchestration of hardware services (or IaaSO).

Our implementation makes use of OpenStack as the underlying Cloud infrastructure. Nowadays OpenStack already propose an infrastructure orchestration service named as Heat, which can take as input a graph of resources such as compute resources, storage volumes and private network between resources, to automatically deploy a virtual service.

II. CLOUD SERVICE ORCHESTRATION RECENT SURVEYS

Fault handling and recovery from runtime failures of cloud applications. Monitoring data and make them accessible to external systems, such as billing platforms, to accurately charge users for the services they consume. The orchestration of the VM was limited to its scaling.

III. METHODOLOGY

Here we are taking input from the RestAPI and YANG Model. By invoking the APIs from the web-client we are able to communicate with the North-Bound Interface which in-turn communicates with the Resource orchestrator that helps to create a Virtual machine in the OpenStack. The complete is stored in the MongoDB. Kafka as the message streaming bus and Ubuntu OS for all these functionalities.

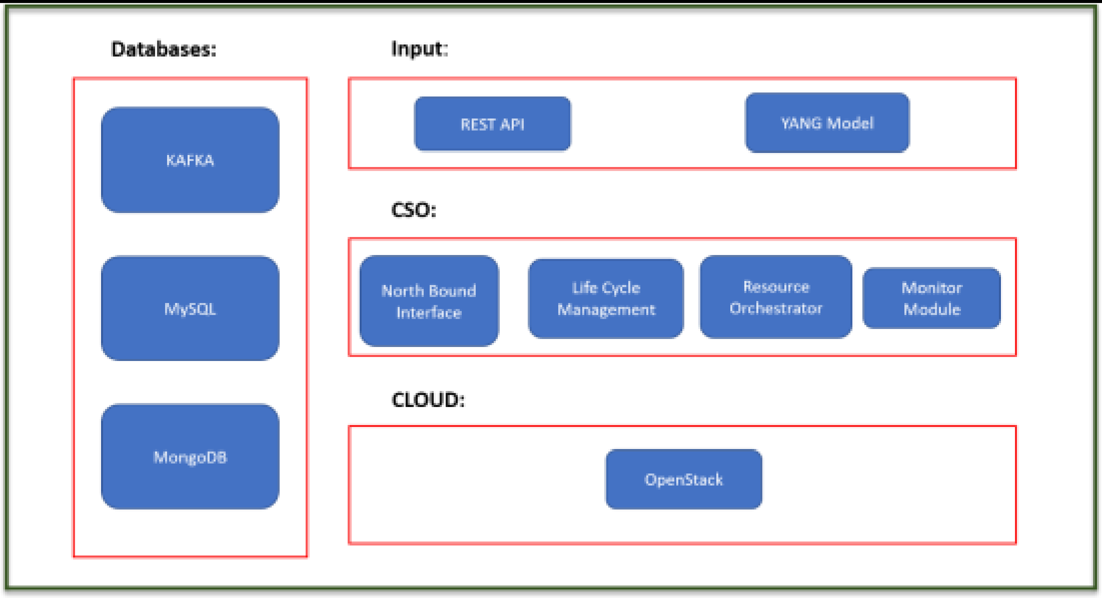

Fig 1: System Architecture

3.1 RestAPI

REST is acronym for REpresentational State Transfer. It is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources. An API for a website is code that allows two software programs to communicate with each other

3.2 Ubuntu

Ubuntu is a Linux distribution based on Debian and mostly composed of free and open-source software. All the editions can run on the computer alone, or in a virtual machine. Ubuntu is a popular operating system for cloud computing, with support for OpenStack.

3.3 MongoDB

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

3.4 Kafka

Apache Kafka is an open-source stream-processing software platform developed by the Apache Software Foundation, written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

3.6 MySQL

The results coming from the OpenStack is maintained by the MySQL Database.

3.5 Docker

Docker is a set of platforms as a service (PaaS) product that use OS-level virtualization to deliver software in packages called containers. Here all the components are built in docker container.

