Minh Tran – Student ID: 6355049
COSC349 - Assignment 2

REPORT ON MULTIPLE VIRTUAL MACHINE INTEGRATION USING AWS CLOUD SERVER

Quick Shopping Application

Commit: 902d0dbdf740b7062c85db348412177ef1f99c9c
https://github.com/minhtran822/Quick-Shopping/commit/902d0dbdf740b7062c85db348412177ef1f99c9c

Introduction

Quick Shopping is a webpage simulating the online shopping experience that allows user to select products and add to cart. The current webpage only allows adding to cart as the project focuses on illustrating interactions between different servers using various Cloud services. For this project, we were granted access to Amazon Web Service (AWS) as Cloud infrastructure for us to host our projects. The skeleton and the layout of the webpage was sourced from Simple Php Shopping Cart (https://phppot.com/php/simple-php-shopping-cart/). The project is created and uploaded onto Github directory.

Application Description

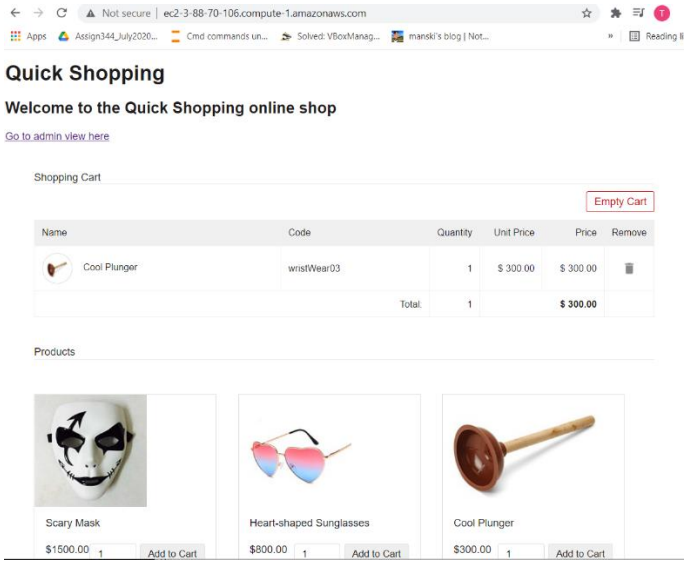
My project Quick Shopping consisted of 2 web servers and a database server. The web servers are hosted on Amazon Elastic Compute Cloud (EC2) Virtual Machines (VM) hosting a web server each. The database server was stored on AWS online service called Relational Database Service (RDS). The web pages components are generated from the data stored in the database, with the addition of images stored in Amazon Simple Storage Service (S3).

The web servers consist of Public Web Server and Admin Server. The Public Web Server is responsible for displaying and handling customer request, which currently are displaying existing products and allowing the customer to add items to a cart. The Admin Server is for the administrative to carry out core functions to manage the data, such as adding new items to the database. These two web servers have direct access to the internet, and they can directly connect to the database. The database does not require a VM server as it is stored in the Cloud in RDS service. It is not connected directly to internet; it can only be modified through the web servers. The images are stored in the S3 and references by their URL in the database. The links below can lead directly to the web pages.

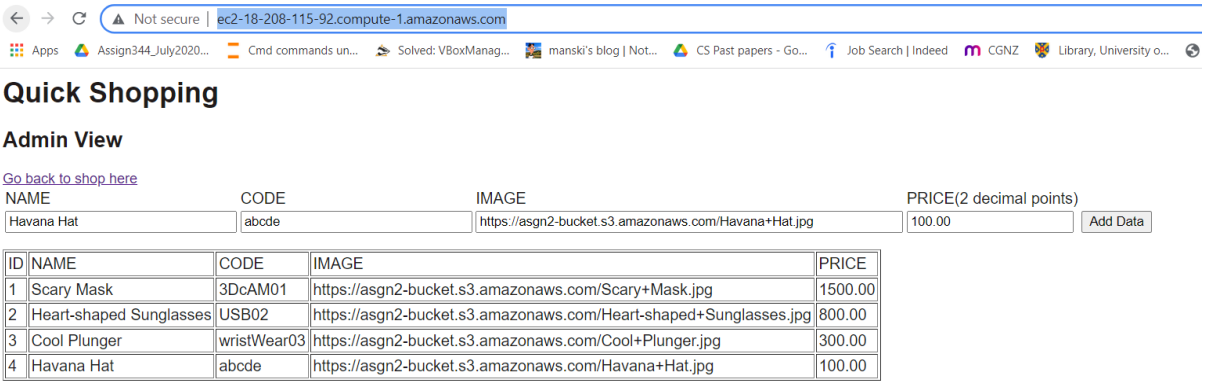
Public Web Server: http://ec2-3-88-70-106.compute-1.amazonaws.com/

