COSC349: Assignment Two Report
Deployment of Software Applications via Virtualisation

AWS Virtual Machines:

An Amazon Web Services (AWS) virtual machine can be based on the EC2 service they provide. This EC2 virtual machine (VM) allows users to create a VM on the cloud, which can be automated by the use of a downloadable software called vagrant. In which the users edit the vagrantfile, which is downloaded via the use of the command *vagrant init,* allowing for VM automation through the use of AWS.

Within this assignment, I followed the same website and database design as assignment one, it being called Dog Museum in which users and administrators can have access to a database with potential dog adoptees. For the customer-facing site, I used an EC2 VM with vagrant automation, having the VM contain the PHP files necessary for the website. The admin-facing site is also an EC2 VM and automated through vagrant, with an extended feed of the database. The database has been created through a Relational Database Service (RDS) instance; which has not been automated through vagrant. An additional site has also been added using S3 to create a static website for the adoptee dog photos.

Access AWS session:

In order to access your AWS session via terminal, use the following commands within the terminal. Session keys are found within the https://labs.vocareum.com console under the account details button. It should look like the below screenshot. Do not share these details.

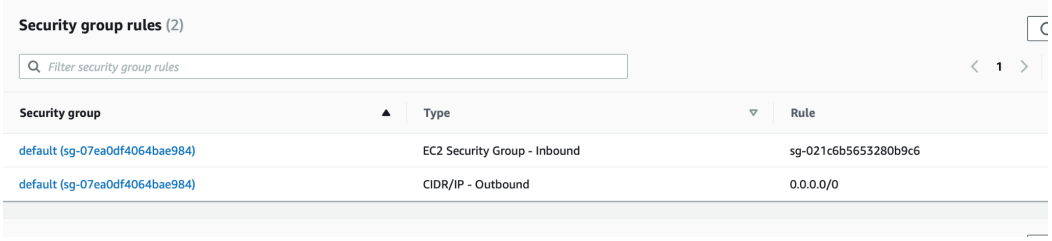

```
export AWS_ACCESS_KEY_ID= <insert your sessions aws_access_key>
export AWS_SECRET_ACCESS_KEY= <insert your sessions aws_secret_access_key>
export AWS_SESSION_TOKEN= <insert aws_session_token>
```

Virtual Machine One: Customer-Facing Site

VM1 contains the webserver for the customers, which displays a functional website with two navigation tabs. This webserver has been created through the use of a vagrantfile, for this to function download both vagrant and as well as the following commands on your cmd line:

```
vagrant plugin install --plugin-version 1.5.11 nokogiri
vagrant plugin install --plugin-version 1.0.1 fog-ovirt
vagrant plugin install --plugin-version 0.2.0 dry-inflector
vagrant plugin install vagrant-aws
```

Once you have downloaded the necessary software and packages, use *git clone* *https://github.com/fkraayvanger/COSC349A2* to have access to the necessary files. The vagrantfile for this webserver is found under the VM1 file, alongside the PHP files. You may have to create a new keypair, if the file is not included, in which change the private_key_path and keypair_name to the new keypair. After these changes, use the command *vagrant up --provider=aws* (you must have access to an AWS account) through your terminal which is cd'd on the VM1 directory. The VM will begin to start up within your EC2 instances. In order for the website to appear, *vagrant ssh* into the VM via your terminal, and *sudo -s* into the root user. Head into the /var/www/html directory and remove (rm) the index.html file, in which then copy the index.php and dogs.php into the html directory. The webserver should now be functioning. To access the customer-facing webserver, type **http://34.203.77.133** into your browser.



Security group rules (2)

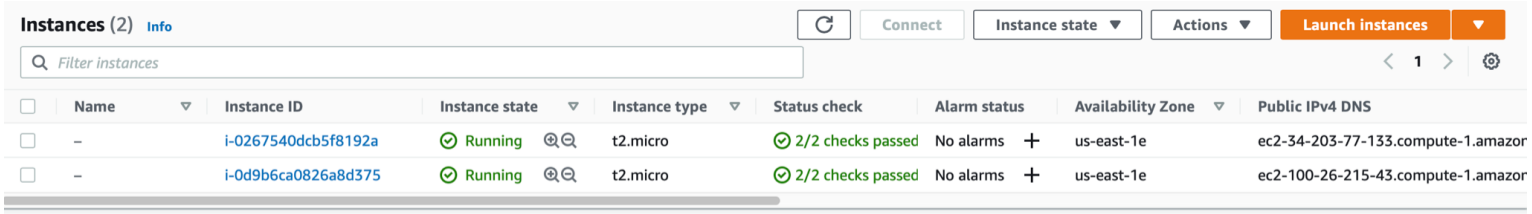
Filter security group rules

Security group	Type	Rule
default (sg-07ea0df4064bae984)	EC2 Security Group - Inbound	sg-021c6b5653280b9c6
default (sg-07ea0df4064bae984)	CIDR/IP - Outbound	0.0.0.0/0