IV. IMPLEMENTATION

4.1 System Flow

The YANG model script is processed by the NBI of the CSO and subsequently through RO uses the specified cloud to deploy the required services. Once the required services are deployed, the status of the deployment is returned back by the RO. LCM module will manage the Life cycle of the application service. MON module of the CSO will continuously monitor the given metrics of the of the service, if it finds any discrepancy, will pass on the information to the LCM for necessary action.

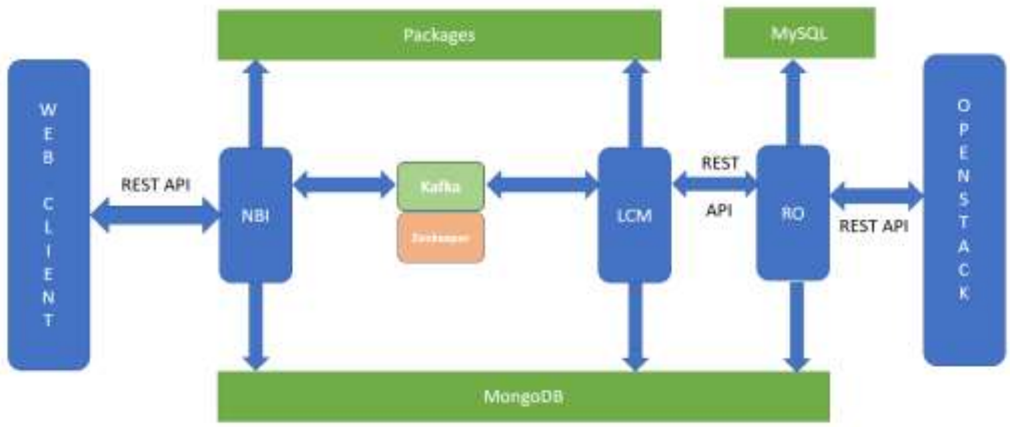

Fig 2: Flow Diagram

The above architecture presents the components of the Cloud Service Orchestrator. The application's compute requirements, its configuration and the network configuration and monitoring parameters is modelled through YANG modelling language and is presented to the NBI of the CSO. The operations on the application service are specified through REST API.

4.2 Use Case Diagram

Fig 3: Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Here the admin initially invokes an API using the Web Client, he also launches a Virtual Machine (VM). When VM instance is launched the Virtual Infrastructure Manager (VIM) issues an ID which can be used to communicate between another actor OpenStack. At this time, he manages all the life cycle of the launched VM instance. Finally, if admin wishes to delete the launched Instance, he can delete that. This completes the life cycle of the admin role.