Admin Server: http://ec2-18-208-115-92.compute-1.amazonaws.com/

Minh Tran – Student ID: 6355049
COSC349 - Assignment 2



Screenshot 1: The Public Web Server - http://ec2-3-88-70-106.compute-1.amazonaws.com/



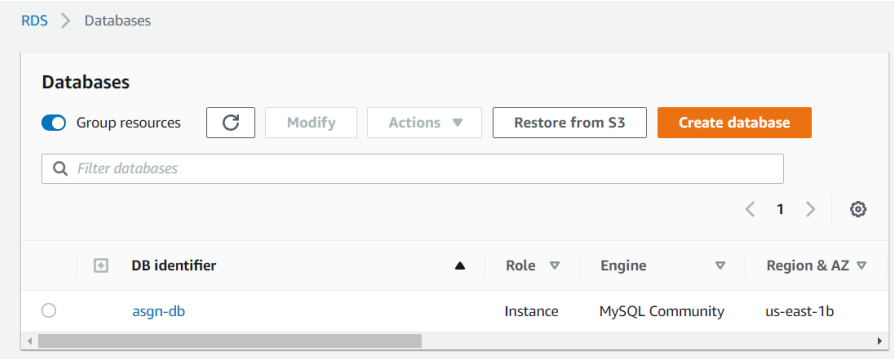
Screenshot 2: The Admin Web Server - http://ec2-18-208-115-92.compute-1.amazonaws.com/

Project Development

During the development of the project, I first started out with constructing the subnets in the Virtual Private Cloud (VPC) following the tutorial provided in the lab during the course. All the machines and services are connected to the same VPC. The VMs were configured identical to each other using the same security groups and the same subnets. The subnets that they are configured with are public subnet, which attached directly to a route that allows them to connect to Internet. Their security groups allow them to connect to the HTTP port for displaying webpages and the SSH port for modification of the VMs. The database, on the other hand, is in private subnet that prevent it from being accessed over the internet, and its security only allows connection through MySQL. The VPC is also free for estimation.

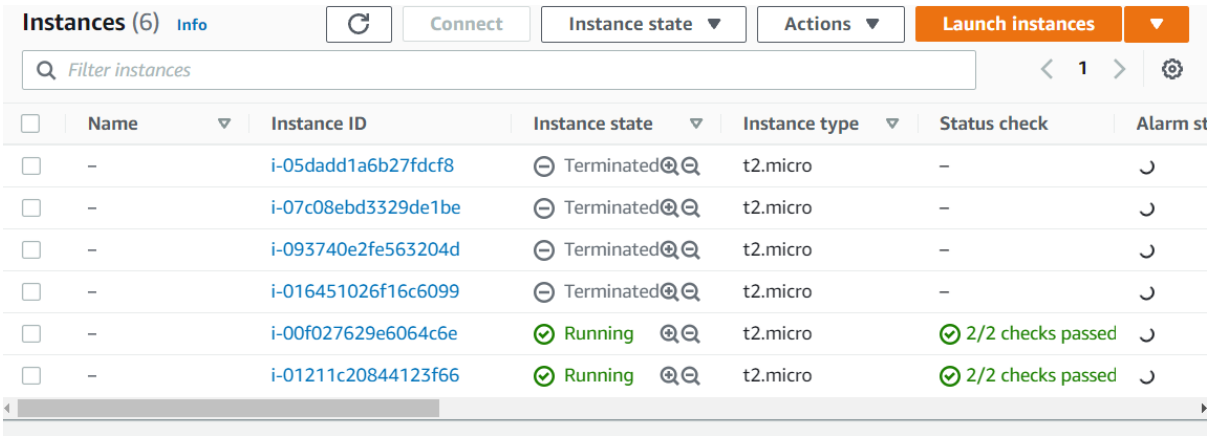
Having established a VPC configuration, I then created a database on RDS, which can be seen in screenshot 3 . Having a Cloud-hosted database allows the database size to scale easily. The current

project only storing a few products, however, the nature of Quick Shopping and online shopping in general requires great products database to store new products. In addition, although not illustrated in this project, database for online shopping also need to store authentication, customer information as well as the sales information. Against a physical database server that may be time consuming or disaster sensitive to allocate more resources, cloud service can easily give more virtual storage. Therefore, RDS is a good choice to give scalability to the growing database. The current idling cost with 1 hours of on-demand storage is 2.82 USD per month, and the cost of running is 6.44 USD with 8 hours of demand



