Spawning Virtual HPCs Using OpenStack

Michael Olson
SPIN Undergraduate Researcher,
NCSA
maolson2@illinois.edu

Lin Cheng
Senior Computer Science Major,
UIUC
lcheng10@ncsa.illinois.edu

Dr. Volodymyr Kindratenko
Senior Research Scientist,
NCSA
kindr@ncsa.illinois.edu

Abstract

The emergence of cloud computing – using remote servers to store, manage, and process data – offers an effective framework for the optimization of operational expenses. One such solution, OpenStack, allows for the installation of a cloud operating system across multiple nodes to create a highly configurable high-performance cluster (HPC) [1]. To effectively utilize the entire cluster, OpenStack allows multiple users to create virtual HPCs within the restrictions of the physical hardware. In this way, the needs of many users can be met simultaneously, and the physical hardware's average idle time is decreased. Additionally, the resources available to one user can be scaled to use more or less than any other user in real time, enabling each user use only what they need, and optimizing the performance of each project. One particular service officially supported by OpenStack is called "Heat." This allows the user to use customized templates to spawn their virtual HPC's, speeding up the process of allocating and freeing resources. All of this can be done in an easy to use online dashboard called "Horizon," allowing the user to log into the cluster from their web browser anywhere they have internet access. This work outlines the benefits of OpenStack, the installation of version "Mitaka," released in April 2016, and the configuration and use of the Heat orchestration service. I also outline the strengths and weaknesses of Heat when considering deployment in a production environment, particularly the lack of a strong mechanism for sharing templates. For the purpose of this report, several acronyms such as HPC will be used in place of high-performance cluster and other repetitive terms.

1 Introduction

OpenStack is an open-source cloud operating system built by a thriving community of developers and users. Users seeking commercial support from OpenStack's strong expanding ecosystem can choose from a wide variety of products and services offered in the OpenStack Marketplace. This revolutionary software controls large pools of compute, storage, and networking resources throughout a datacenter, and hundreds of the largest companies in the world, such as Intel, CERN, Yahoo, Cisco, and the Wikimedia Foundation, rely on OpenStack to run their businesses every day [2]. The Heat orchestration service uses a Heat Orchestration Template (HOT) to easily create and manage cloud resources. Heat also offers support for Amazon Web Services' (AWS) CloudFormation templates to allow the user to describe any AWS resources, associated dependencies, or runtime parameters. My research was to install OpenStack with Heat on a 10-node HPC to determine the strengths and weaknesses of deploying Heat in a production environment. If it proves effective, this solution would optimize resource allocation and operational expenses on the HPC.

2 Methodology

The OpenStack system consists of several key services that are separately installed. These services work together depending on the user's cloud needs. The official nicknames of these services are provided in parentheses to avoid confusion.

The installation of OpenStack begins by installing a supported operating system on each node. In this case, I chose Ubuntu 14.04 for its long term service and support. During this installation, the public IPs, private IPs, gateways, and DNS name servers were all manually configured, and "OpenSSH" was marked for installation when prompted. I then updated each node and installed the Network Time Protocol (NTP) and OpenVSwitch packages. OpenVSwitch is a production quality, multilayer virtual switch designed to enable massive network automation through programmatic extension. I then configured each node to communicate using OpenVSwitch such that a specific network topology was reached. This network topology creates one controller node, one network node designated as a gateway, one storage node, and 7 compute nodes. It also restricts remote connection to the cluster by first having to connect to the controller node. In this way, the user must authenticate themselves with the controller node before accessing any resources. Once all of this is complete, we are ready to move onto the official installation guide [3]. Since I used OpenVSwitch and already configured NTP on each node, I was able to skip the Security, Host Networking, and NTP sections when installing the Environment for OpenStack. The NoSQL section was also skipped. The only dependency of NoSQL is the telemetry service (ceilometer), and was ignored during this particular installation.

