COSC349 Assignment One

Nikki Meadows 7225504

‘GLOBAL GYM’ - A Member Management System

Purpose of each VM:

For the purpose of this assignment, I created a gym membership management system where a user is able to enter their personal information required for a membership, and all of the members and their entered information are able to be viewed in a table. To create this system, I have used Vagrant to manage three virtual machines. Each virtual machine runs using the Ubuntu xenial64 Vagrant Box.

The first virtual machine ‘webserver’ is an Apache web server that runs a PHP web page which provides a form that can be interacted with by a user. Webserver installs Apache and PHP when built. A user can enter their personal information for a gym membership on this webpage and once they click save, a PHP script is run to POST the entered information to the database using a sql insert statement. The second virtual machine ‘dbserver’ installs mysql and is a database with a table to store the information that was entered by a user into the first virtual machine. The third virtual machine ‘queryconverter’ is another Apache web server that runs a PHP webpage that queries the database by using a SELECT statement in a PHP script to obtain all of the members and their information from the database and display the contents in a table. A user can navigate to this page from web page displayed by the first virtual machine by following a link containing the IP address for the third virtual machine. The two web servers do not communicate with each other directly, but instead each communicate with the database server by using the IP address for the database server. The PHP files for each of the web server VMS have been separated into different folders, www/ for the ‘webserver’ and queryconverter/ for ‘queryconverter’.

Approximate build download volume:

The initial download volume for the first build is approximately 270 megabytes for the Ubuntu xenial64 box. After the first build, the Ubuntu xenial64 box is stored locally, and subsequent builds will use the same box file that has been downloaded. On each build, checks will be performed for any updates for the webservers or mysql.

How to use the application:

Prior to using this application, Virtualbox and Vagrant need to be installed if they are not already. Vagrant can be installed from: https://www.vagrantup.com/downloads.html and Virtualbox can be installed from: https://www.virtualbox.org/wiki/Downloads

After this, please clone the repository at: https://github.com/meadows115/COSC349-Assignment1.git

Once this has been cloned, open a terminal and navigate to the cloned folder, and use the 'vagrant up' command. The initial build will be approximately 15 minutes and subsequent builds will be approximately 11 minutes depending on your computer. Once the build has completed, open your web browser, and navigate to http://192.168.2.11.

From here, please complete the form to be added to the gym membership database. If you do not want to complete the form, you can click the link titled "view all members here". If you have chosen to compete the form, once you click the save button an alert will pop up to inform you that you have been successfully added to the database. You may now click the "view all members here" link, and you will be redirected to a webpage displaying all the members that are currently installed within the members table in the database.

Once you are finished using the application, navigate back to the terminal and type 'vagrant destroy'. You will have to confirm this command for each running virtual machine my typing 'y' and pressing enter/return. If you wish to run the application again, please use 'vagrant up' again.

Future improvements to the application's code and how to rebuild and rerun after these improvements:

1. An improvement to the existing application would be to assign a unique ID to each member that signs up in order to be able to uniquely identify users within the database and a search function on the members web page to be able to search for an individual member. To make these changes, the database VM would need to be updated to store the unique ID, and the webserver VM index.php form would need to be modified. Additionally, the second VM webpage table would need to be updated to show the unique ID. Before implementing these changes, destroy the three VMs using the vagrant destroy command. Once the changes have been implemented, use the vagrant up command to rebuild all three VMs.

2. Another improvement to this application is to add in more fields to the sign-up form for specific payment and banking information and update the database to store this information. To do this, the webserver index.php form will need to be updated to add extra text fields, and the database VM table will need to be modified to store this additional information. When making changes to the php files, there is no need to use the vagrant up command as these changes will be automatically reflected on the webpages once that pages are refreshed. Once the changes have been made to the database, the machines can be rebuilt using the vagrant up command.

3. Another possible improvement is to allow a user to update their information if it changes. To do this, the search function mentioned in improvement 1 would need to be implemented to bring up that specific member, and they would then need to be able to return to the form on the first webpage to update their information. Therefore, an UPDATE query would need to be implemented in the first webpages php script to update the information stored in the database. Once these changes have been made, the machines can be rebuilt using the vagrant up command.

4. Lastly, an ideal improvement would be to only allow gym employees to access the members database using login credentials so that all users are not able to navigate to the page containing the information for all members that have signed up. Once these changes had been made to add a login feature for employees to access all member information, the machines will need to be rebuilt using the vagrant up command.

References

- Starting code:

  David eyers Gitlab repository https://altitude.otago.ac.nz/cosc349/vagrant-multivm

- Starting code for the html form used in index.php:
  https://www.w3schools.com/tags/att_form_method.asp

- Setting up the POST method for the members to be sent to the database:

  http://people.cs.ksu.edu/~h ankley/d764/tut06/GopisettyPHP.html