4.3 Sequence Flow

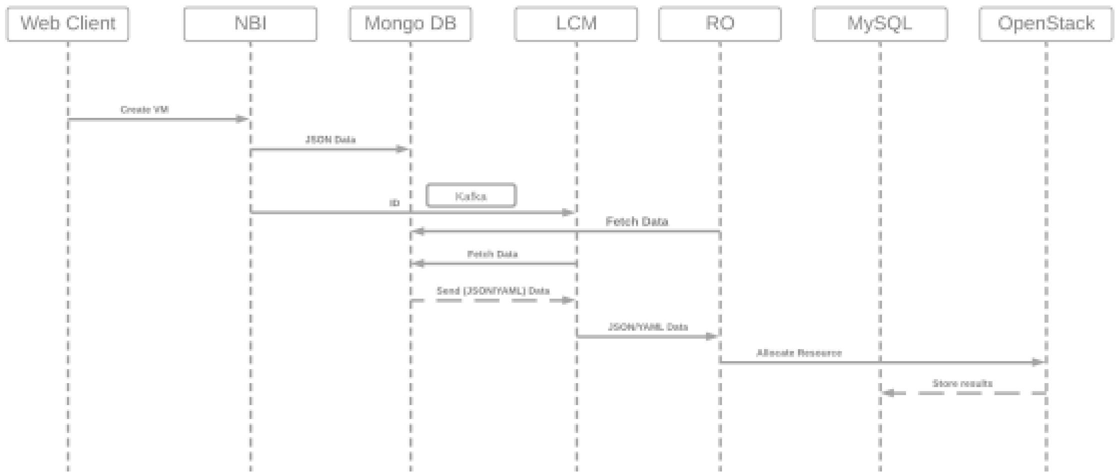

Fig 4: Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

The above sequence diagram illustrates the workflow of the Cloud Service Orchestration. The user will provide the required configuration of the VM through the web client. Further, the web client will pass the required configuration to NBI. The operations on the application service are specified through REST API. NBI stores the configurations in MongoDB and it sends the ID of this configuration to LCM through Kafka. The LCM will get details of the configuration which is stored in MongoDB by the received ID. LCM will invoke RO to allocate the resources specified in the configuration. The specified resources are allocated in Open Stack and the results are stored in MySQL.

V. Results and Analysis

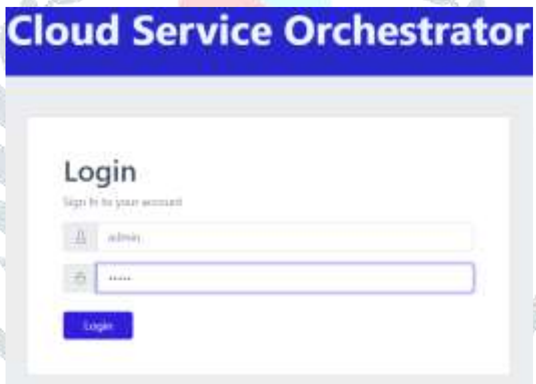

Fig 5: Login Page

Here the user authentication takes place where if the credentials are correct the only the token is issued for the user. This token is used for the further transactions.

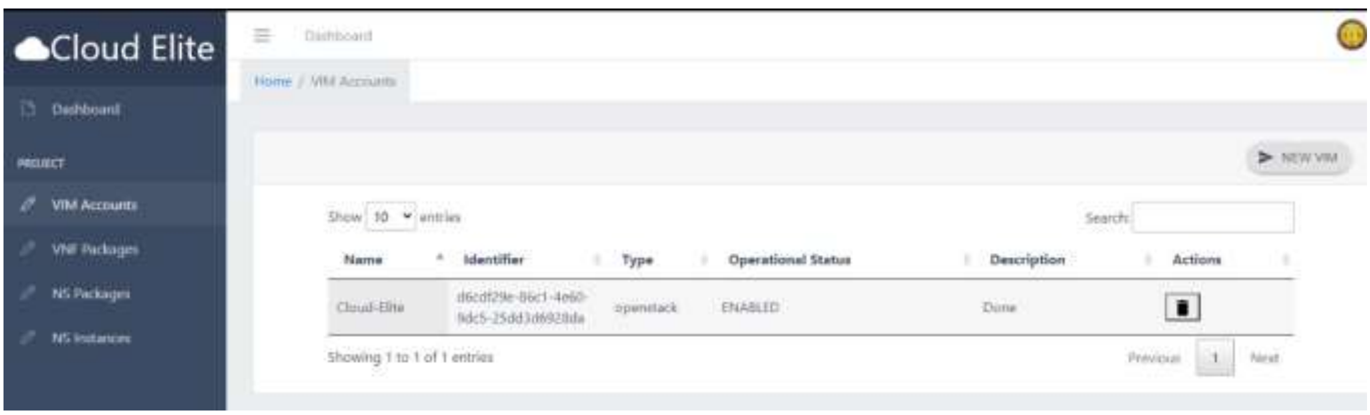

Fig 6: Creation of VIM Account

Here we pass the credentials of the OpenStack account so that it establishes the bridge between the web-client and the OpenStack.

Fig 7: Uploading VNF packages

The details of the image and flavor of the VM instance is given here.