The identity service (keystone) provides a single point of integration for managing authentication, authorization, and cataloging services. Keystone is installed on the controller node. During this installation, the user will install and configure a MySQL database and an Apache HTTP server.

The image service (glance) enables users to discover, register, and retrieve virtual machine images. This is a very simple installation and is a core component to OpenStack and Heat.

The compute service (nova) is a very extensive service that creates and terminates virtual machine instances through hypervisor APIs. It also handles metadata requests from instances and provides a proxy for accessing running instances through VNC, SPICE, or an OpenStack specific Java client called XVPVNC. The database also stores information on available instance types, instances in use, available networks, and project spaces.

Installing the networking service (neutron) was special in this case, as OpenVSwitch is no longer the officially supported networking method. As a result, I had to refer to the older documentation for OpenStack "Kilo" (January 2015) to properly configure and test neutron. The Mitaka documentation is used to set up the MySQL database, however, when reaching the networking options for a self-service network in the instructions, the user must refer back to Kilo documentation. The user may then proceed again with the Mitaka documentation when configuring the metadata agent and the compute service (nova) to use neutron. Even when properly configuring the networking section, there seems to be a bug with OpenStack Mitaka and OpenVSwitch that interferes with the iptables. We were able to temporarily circumnavigate this by disabling the iptables, however, disabling iptables is a security risk for production environments. Further, it would take extra time to find and fix the bug, and should be taken into consideration when deploying in a professional setting. The previous documentation also provides some helpful suggestions that the new documentation does not provide for checking neutron's functionality. I highly suggest using these resources.

It is at this point when the user can install the dashboard service (horizon), a web interface for cloud administrators and users to manage the various OpenStack resources and services. This deployment uses an Apache web server. As an admin, this is also a way to allocate project spaces for different users, assign those users into groups, and even designate roles for each user.

The block storage service (cinder) provides block storage devices to guest instances. The shared file systems service (manila) provides coordinated access to both shared and distributed file systems. The object storage services (swift) work together to provide object storage and retrieval. Together, these three storage services are able to tend to the user's unique needs.

Finally, the orchestration service (Heat) can be installed to create and manage cloud resources using templates. The service itself acts as an OpenStack user with administrative privileges when parsing the templates. Once installed, an "Orchestration" tab appears in the dashboard (horizon) that allows the user to browse the available resources for use in HOT templates. A few basic AWS CloudFormation templates are provided in this list as well, such as auto scaling web-servers and volumes.

Using Heat involves writing scripts in YAML format - a human friendly data serialization standard for all programming languages - (.yml). In the Resources tab added to the horizon dashboard when installing Heat, the parameters are laid out for the user to start writing their scripts. Unfortunately, though it uses a simple interface, it is not a particularly friendly service. The scripts are fairly difficult to read at first, and minimal examples are provided, so the user is forced to either test their scripts or request their system administrator to write it for them. The heat template guide does help, however, someone has to take what the user wants and turn it into a script, which takes time. The script can either be written directly in the dashboard or uploaded to the dashboard, or could even be run in the terminal. Though there is no built-in method for sharing templates, or for extracting a script from an existing environment, the YAML files can still be emailed around to ensure a user or group of users acquire their personalized virtual HPC.

3 Literature Review

A group of four researchers published an article in the Journal of Cloud Computing in 2014 exploring an experimental template management technology that overcomes the challenges that Heat presents. They found that Heat does not have a sharing mechanism for templates, it cannot extract templates from existing virtual environments, and it does not reflect actual environment changes to stack information. As a result, they proposed their own template management system. The template management system they proposed was successfully implemented on OpenStack version "Folsom." From this template system, they confirmed its feasibility and measured performance. It was listed in the research report that this technology will be modified for newer versions of OpenStack such as "Icehouse" or "Juno," but I was unable to find anything confirming these modifications, nor any mention of the latest version of OpenStack "Mitaka" [4].

