VIGNESH MURUGAN(S3822789)

Rationale:

Here we are building a website for a Wholesale seller company. They need a platform to organize and interact with the businesspeople. It is a startup business and they need to cover the investors, retailers, entrepreneurs across the globe through online medium. The website which we are developing through the AWS services will make the process easy. AWS provide every service to build a website and maintain the multiple servers in low cost. Due to this reason I preferred to build the website using the AWS services. I used S3, EC2, and Route 53 services to create a website which will list the products available, the information regarding the company and additional features. I chose these three services because this fulfilled my requirements. EC2 is the main base for the website which is used as webserver. The S3 bucket is used as a storage device to store the images/videos which is used in the website. The Route 53 is used to provide a registered DNS name rather than a DNS name generated in EC2 by AWS. There are many alternative ways to build for example by using LightSail or by CloudFormation which is more complicated and expensive compared to the services which I used in my website.

Installation manual:

Steps to create S3 bucket and upload the files

- Open S3 service from Amazon console and choose **Create Bucket.**
- Enter the bucket name and click **next** twice.
- In set permission, checkout the **block all public access** option and acknowledge it and then click **next.**
- Choose **Create bucket.**

- Open the created bucket and select **Upload** and choose **Add Files.**
- Select the image and video files from the local drive and select Upload.
- Select the uploaded image/ video and select the **Make Public** option.

- Copy the object URL of the images/videos which will be used in the HTML files placed in EC2 server.

| Open | Download | Download as | Make public | Copy path |

Owner
awslabsc0w727930t1586799312

Last modified
Jun 4, 2020 1:50:07 AM GMT+1000

Etag
25cddb8071c22bcc7d67f91f4140f04f

Storage class
Standard

Server-side encryption
None

Size
89.0 KB

Key
Wholesale-Products.webp

Object URL
https://imagewholesaleproducts.s3.amazonaws.com/Wholesale-Products.webp

Steps to create the EC2 instance

- Open Amazon EC2 console and choose launch instance.
- Select a **Amazon Linux 2 AMI (HVM), SSD Volume Type [64 -bit(86)]** from the list of AMI.
- Make sure Instance type is t2.micro (free tier eligible) and click on **Next: Configure Instance Details.**

- Under Advance details, fill the data in **user data**.
  User data :   #!/bin/bash

                 yum install httpd -y

                 service httpd start

- Choose Next: **Add Storage.**
- Choose Next: **Add Tags.** Name the Instance and choose **Next: Configure Security Group**.

- In **Step 6**: Configure Security Group, set **Assign a security group** to Create a new security group. Add the necessary security groups. In our case we need two type of security groups: SSH and HTTP. use the following settings to add HTTP access:
  - **Type:** HTTP
  - **Protocol:** TCP
  - **Port Range:** 80
  - **Source:** Custom 0.0.0.0/0, ::/0

- Choose Review and Launch.
- Choose Launch.
- Select create a new key pair and name the key pair and download it.
- Choose Launch Instances and view the Instance.

Steps to connect the EC2 instance and place the html files inside the html folder through command prompt.

- Select the instance and choose **connect** option.
- To access the instance, run the given linux commands in command prompt.
- Before executing the command, connect to the **titan** server through **ssh** command.
    - ssh s1234***@titan.csit.rmit.edu.au

s3822789@csitprdap01:~

```
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\vigne>ssh s3822789@titan.csit.rmit.edu.au
s3822789@titan.csit.rmit.edu.au's password:
Last login: Fri Jun  5 02:52:55 2020 from 14.137.208.109

                *** PLEASE READ CAREFULLY ***
        ***** This service is for authorised users only *****
```