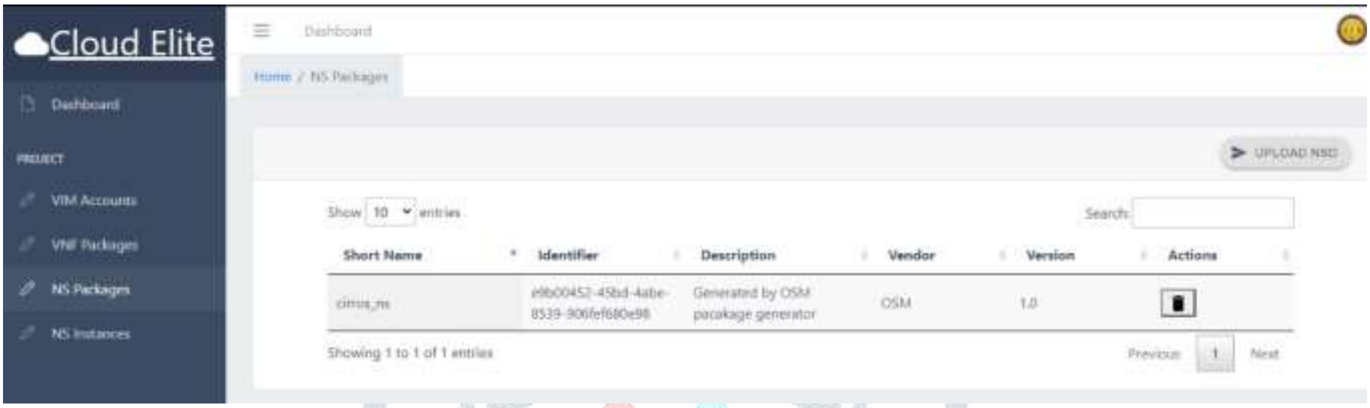

Fig 8: Uploading NSD packages

Here the network configurations required for the VM is specified.

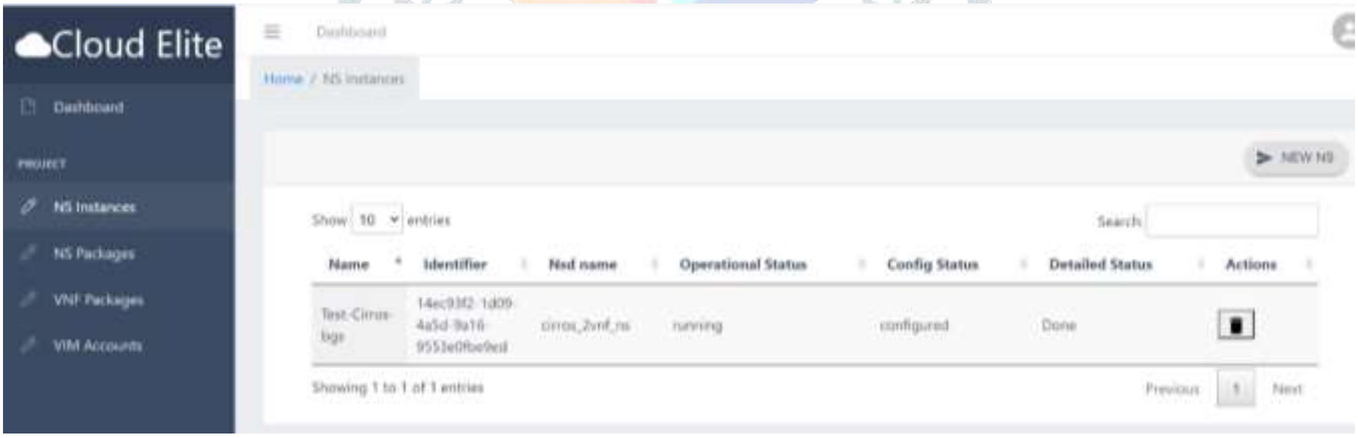

Fig 9: Creating NS instance

Here the we mention the name of the VM to be launched and once all mentioned steps are success then a VM is ready to launch in the OpenStack.