Virtual Machine One: Admin-Facing Site

Folder VM2 contains the webserver for the admins of the site, meaning these users can have access to the extended database presented within the admin site Dogs tab. The creation and use of the webserver is the same as the above VM, except this VM is found within the directory named VM2. To access the admin-facing site, which admins are the only users with access to the address, type **http://100.26.215.43** into your browser. The admin facing site contains more than the basic customer site, with access to more information about the dogs as well as a link to the S3 static website containing the dogs images. To get

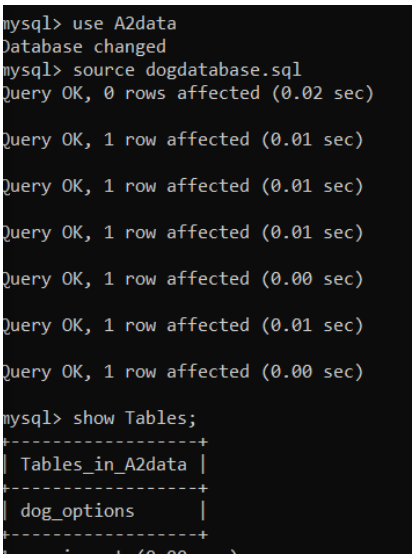
this VM running, use the *vagrant up --provider=aws* command within the VM2 directory.



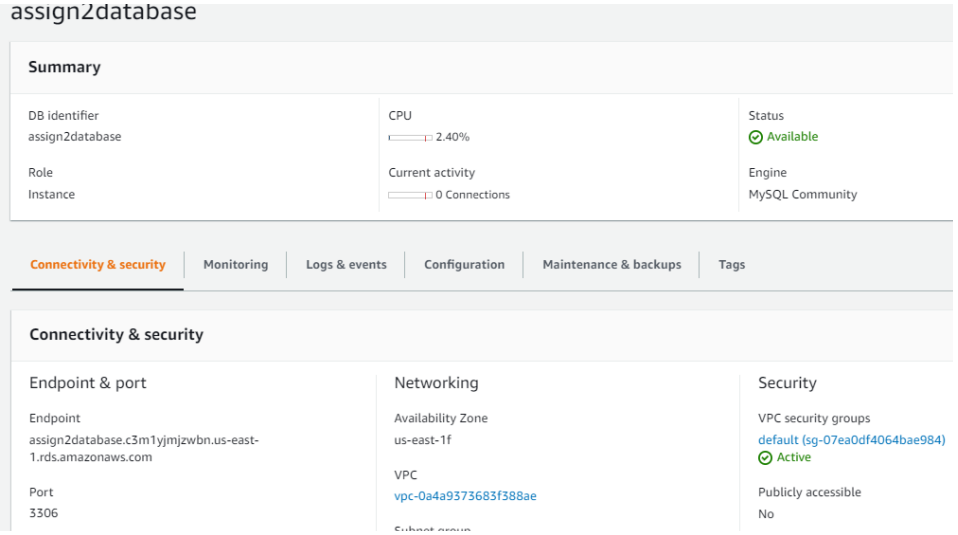
RDS Database:

The use of this service is also within the AWS site, however, this database does not have an automated startup. Meaning if you are trying to create these servers, you will have to set up a basic RDS; follow the instructions within this link for a basic setup: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateDBInstance.html)

Once the RDS Database has been started, head into the terminal which is running the VM1 and *vagrant ssh* into the VM. Once you have accessed the VM, download the mysql package: *sudo apt install mysql-server.*



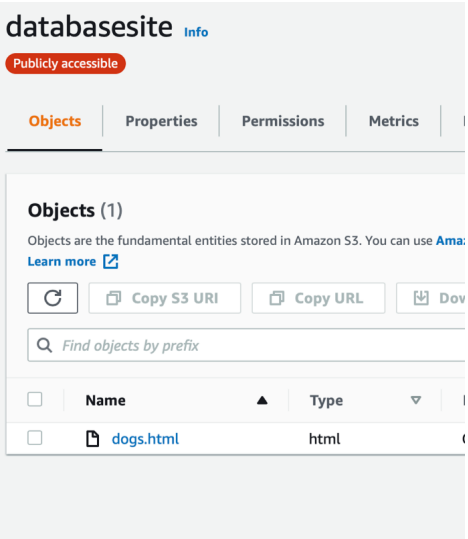
After the mysql package has been downloaded, scp the dogdatabase.sql file into the instance, and then use the command *mysql -h assign2database.c3m1yjmjzwbn.us-east-1.rds.amazonaws.com -P 3306 -u admin -p* to gain access to the RDS database within the VM terminal. In order to input the database queries into the RDS, use the commands *use assign2databse* and *source dogdatabase.sql.* The RDS has now been set up.



