COSC349
Assignment 1 Report
04/09/2020
Darcy Knox
7712753

NBA Greats Game Simulator

For this assignment I have designed a basic game simulator using NBA basketball players selected into two teams by the user. The application uses three virtual machines that interact with each other.

User interaction

If not already done, locate a directory to put the application into and enter the command:
$ git clone https://github.com/darcyknox/349-assignment1.git

Cd into the 349-assignment 1 directory and enter the command :
$ vagrant up

Once the virtual machines are created, go to the address http://192.168.33.10/ in a web browser. This is the private network IP.

This page has a table with several NBA players. The user should select any amount of players into two teams (Team A and Team B) by clicking in the checkboxes in the row of the player they want to select, and the column of the team they will 'play' for.

Once the user has picked their team, they will click the submit button and a link will appear, as well as a list of players in their respective teams. The link will lead to the results page.

The results page displays the skill ratings of the selected players. The sum of these ratings determine the winning team. The user then clicks the 'See results' button to see which team wins. This is the end point of the application and the user can choose to follow the link to return to the player selection screen if they wish.

Implementation

I used the configuration file from Lab 6 (vagrant-multivm) as a template for my config file. I have used three VMs in the application, two are web servers and one is a database. Both web servers interact with the database in a sequential manner, the first web server pulls the data from the database to display to the user. The users selection of players as well as the form

submission will have the web server push data back into the database (SQL insertions). The second web server interacts with the database by pulling the data from the database that has been inserted by the user from the first web server. Essentially, web server 1 passes user input to the second web server indirectly through the database server.

The application downloads apache, php, and sql packages. Build time is usually around 4 minutes (5 for initial build).

The VMs are separate from each other mainly for security reasons. This is a very barebones version of what could be a larger application. I think that this model could be improved to be an online game simulator where players pick their teams in real time and make choices at game time that determine the winner. For this to be secure, the three servers need to be separate to prevent cheating by players. E.g. malicious players may alter the database to make their players more skillful, players may change their teams at game time, etc. This is only an issue for the non-existent, extrapolated, and much improved model of the game simulator.

Review/feedback

NB: I initially intended to pass the user input from one web server to the other directly. I tried using AJAX and http push/get/fetch requests. I wasn't able to implement this. I struggled with CORS and other permission issues. I think if I had more knowledge about web technologies and networks I would have been able to find a way around this, but it ended up being too large of an obstacle.

Using http requests (as stated above) could be one way that a user could improve on this model. I would have liked to have styled the user interface. I think the two data files could probably be merged into one to save a little bit of space. Had I had more time I might have added in a feature where the user can enter the name and rating of their own player, or where the position of the players mattered and a team had to satisfy conditions on how many of each position a team must have. I think the application can be improved in several ways, and given more time and expertise, I would have done it.