DevStack is a series of extensible scripts to quickly create an OpenStack development environment. Heat is an orchestration tool for all of the infrastructure that OpenStack provides - not only servers, but volumes, networks, floating IPs, containers, databases, queues, and so on. Older versions of OpenStack such as "Icehouse" and "Juno" have Heat enabled in Devstack by default, but newer versions require enabling Heat services in Devstack [5].

Heat was successfully implemented at CERN in Summer 2013 to deploy webservers. With Heat, they were able to scale the number of webservers up or down according to their memory usage. The configuration of each webserver was described beforehand, and they were deployed as part of an auto scaling group [6].

4 Results

The core services of OpenStack Mitaka were successfully installed on the 10-node cluster. Heat was also successfully implemented on this installation. In deploying a few of the YAML scripts in Heat, the most prominent issue seems to be that Heat does not help the user write these scripts. A built-in IDE for the HOT template and CloudFormation template could be built into horizon for useful features such as auto completion and typo detection. Also, as published in the Journal of Cloud Computing in 2014, Heat does not have a sharing mechanism for templates and it cannot extract templates from existing virtual environments. That being said, it still remains a very powerful tool for managing the entire lifecycle of infrastructure and applications within OpenStack clouds.

5 Discussion

Two Github users by the name of Brint O'Hearn and Gaurav Bajaj have compiled a repository of orchestration templates for use in Heat that list the requirements to run each script including version number and installed resources. Since there is no template extraction offered through Heat, this is a great way to get the community involved and provide example templates for those learning to write the YAML scripts [7]. This is not a perfect solution, as it requires the user to scrape through the repository to find what they are looking for without any network topology diagrams provided. It is, however, a step in the right direction. Additionally, this successful implementation of Heat on OpenStack Mitaka brings reassurance for the deployment of Heat on production clusters for its clear and concise documentation.

6 Conclusion

OpenStack is an efficient and powerful tool with great documentation on the installation and configuration process. The Heat template describes the infrastructure for a cloud application in a text file that is readable and writable by humans. It can also be checked into version control, diffed, &c. When the user needs to change their infrastructure, they can simply modify the template and use it to update the existing stack. Heat knows how to make the necessary changes. It will delete all of the resources when the user is finished with the application, too. Additionally, the Heat team is still working on providing even better integration between infrastructure and software.

7 Recommendations

I suggest the next steps for Heat to be the implementation of template extraction from an existing virtual cluster and hiding the YAML script writing all together. A graphical user interface could be created for simple drag-and-drop cluster building. The network topology could then be easily visualized and saved as a YAML template and used/shared at a later date. If the user intends to create a very complex setup that this graphical interface couldn’t satisfy, they could edit the YAML script generated by the graphical interface in the same way the user is currently being forced. This interface would be more intuitive and easy to use for those who want to deploy Heat quickly and efficiently.

References

[1] "Open Source Software for Creating Private and Public Clouds." *OpenStack Open Source Cloud Computing Software*. Rackspace Cloud Computing, n.d. Web. 27 July 2016.

[2] "OpenStack Powers Demanding Production Workloads Worldwide." User Stories » OpenStack Open Source Cloud Computing Software. Rackspace Cloud Computing, n.d. Web. 27 July 2016.

[3] "OpenStack Installation Guide for Ubuntu." *OpenStack Docs*. Rackspace Cloud Computing, n.d. Web. 27 July 2016.

[4] Yamato, Y., et. al. “Development of template management technology for easy deployment

of virtual resources on OpenStack”. *Journal of Cloud Computing* v. 3.1. December 2014:1. PDF. 16 June 2016.

[5] "Heat and Devstack." *OpenStack Docs*. Ed. Heat Developers. N.p., n.d. Web. 16 June 2016.

[6] Michelino, D., 2013. *Implementation and Testing of OpenStack Heat*. CERN openlab Summer Student Report. September 2013. PDF. 16 June 2016.

[7] O'Hearn, Brint, and Gaurav Bajaj. "Rackspace Orchestration Templates." *GitHub*. Github, Inc., 9 Sept. 2014. Web. 27 July 2016.