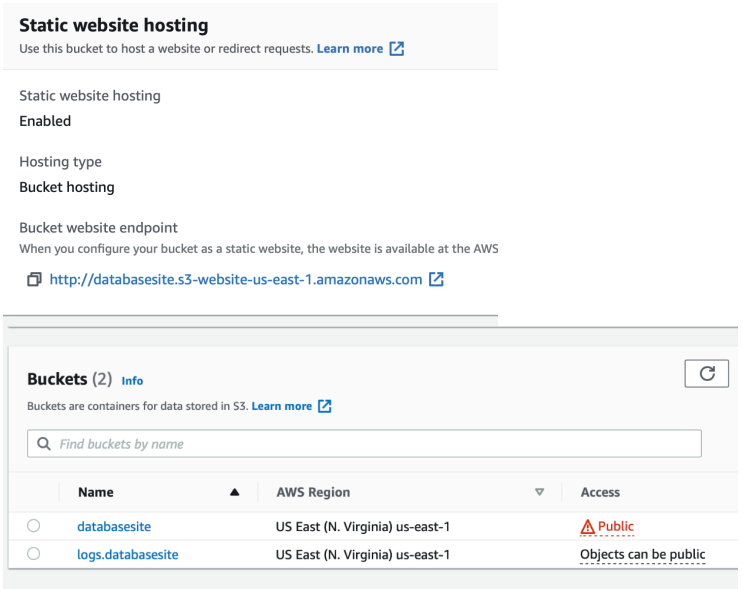
With the PHP code within VM1 and VM2 change the PHP db_<blank> lines to the information from your setup RDS.

S3 Static Website:

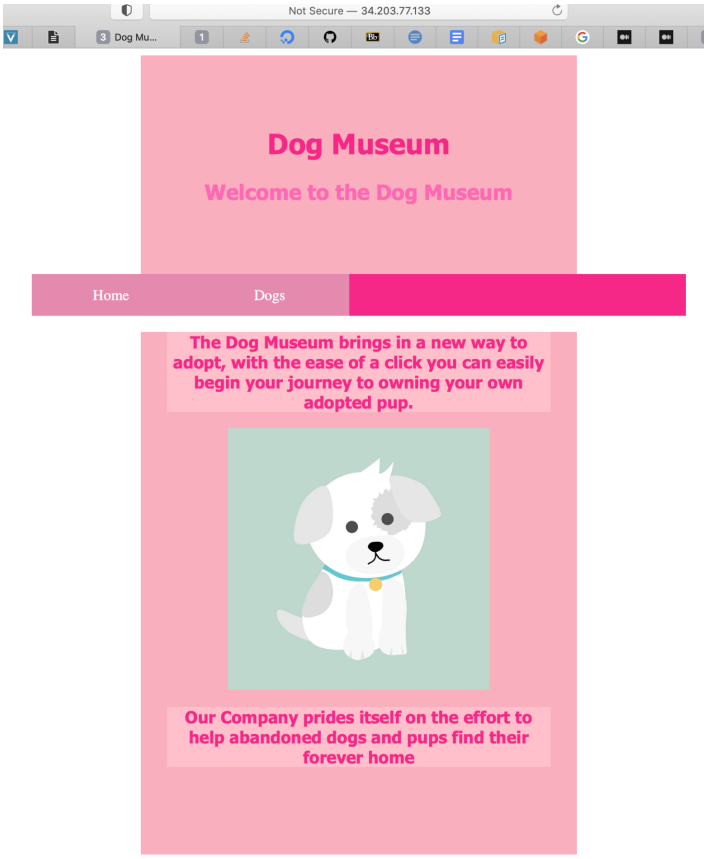
This S3 bucket was formed by following the lab 8 material. Once you have set up your S3 bucket, instead of inputting the index.html and error.html files use the dogs.html file provided within the S3 folder. This bucket is linked to the admin facing site (VM2), and displays the adoptee images for the admins. The following screenshots provide the basic information behind the S3 bucket. Follow this link to access this site:
http://databasesite.s3-website-us-east-1.amazonaws.com/