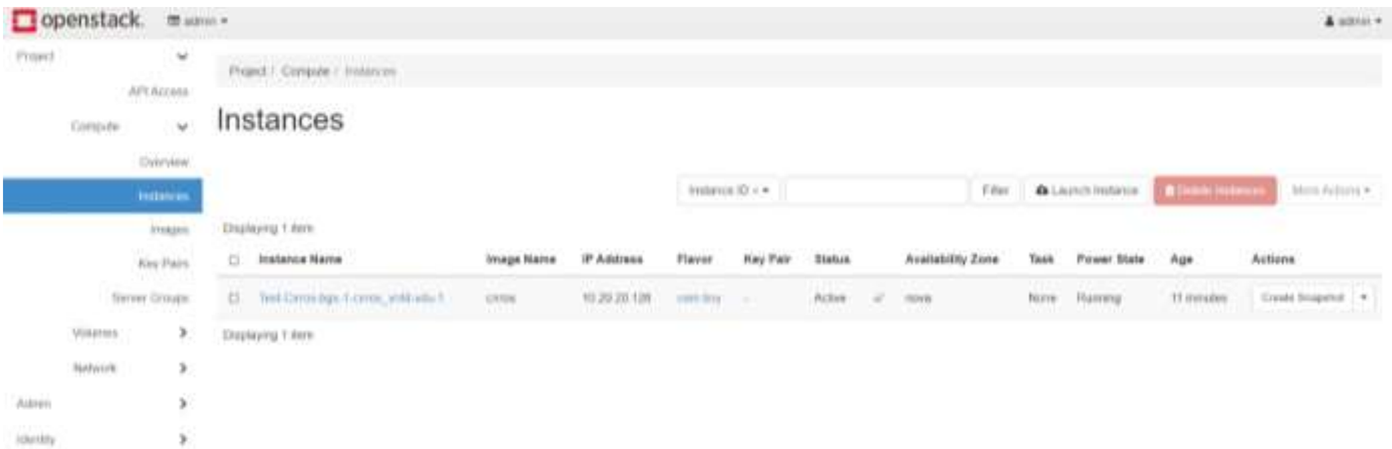

Fig 10: VM launched in OpenStack

VI. CONCLUSION AND FUTURE SCOPE

Cloud Service Orchestrator will aid application/network service providers to seamlessly deploy their services on the pre-specified cloud, manage their services, monitors the services and automates the autoscaling of the services which will help the service providers to meet the dynamic needs of the customer demands.

Now, it is only possible to connect our service orchestrator to one cloud service provider but in future it can be connect to multiple cloud service providers.

REFERENCES

[1] https://osm.etsi.org/wikipub/index.php/Welcome_to_OSM!

[2] S. Van Rossem et al., "Deploying elastic routing capability in an SDN/NFV-enabled environment," 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, 2015, pp. 22-24

[3] S. Schneider, M. Peuster, W. Tavernier and H. Karl, "A Fully Integrated Multi-Platform NFV SDK," 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Verona, Italy, 2018.

[4] A. Brogi, J. Carrasco, F. Durán, E. Pimentel and J. Soldani, "Robust Management of Trans-Cloud Applications," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 2019.

[5] M. Anand, "Cloud Monitor: Monitoring Applications in Cloud," 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, 2012.

[6] M. Jammal, A. Kanso, P. Heidari and A. Shami, "Availability Analysis of Cloud Deployed Applications," 2016 IEEE International Conference on Cloud Engineering (IC2E), Berlin, 2016.

[7] A. Noor et al., "A Framework for Monitoring Microservice-Oriented Cloud Applications in Heterogeneous Virtualization Environments," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 2019, pp. 156-163.

[8] J. P. Magalhães and L. M. Silva, "A Framework for Self-Healing and Self-Adaptation of Cloud-Hosted Web-Based Applications," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 555-564.

[9] V. Eramo, F. G. Lavacca, T. Catena, M. Polverini and A. Cianfrani, "Proposal and Investigation of a Scalable NFV Orchestrator Based on Segment Routing Data/Control Plane," 2018 14th International Conference on Network and Service Management (CNSM), Rome, 2018, pp. 426-431.

[10] S. Dhakate and A. Godbole, "Distributed cloud monitoring using Docker as next generation container virtualization technology," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-5.