COSC349 Assignment 1 Report

Dan Skaf 7138265

Hayden McAlister 1663787

Project overview

This project has three virtual machines (running ubuntu/xenial). One is hosting a user-facing web-server, port forwarded for access on a local machine. Another is hosting a MySQL database, for storage in the project. The final VM is running an admin-facing web-server, for ease of administering the project with a webpage gui.

All VMs are protected by UFW firewall, to block requests on ports that aren't needed on a VM by VM basis.

The project itself is based around a game of minesweeper, implemented in Javascript in the userfacing site. The scores of this game are sent to the database server upon each game being completed. The admin server can interact with this database to examine information about these games.

We have made each of these services on separate VM's, as this can achieve better security. By only allowing communications on certain ports we can have better understanding and filtering of traffic to and from each virtual machine. This means each service is better protected than if they were all on one machine. Using three separate virtual machines also means we can abstract away implementations of each service more easily, so if we change the admin server to run a windows virtual machine and Apache, it will have no impact on the function of the rest of the project. Finally, using three virtual machines as opposed to one means we can scale each service independently, so if there is a lot of load on our user-facing web-server we can simply scale up (or duplicate) that one virtual machine and use all of the extra resources for this one application as opposed to wasting many resources on the services that do not need them (like the admin server).

Component Downloads

Clean build

Git cloning this repository will create a folder of size ~500kB in size, so watch out for that! Of course, running `vagrant up` for the first time will Download/install a lot data initially! Here is a quick breakdown of downloaded data for a clean build:

- `Ubuntu\Xenial`: 133 MB
- web-server: 26 MB
  - `Ubuntu` security patches: 19.5 MB
  - `Nginx` packages: 6.5 MB
- Database Server: 37.5 MB
  - `Ubuntu` security patches: 19.5 MB
  - `MySQL` packages: 18.0 MB
- Admin Server:
  - `Ubuntu` security patches: 19.5 MB

- Nginx packages: 6.5 MB

Subsequent Deployments

After the first vagrant up, it appears that no further data needs to be downloaded.

Changes to this project

If you want to use this repository as a base for your own web development project there are several changes you can make.

Changing the game

If you decide to implement a different game (other than minesweeper) you can very easily alter the files www/game.php and www/game_logic.js to change this. As long as you return a score to the database upon completion of your game then nothing else should need to be changed.

For example, you could change the game and game logic files to implement Tetris, even keeping the score as a time.

Changes like this should not need a reprovisioning of the virtual machines, just a refresh of the webpage.

Starting fresh

If you decide to alter the entire project from the virtual machine up, simply changing the contents of the www, database-setup, and adminwww folders would allow you to implement your own web-servers and database. This would mean you could use this projects provisioning and firewall security for your own deployment and focus instead entirely on the web development.

For example you could implement an online store using the platform we have set out by redesigning the folders above.

Changes such as this may need reprovisioning of the virtual machines, especially if site names change or if the database is altered.

Updating Firewall rules.

The typical ports for a LEMP stack server are the only open ports in the firewall at the moment i.e 80 - http, 22 - ssh, and 3306 - mysql. We decided to provision the firewall using a bash-script for each virtual machines. This ensures modularity to the provisioning and makes it easy to modify the rules if a new service is whished to be created or modules like SSL or virtual host redirection are added to the web-server. This file can simply be edited per virtual machine as desired.

Changing the web-server

If you wanted to change the web-server software from nginx to, for example, Apache, this could be done by altering a few simple lines in the vagrantfile and altering the usersite and/or adminsite files. This would also require reprovisioning or perhaps recreating the virtual machines, but if a developer wanted to do this it is entirely feasible.

By default the web-server document root is at `/var/www/html` and the files are copied to this location from the `/path/to/project/www` and `/path/to/project/adminwww`. Form the nginx config files `/path/to/project/adminsite` and `/path/to/project/usersite` this location can be changed and further configuration modules can be added here. As there is a separate server for each site, the configuration files for each must be changed accordingly, again preserving modularity incase modules or extensions only wished to be added to certain sites.

Problem Points

- Conflicting versions of Vagrant(2.2.18)/VirtualBox(6.1.26r145957) on Windows machines often required vagrant destroy to successfully run vagrant up again
- Learning PHP, some JS, and AJAX caused many headaches
- Securing SQL queries using prepared statements was an adventure, to stop SQL injections from malicious users
    - Some queries could not be secured with this method (adminsite/tables.php has a query that uses POST variable to set table we query, not easy to parameterize)
- Use of conflicting SSH agents on Windows appears to hang vagrant when trying to establish the SSH interactions for virtual machines. This was solved by killing the SSH-agent on windows.

Credits

- A large portion of this project has been developed based on repurposed files from COSC349 labs (link working: July 2021). A massive credit goes to Dr. David Eyers, who provided these resources and allowed for them to be reused. Specifically, large parts of the `vagrantfile`, and `webserver.conf` file are based on these lab files.
- Virtual Machines in this project run `Ubuntu/Xenial`.
- Web-servers are using Nginx.
- The styling for the webpages for this project was aided by SkeletonCSS
- Icons are taken from icons8 with license for personal use