Follow this link to view the dogs

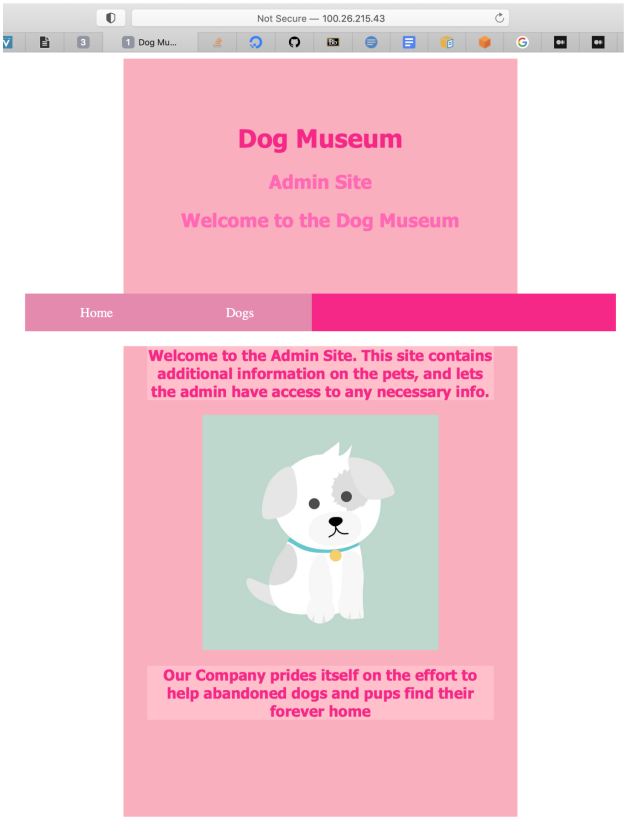


Presentation of the Costumer Facing Site:



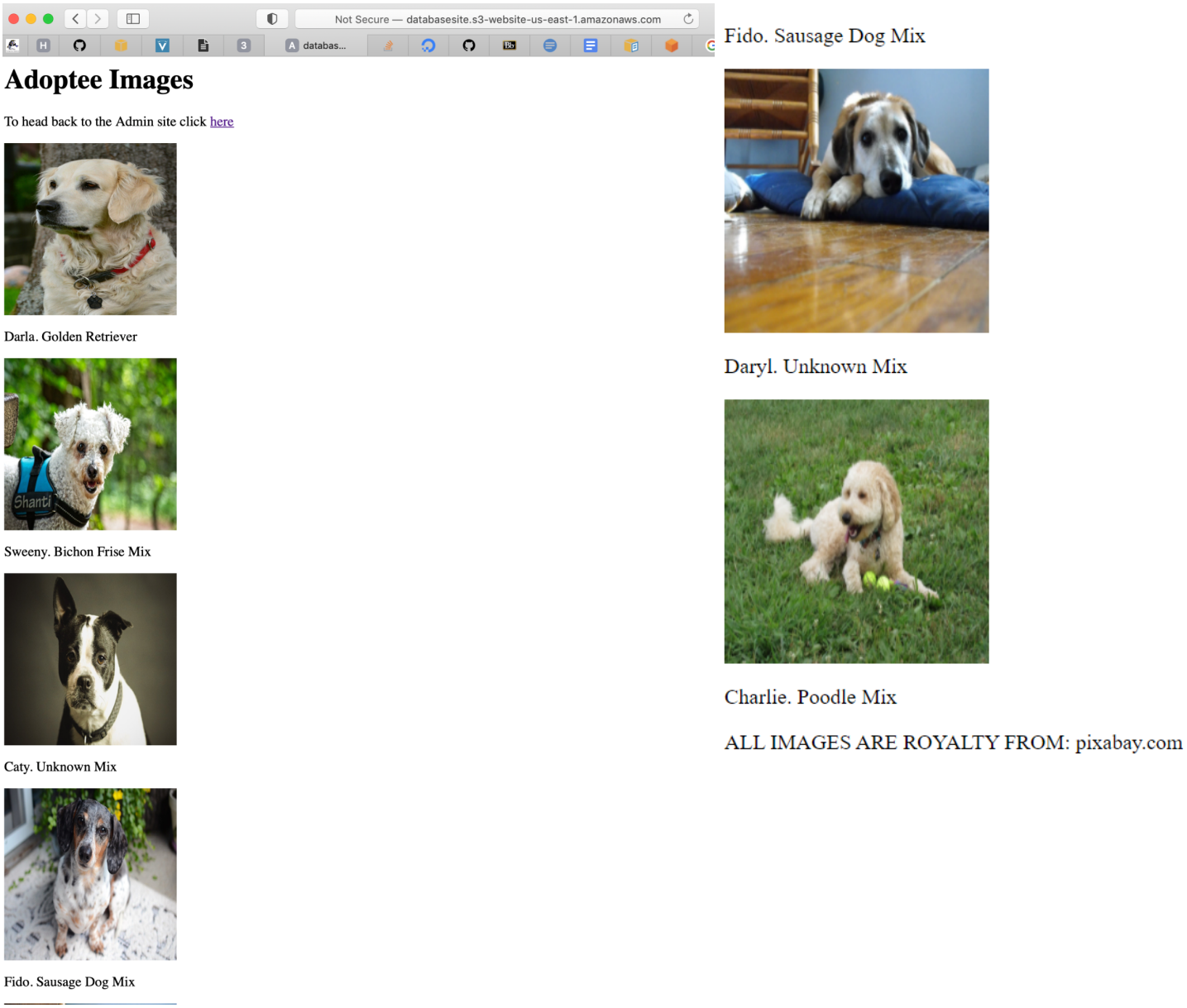
All images are royalty-free from: pixabay.com

