**Deploying a Backend SQL Server With Cloudlab**

**By: Steve Keashon, Jake Steckel, and Kevin Hampford**

**Abstract:**

For our cloud project we chose to do option three, installation and deployment. This required us to have an in-class demonstration on our selected computing services on CloudLab, or on a local machine. What we chose to have our project focus on was the creation of a web server via docker, and through the docker terminal, create a simple database of our group contact information that we can access and edit. We are then able to view the database on the web browser by visiting the page we created, so it then can be displayed when we want to access the data. Along with docker, we’re also using several other services to achieve our end product. We are using LAMP, which is an acronym of for the Linux operating system, Apache HTTP server, MySQL relational database, and PHP programming language. Before we started our project, we set out several milestones we wanted to achieve. By setting forth interim goals to keep our pace, our visualization of our project became much clearer as we progressed through our steps.

**Introduction:**

The problem we chose to base our project off of and solve was the third project option, Installation and Deployment. To summarize this, our objective is to install and deploy a multi-node computing service on CloudLab’s OpenStack Cloud. Primarily, we are asked to select a computing service consisting of at least two service nodes like a web server with a backend SQL server, or even a Big Data Analytics Hadoop cluster. After selecting our service, we then deploy the corresponding VMs inside CloudLab’s Openstack infrastructure. Aside from our primary tasked, we are asked to create a cloud project that is able to setup VM nodes that work together to provide a unified computing service to users and to move these VM nodes into the Cloud, setup proper network environment, and link them together.

Our idea to for this project was to create a very simplistic database that we can edit and view. So to accomplish that we set our milestones to track and gauge how our project was coming along. They are as follows:

1. Using services on local desktop successfully (deploying sql server, installing phpmyadmin, creating a network then linking the two together on the same network).

2. Accessing our database/modifying it within local machine.

3. Repeat above steps on CloudLab.

4. Automate the above steps.

**Services:**

The services we employed in our project included Docker, Cloudlab, Linux Operating System, Apache HTTP server, MySQL relational database, and the PHP programming language via phpMyAdmin. The use Linux, Apache, MySQL, and PHP all working together is known as the LAMP stack.

1. **Docker**

Docker is a computer program which enables performance involving operating system virtualization. Docker is used to run the containers employed in the project. Docker is primarily developed for Linux.

1. **Cloudlab**

Cloudlab is a facility for building for building and deploying clouds. It enables the capability to experiment with architectures that form the basis of computing programs.

1. **Linux**

Linux is a family of open source operating systems which are typically packed in Linux distributions. A common distribution for example is Ubuntu. Linux runs on embedded systems. It is an open source way for software collaboration.

1. **Apache HTTP Server**

Apache HTTP Server’ instances are mainly ran on Linux distribution, but are versions with Windows and other Unix systems. Instead of implementing a single architecture, Apache provides MultiProcessing Modules.

1. **MySQL**

MySQL is an open source relational database system. MySql is part of the LAMP web application software stack. It is the leading database choice for web-based applications. Oracle drives MySQL innovation, and have created one of the most recognized and widely used services.

1. **PHP**

PHP stands for Hypertext Preprocessor. It is a programming language originally for web development. The code may be executed through a CLI, embedded in HTML, or used in different web templates.

**Results:**

The results of our experiment were not as anticipated. We did not meet every milestone as expected. As stated earlier, we intended to deploy a basic interface displaying our contact sheet, with phone number and names. Our project were unable to display the contacts as hoped, but were able to access PHPMyAdmin and properly display/edit the database. Essentially, I connected these two containers (PHPMyAdmin and MySQL) by creating a network and connected these containers to the network. For PHPMyAdmin, we opened up port 8080 to display the interface of PHPMyAdmin on our web browser successfully. We was also successfully able to open port 3306, the standard mySQL port utilized. However, I was unable to implement the use of an Apache web server to display our database successfully due to technical issues. Executing ‘docker container ls’ displayed the active containers of MySQL and PHPMyAdmin. The preceding steps were executed on my local machine’s installation of Docker. Upon completing these steps, this architecture was recreated on a CloudLab node. Screenshots were taken in order to display the proof of function. Difficulties were encountered with how the project was executed. A poor understanding of CloudLab and networking architecture did not provide for a successful experiment. Upon working on the automation portion, the original Github repo was not receiving the documentation along the way. A new repo has been created, and necessary team and administrative collaborators were invited.