Automation and Customization of Openstack

Muhammad Usman Ghani
Computer and Information Systems Engineering
CIS
Karachi, Pakistan
mghani828@gmail.com

Danyal Javed
Computer and Information Systems Engineering
CIS
Karachi, Pakistan
daniyal.mir.dj@gmail.com

Muhammad Ammar
Computer and Information Systems Engineering
CIS
Karachi, Pakistan
maansaridot@gmail.com

Abstract*— A user-friendly platform to automate the whole process of cloud development which can eliminate the time of deployment and substantial cost of hiring professionals, based on an open source software OpenStack. The OpenStack provides multiple cloud components for different use cases, and because of its open nature, anyone can add additional components to OpenStack to meet their needs. This makes the installation of OpenStack for the common user very confusing and time-consuming. The idea behind our product is to automate the whole process of cloud development. For this, we need some sort of automation tool that should generate server configuration files on run-time and help us to achieve our goal. Automation tools play an important role in terms of server configuration and management. These tools use provisioning scripts to make the server reach a desirable state. Our product has automated the process of configuring individual components of OpenStack using an automation tool and provides users an effortless installation of the cloud. Different scripts are created to automatically fetch the required components and configure them according to the specification set by the user. These scripts are written in YAML language and we have used Ansible as our configuration manager. It detours all barriers and provides users with their cloud platform running on their machine.

Keywords: *Openstack, Automation, Ansible, Customization*

I. INTRODUCTION

A cloud solution based on OpenStack that allows users to create their own private or public cloud according to the requirement of the user, without the hassle of installing and configuring the individual technologies and services needed to build a complete running cloud environment. This solution eliminates the time of deployment and the substantial cost of hiring cloud architect professionals to build the cloud environment for you.

Our product has automated the process of configuring individual components and provides users an effortless installation of the cloud. We have written some scripts that automatically fetches the required components and configures them according to the specification set by the user. It detours all barriers and provides users with their cloud platform running on their machine.

Small scale organizations and educational sectors can take advantage of our solution. They can build their cloud platform without spending on public cloud service providers. It can help organizations to focus more on their business rather than wasting time and money on the technical side of cloud development. This solution can also come very handy for DevOps Engineers and others as they can use this platform for learning and training. All in all, this solution provides a cloud platform to everyone without the struggle of configuration and installation.

II. BACKGROUND

Cloud computing provides users with a shared pool of resources that includes resources related to networking, storage, processing, and servers. There are unlimited use-cases for cloud computing. A tremendous increase has been noticed for computing resources and to deal with the demand people come up with different solutions. There are several types of cloud models such as private cloud, public cloud, and hybrid cloud. Our project focuses on automating the process of deploying a private cloud through open-source software. Girish L S. [1] has discussed in detail about building a private cloud using OpenStack. In the open-source market, plenty of software is available that can build a private cloud such as OpenStack, Eucalyptus, CloudStack, and OpenNebula. We choose OpenStack because of its highly scalable structure and very cooperative community. Also, OpenStack has been used by so many large companies which makes it even more favorable over other Software. OpenStack itself is highly customizable and configurable but at the same time, it can get very confusing for the users to implement it themselves. Our goal is to automate the installation and deployment phase of OpenStack so that small startups, the education sector can set up their private cloud without a hassle. There are not many products or solutions that directly automate the process of setting up a private cloud with OpenStack. Two such products that are similar to our project are DevStack and PackStack, but these products have their shortcomings. DevStack and PackStack are widely supported on different Linux distros but the most preferable distros for OpenStack installation is Ubuntu, CentOS or RedHat DevStack is a set of pre-defined scripts developed by OpenStack that can set up an OpenStack cloud on your machine or any test server. It is installable on the latest Ubuntu version, Fedora, and CentOS. DevStack is an all-in-one installation that is good for people or developers to get acquainted with the OpenStack environment without having in-depth knowledge about the OpenStack lego-like structure. the installation of DevStack is very susceptible to error and it is not meant to be used in production because after a reboot. It is impossible to get back to the recent stable state all the running processes crashed, and data will be lost permanently. PackStack is similar to DevStack. It is also a utility to install OpenStack. but it is only supported for RedHat based Operating Systems. PackStack is a little mature in terms of the installation process and stability compared to DevStack. PackStack uses puppet under the hood to deploy various services of OpenStack on multiple servers. packstack is also persistent. Our solution is comparable to PackStack implementation in a way that we also have a configuration management tool to install and configure all the services. Our installation process is not limited by vendor-specific operating systems like PackStack. It is installable on all OpenStack supported Linux distros. We also enforced persistence so that in case of a power failure we can go back to the last stable state, unlike DevStack. In terms of architecture, we also want to provide flexibility to use the user so that they can build around their needs. PackStack and DevStack do not have any kind of central logging mechanism which can be problematic if anything fails it can be hard to trace it down without proper logs. In the project, we also implemented a centric logging mechanism to provide end-users with complete formatted and GUI friendly logs. Our target is to automate the installation at the same time providing a production-ready solution.

III. OPENSTACK DEPLOYMENT

We use a configuration management tool called chef for production-ready OpenStack deployment. We create our customized Ansible playbooks for each OpenStack component which are mentioned above. The overall deployment process takes about one hour and all the magic happens in this period.

Openstack deployment is not like typical application deployment, instead, it takes careful planning of resources and network fabric design underlying it. Due to this reason, we have to perform introspection on each node before deploying OpenStack on them. The process of introspection is done by using the Out-of-Band Management network (OOB). Introspection can be done in multiple ways. For Virtual deployment on the environment we use an ipmi driver, for Dell server we use iDRAC and manual introspection is also supported. Deployment can be done in any architecture as the user wants. We support HA for multiple controller nodes as well.

IV. CENTRAL LOGGING SYSTEM

We implemented a log management platform that helps in debugging the overall cloud environment state. ELK is an open-source, most popular log

analysis software provided by the elastic company. The ELK stack comprises Elasticsearch, Logstash, and Kibana. Elasticsearch, It is a search engine that powers the stack, used for deep search and data analytics. Logstash, It processes the data and sends it to Elasticsearch for storage and indexing. It is also used for centralized logging, parsing, and log enrichment. Kibana, It is a powerful and beautiful user interface of the Elasticsearch cluster where users can visualize log messages.

V. CONCLUSION

The main idea behind this product is to automate the whole process of cloud development using OpenStack, and we have successfully automated the process of configuring OpenStack based on user's requirements. In our current workflow, the requirement is gathered and fed into the script through the command-line. But we wanted to gather these requirements through a GUI and process them directly without the use of command-line. Some more enhancement can also be made, such as additional pre-configured packages can be provided to the user to choose from. Currently, this product is limited to a basic learning purpose environment and a web hosting environment. Integrating more OpenStack services will allow us to make more pre-configured packages e.g. packages for Big Data processing, Database-as-a-service, and much more. Another enhancement would also be made, is to add some type of self-healing mechanism to deal with crashes. The current central logging system can be improved with better reporting capabilities. Most importantly the project should be licensed and implemented with proper pricing structure before product distribution. These enhancements will allow the project to reach its full potential by providing a reliable automated OpenStack deployment.

VI. REFERENCES

[1] Girish L S, Dr. H S Guruprasad. (June 2014). Building Private Cloud using OpenStack https://www.ijettcs.org/Volume3Issue3/IJETTCS-2014-06-17-090.pdf