Presentation of Admin Facing Site:



All images are royalty-free from: pixabay.com

Presentation of S3 Static Website:

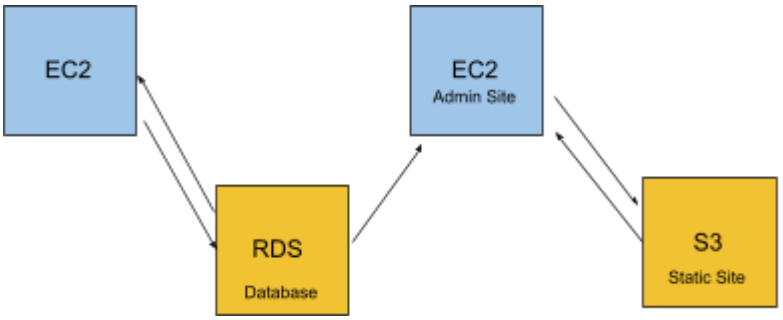


Use of non-EC2 Cloud services:

The two non-EC2 services I used were RDS and S3, with the database for the adoptee dogs being stored within the RDS. This service is necessary for the functionality of this site, as any changes to the database will be displayed within the webserver. Meaning if the RDS was not included within this project, the admins would not be able to modify any of the data entries, meaning the website would not be updateable.

The other service that was implemented was the S3 bucket. The S3 bucket was created to host a static website that displays dogs images for the admins. This

website is accessible through the admin sites dog page, meaning that the only users that have access to the static site are the admins. This service lets the admins view the images of each dog on a parallel site. The inclusion of an outside site creates better flow within the website, meaning that the admins do not have to deal with an inefficient site. The choice of the S3 bucket was due to the want for an additional site without the need to boot or use an EC2 VM, as well as a decrease in cost. However, the site is static and I would have preferred the site to have access to a database containing these images. Perhaps with an additional database within the RDS.



Cost of each Service:

I do not have permission to access the costs reports. However, I used the AWS cost calculator and provided information to form a rough estimation. Initially, the RDS size was db.m6g.large, however, I have changed this to db.t3.micro for cost reasons.

RDS

- db.t3.mirco costs $0.017 per hour. With a daily cost of $0.408.

EC2

- Both EC2 instances use the t2.micro for their instance types.
- t2.micro costs
- The use of these EC2 instances costs $0.404 per day, with an hourly rate of $0.0168.

S3

- Due the hosting of this site being under the bandwidth restrictions, it currently does not have any cost. Meaning $0.00 per day.

Overall, if these services ran for upwards of a month it would cost roughly ~$24.36 for each month, with the main costs coming from the EC2 and RDS services. However, this an estimation for a site with currently no visitors.

Reasoning Behind Late Submission:

At the beginning of quarantine, I fell into a depressive episode. After quarantine was lifted I was unable to come out of my slump and became worse due to family issues, in which I reached out to a psychiatrist. After a few sessions, I was diagnosed with ADHD and general anxiety; in which I began medication during the weekend. It has been an odd adjustment period, and I am grateful for this time allowance as it has allowed me to finish the project with much more detail and helped with the anxiety about my current situation. If needed, evidence of prescription can be given. If you need any more information about my situation, please send an email.

Screenshots of EC2 Instances: