Francesca Kraayvanger 1150149

COSC349 Assignment One:
Deployment of Software Applications via Virtualisation

Virtual Machines Descriptions:

This ‘portable’ software application is a basic management system for an adoption facility. This application will allow customers to have easy access to view the dogs currently being put up for adoption. The admins have the ability to enter another site, specifically for them, that allows for them to view any additional information about the dogs that are up for adoption.

The basic management application contains three virtual machines, in which two of the virtual machines(VMs) contain web servers, and the other being a database server. The database server allows for the two web servers to have easy communication. The customer web server contains brief information about the dogs, with name, price and age being the only viewable data. Whilst the admin web server contains the ability to view any other additional information, with access to not only just the name, price and age but any other data that is contained within the database VM.

Build Process:

Download times: ~1 minute 43.59 seconds
The file size may vary depending on changes made, but the file size within the current folder is roughly 170KB.

Reasoning for Three Virtual Machines:

→ The use of three VMs is needed for this application, the two web servers (admin and customer-facing) need to be separate in such a way that there is no crossover. We want to liit the access customers have to the website, keeping it restricted to one VM allows for that to occur. Whilst the second VM for the adminserver is functional in the fact that the admin will not need to access the customer-facing site to retrieve any information, but rather jus the admin website.
→ The database helps hold all the dogs' information and having two web servers with different outputs of the data make the database VM a need for this project, otherwise any updates to the adoption roster (dog_database.sql) will not be applied to the web servers. Making the web servers unusable.

Using the Management Applications:

In order to be able to have access to this application, you will have to have Vagrant installed, which can be accessed from the Vagrantup website (I do recommend using the same version as which was used for these VMs, 2.2.18):
https://www.vagrantup.com/downloads

Francesca Kraayvanger 1150149

Alongside Vagrant, in order for the application to be used you must have the most recent VirtualBox version installed, which can be accessed from the VirtualBox website (I do recommend using the same version as which was used for these VMs, 6.1.16):: https://www.virtualbox.org/wiki/Downloads

After having downloaded the above content, the repository will have to either be downloaded or cloned into your own repository. The cloning can be done via the terminal, with the command: *git clone https://github.com/fkraayvanger/COSC349_AssignmentOne* or downloaded via pasting this URL: https://github.com/fkraayvanger/COSC349_AssignmentOne/archive/main.zip

After the project has been cloned or downloaded, find the directory in which it has been downloaded or cloned into --> Use the command: vagrant up. This command will launch all three Virtual Machines and allow full access to the VMs.

To venture into the customer facing website, head to:

→ 192.168.2.11 This website has two sections the HomePage and the DogPage, the DogPage contains the information from the database about the adoptee dogs.

To venture into the admin facing website, head to:

→ 192.168.2.13 This website also has two sections, HomePage and DogPage, but the DogPage contains additional information about the dogs in comparison to the customer facing website.

Two Potential Modifications:
This project could benefit from customer interaction, with the ability to enter their email and the name of the dog they want to meet being passed from the customer facing website to the admin facing website. This would allow for the admins to view the customers that are willing to adopt. This would be done by the use of another database, which would hold all the necessary data that has been feed by the html form on the web server. In order for the form to be created you must insert the code in the dogs.html file. Much similar to the following.

Another modification could be to do with the potential for an image section within the dog_database.sql dog_options table. This would allow for the customers and admins to view the dogs without having to be surprised when they meet them. This could be done via the use SQL, with using the Bulk command followed by the directory path. This change would be within the admin and customer facing servers.

Delay in the Submission:

Francesca Kraayvanger 1150149

This project was rather difficult to try and accomplish at home, my laptop which was sent to get fixed before Level 4 was not sent back to me until a few days before the project was initially due. I was using my flatmate's laptop for the majority of this assignment, at the beginning I could not use their laptop because of their own University work, but I managed to start the assignment. It did not function well with the mass downloads I was performing but it did manage to make the database server function and the customer server function, in which I went to the Lab on Monday to finish the presentation of the admin web server and database server.

Debugging:
The process of debugging this project consisted of back and forth re-creation of the VMs, as well as close looks at the default vagrantfile (from gitlab){1} and my vagrantfile. Using the database with a VM was perhaps the most difficult to debug, with a lot of the time being that the database tables do not get fed into the dbserver.

Resources:
{1}https://altitude.otago.ac.nz/cosc349/vagrant-multivm/-/tree/master → To help create the default VagrantFile.