Screenshot 3: The asgn-db database stored on RDS

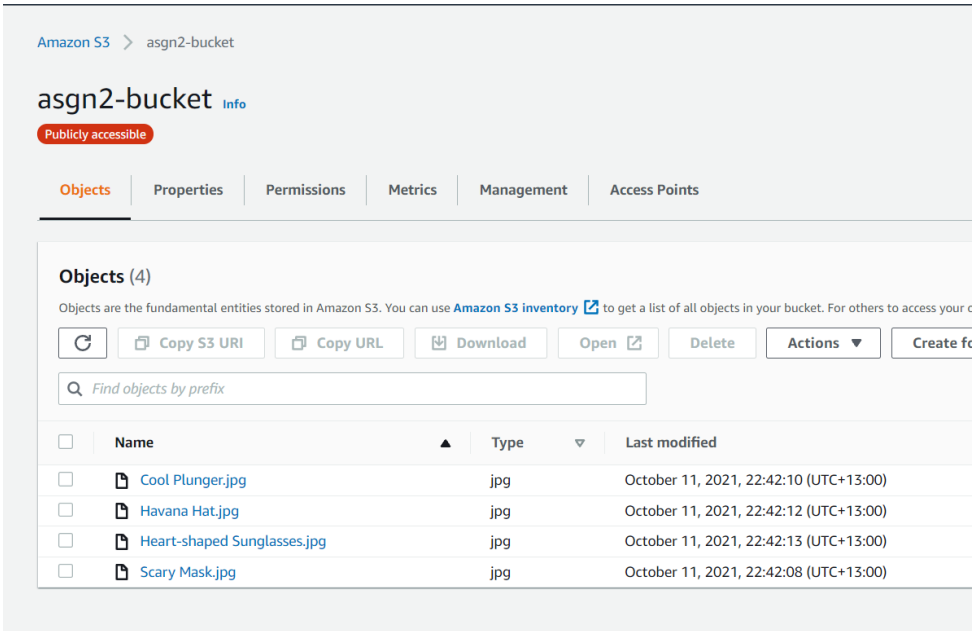
The EC2 are created following using vagrant. The instances of the EC2 were configured in the Vagrantfile that can be found in the Server directories in the repositories. The template to create a EC2 VM from vagrant was lab 9 of the course (https://altitude.otago.ac.nz/cosc349/lab09-vagrant-aws). Using vagrant can aid in automation in building of the VM, however, in order to boot up the VM through vagrant, developers have to configure the credentials to access AWS resources, and the Vagrantfile was created using Windows 10 in 2021, other operating systems and later versions may require updates and maintenance. I have deployed the VMs through the student account granted with the course. Regarding EC2, as a cloud service like RDS, EC2 also aids the growing in usage. EC2 is the infrastructure as a service to host and run the server live through internet, which can prevent the physical limitation in shopping traffic. Having an online shopping web page, we can expect to have large traffic spike during discounts or during holidays, which may need more connection bandwidth, and EC2 can quickly allocate these resources during the season. Resource which grows seasonally are also reduced when the traffic decreases, and hosting on EC2 can minimise cost of running the website. The idling cost of having 2 EC2 is 12.11 USD per month, and the cost of running with extra VM is 20.06 USD



Screenshot 4: 2 deployed EC2 instances running

The design of the project also aims to aid in reducing traffic. Having an Admin server responsible for management functions will separate users into customers and the staff, which can reduce the usage on one machine during the sales season and allow staff to modify or maintain the system without having to shut down the whole machine.

Lastly, I also used S3 to store images files for the system outside of the database. S3 store files online that can be used through URL reference. Since Quick Shopping is an online shopping web application, it requires a lot of images to preview the items which can scale exponentially with the number of items, therefore, storing the images as BLOB file in database will be very costly when the database grow, in contrast to only store a URL link. The current project has manually uploaded the images into S3 through the AWS Console. Further development regarding this service can look for a way to write a PHP file to upload and sync the files from the user directory. Currently, there are only 4 objects stored in the s3 storage, and therefore the cost is minimal to none.



Screenshot 5: The asgn2-bucket on S3 to store images

The rough estimate ongoing cost of idling system for Quick Shopping is 15 USD and it goes up to 26.50 USD when it is running. However, this rough estimate is calculated on the smallest and cheapest EC2 machines, and minimal storage in S3 and RDS.