WCU WEST CHESTER UNIVERSITY

Technical Documentation

Deployment and Implementation of CloudLab OpenStack Profile to Support Docker Virtualization

Austin Bramley, Akash Kumar, Endri Koti, Simeon McGraw, Andrew Valenci

Computer Science Department

West Chester University

700 S High St, West Chester, PA 19382

ABSTRACT

The CloudLab OpenStack default profile contains the setup of 1 compute node that supports KVM-based virtual machines without the support for Docker. The primary task of this project is to augment the CloudLab OpenStack default profile to support the inclusion of a compute node type that provides support for Docker virtualization. Through the use of Zun, an OpenStack container service, we augmented a compute node to support the launching of a Docker container as an OpenStack managed resource. The coded bash scripts added to the default Openstack profile provide automated installation of systems necessary to support Docker virtualization. Currently we have augmented the default Openstack profile to install Etcd, Zun, and Kuryr in the controller node. Also, we provided the installations of Docker, Zun, and Kuryr in the compute node.

Keywords: Cloud Computing, Docker Virtualization, OpenStack, Zun, CloudLab, Automated Bash Script

1. INTRODUCTION

1.1. Purpose

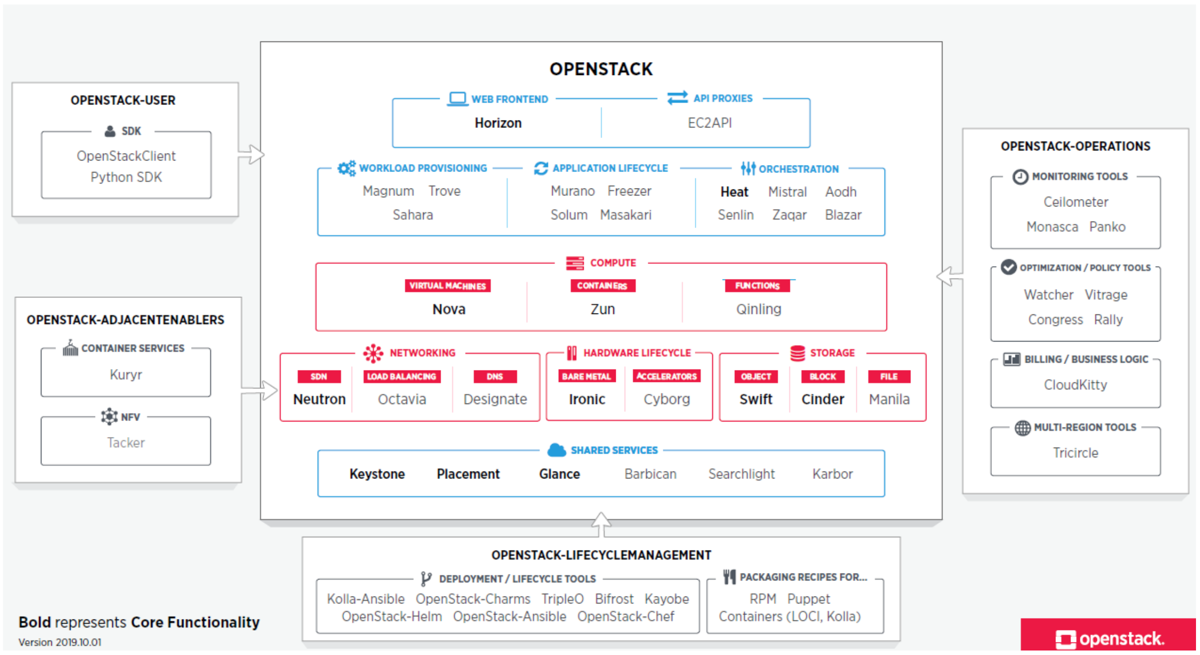
The purpose of this project is to modify the default Cloudlab Openstack profile to provide support for a compute node that allows for Docker virtualization managed by the Openstack dashboard. The current Openstack default profile provides support for Kernel-based Virtual Machines but lacks support for Docker virtualization. Our goal is to implement the Zun Container service to provide the default Openstack profile with the ability to manage containers such as Docker.

1.2. CloudLab

CloudLab is part of the National Science Foundation's NSFCloud program. CloudLab is a project of the University of Utah, Clemson University, the University of Wisconsin Madison, the University of Massachusetts Amherst, Raytheon BBN Technologies, and US Ignite [1]. CloudLab is a testbed designed to allow researchers to experiment with cloud architectures and the new applications that they enable. It is built for running experiments that will lead to new capabilities in future clouds, or to a deeper understanding of the fundamentals of cloud computing [1]. CloudLab is ideally suited to experiments that cannot be run in traditional clouds because they require control and/or visibility over parts of the system that would be “givens” in other clouds, such as the virtualization, storage, or network layers. CloudLab is built around profiles. A profile is a description of everything needed to build a cloud: the physical hardware (servers, disks, switches) and the software needed to transform it into a particular type of cloud. A profile is fully packaged and automated: building a cloud from scratch may take only minutes, depending on its size and complexity [1].

1.3. OpenStack

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms [4]. It is a free open standard cloud computing platform, mostly deployed as infrastructure-as-a-service (IaaS) in both public and private clouds where virtual servers and other resources are made available to users [3]. Users either manage it through a web-based dashboard, through command-line tools, or through RESTful web services. A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface [4]. Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications [3].
The history of OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA. As of 2012, it is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote OpenStack software and its community. More than 500 companies have joined the project [4].



1.4. Zun

Zun is an OpenStack Container service. It provides OpenStack the ability to manage application containers such as Docker. Zun provides API endpoints for Openstack to integrate with other OpenStack Services, such as Keystone, Neutron and Glance [8]. The Zun Compute service allows OpenStack to manage container services, including Docker, and uses Kuryr to connect the containers to Neutron. Zun can also connect to Glance, an option for storing container images, much like DockerHub [9]. Zun also has the ability to connect to block storage through Cinder, OpenStack's block storage service.

1.5. Docker

Docker container technology was launched in 2013 as an open source Docker engine that leveraged existing computing concepts around containers [5]. Docker focuses on the requirements of developers and systems operators to separate application dependencies from infrastructure. Containers are used as an abstraction at the app later to package code and dependencies together with the ability to run multiple on the same machine and share the OS kernel [6]. Containers and virtual machines function differently because containers virtualize the operating system instead of hardware making them more portable and efficient.

2. PROJECT PLAN

2.1. Creating Experiment

The First step of our project was to launch the CloudLab OpenStack default profile and manually install the systems required for Docker virtualization. The control node requires the installation of Etcd, Kuryr and Zun services. Etcd provides a reliable way to store data that needs to be accessed by a cluster of machines [10], [12], [14]. Also the compute node required the installation of Docker, Kuryr, and Zun services [11], [13]. Applications can easily read from and write data to Etcd. Kuryr is designed to function as a bridge between container frameworks and OpenStack. Refer to 1.4 and 1.5 for Zun and Docker details respectively. Once the required services are installed on both the controller and compute nodes it is possible to launch Docker as an OpenStack managed resource [15].

2.2. Installation of OpenStack and Zun with Automated Script

Once we got the installation of each of the services onto the controller and compute nodes, we used those steps to create two separate sets of bash scripts for the controller node, and for the compute node. We then add these scripts to the setup scripts in the CloudLab OpenStack profile to install these services in the CloudLab experiment's setup phase.

2.3. Progress

We have successfully created and implemented into the CloudLab OpenStack default profile a bash script to install the services required to complete our task. The bash script runs successfully and installs all necessary services to both the controller and compute node. We are currently working through issues on the final phase of the project regarding launching Docker as an appcontainer on the controller node.

2.4. Issues

Over the course of the project we ran into a number of issues when installing services to our OpenStack profile. Most of the issues were due to mistakes made during the installation of Zun and the writing of the necessary configuration files. Our automated script has solved these issues by providing scripted installations with no chances of typo created errors. At this time, we are currently faced with one unsolved issue of receiving a 500 error when trying to launch Docker as an appcontainer.

3. DELIVERABLE

3.1. Did team meet project deliverable two?

Considering that we have properly set up a CloudLab experiment from the CloudLab OpenStack default profile, finished installing Etcd, Docker, Kuryr, and Zun on the experiment, wrote the necessary scripts to automate the install of Docker on the compute node, wrote a 5+ page technical documentation, and filmed the presentation video, we believe that we have met the second project deliverable.

3.2. Will the project make the final deliverable?

While we ran into a lot of errors and issues while setting up the experiment and still do not have Docker properly working, we believe we are on track to getting this project done by the time the semester ends. We have a sliver of hope that our work will be recognized and cited. And if we don't succeed in our goal, then at least we will have learned something from our failures.

3.3. How was the group collaboration?

Due to the advent of the 2019-2020 coronavirus pandemic, direct in-person communication between team members was not possible. So, we resorted to using online means of communication to collaborate and stay connected with each other and with our professor. Here are some of the online tools and platforms we used:

- *GitHub* - for collaboration and sharing of project code.
- *Zoom* - for face-to-face communication and screen sharing.
- *Piazza* - for asking the professor questions when encountering issues.

- *Slack* - for instant communication and team coordination.
- *Google Drive* - for editing and sharing documents together.
- *Trello* - for creating a project board to track project progress and key information.

In addition, we scheduled Zoom meetings three times a week Monday, Wednesday, and Friday from 2.p.m to 3:15.p.m to ensure that all team members are actively engaged with the work and keep the project on schedule.

4. CONCLUSION

In conclusion, being a cloud provider is not an easy job. It requires deep knowledge of cloud platforms, infrastructure, networking, containerization, virtualization, Linux, bash scripting, etc. As a team we learned a lot completing this project which broader our knowledge in Cloud computing and Linux distribution.

5. REFERENCES

[1]. *CloudLab: https://www.cloudlab.us/#availability*
[2]. *CloudLab OpenStack default profile: https://gitlab.flux.utah.edu/johnsond/openstack-build-ubuntu*
[3]. *Intro to OpenStack: https://www.openstack.org/software/*
[4]. *OpenStack Wikipedia page: https://en.wikipedia.org/wiki/OpenStack*
[5]. *Intro to Docker: https://docs.docker.com/get-started/*
[6]. *Docker documentation: https://docs.docker.com/get-docker/*
[7]. *Installation Docker Engine on Ubuntu: https://docs.docker.com/engine/install/ubuntu/*
[8]. *Zun Wiki page: https://wiki.openstack.org/wiki/Zun*
[9]. *Zun documentation: https://docs.openstack.org/zun/latest/*
[10]. *Installation and configure Zun in controller node: https://docs.openstack.org/zun/latest/install/controller-install.html*
[11]. *Installation and configure Zun in the compute node: https://docs.openstack.org/zun/latest/install/compute-install.html*
[12]. *Installation and configure Kuryr in the controller node: https://docs.openstack.org/kuryr-libnetwork/latest/install/controller-install.html*
[13]. *Installation and configure Kuryr in the compute node for Ubuntu: https://docs.openstack.org/kuryr-libnetwork/latest/install/compute-install-ubuntu.html*
[14]. *Installation of Etcd in the controller node for Ubuntu: https://docs.openstack.org/install-guide/environment-etcd-ubuntu.html*
[15]. *Launch a container in Zun: https://docs.openstack.org/zun/latest/install/launch-container.html*
[16]. *OpenStack Forums: https://ask.openstack.org/*
[17]. *Project's GitHub Repository: https://github.com/ab922530/496-cloud-project*
[18]. *Project's README File: https://github.com/ab922530/496-cloud-project/blob/master/README.md*
[19]. *Bash Script: https://github.com/simeonjmcg/cloudlab-zun-docker*