- Once the EC2 server is connected, copy the public DNS mentioned in the instance page and run it in the browser.
- A static Apache webpage will be displayed. For making it into a dynamic webpage, we need html files to be placed inside the html folder of the ec2 server.
- I have created five html files and attached the html files for reference. Commands to place the files:
   - Cd /var/www/html/
    - Sudo vi index.html
      //paste the html script and close the file. Repeat for the rest four files.
    - Sudo vi contact.html
    - Sudo vi about.html
    - Sudo vi map.html
    - Sudo vi image.html
    - Sudo service httpd restart

```
[s3822789@csitprdap01 ~]$ ssh -i "keypair.pem" ec2-user@ec2-18-204-13-192.compute-1.amazonaws.com
Last login: Thu Jun  4 17:20:10 2020 from coreteaching01.csit.rmit.edu.au

     __|  __|_  )
     _|  (     /   Amazon Linux 2 AMI
    ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-59-142 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-59-142 html]$ sudo vi index.html
[ec2-user@ip-172-31-59-142 html]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-59-142 html]$
```

- Now refresh the static webpage, dynamic website will be created and able to route between the webpages.
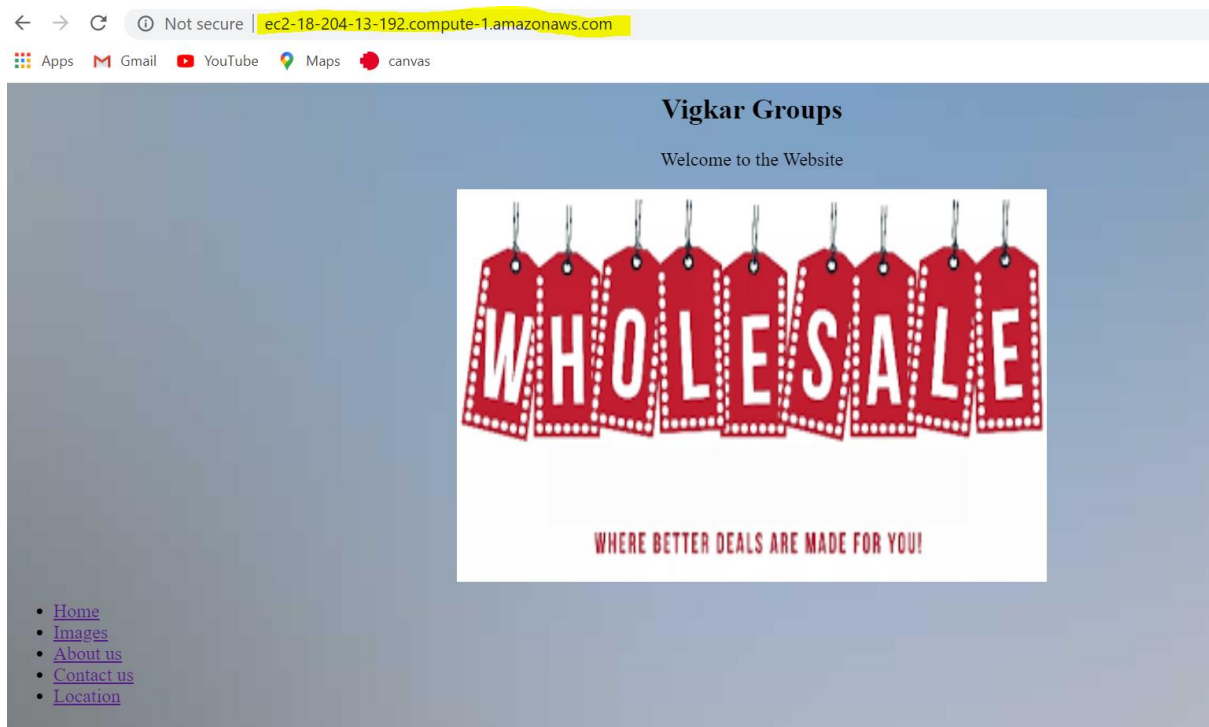
Steps to create an elastic load balancer for the EC2 instance

- Before creating load balancer, launch one more instance by following the steps mentioned above.
- On the EC2 console navigation pane, under **LOAD BALANCING**, choose Load Balancers.
- Choose **Create Load Balancer**.
- In our case, we are creating Application Load Balancer, choose Create.
- Type the name and under availability zones, select all the availability zones and choose **Next: Configure Security settings**.

- choose **Next: Configure Security Groups**.
- Either you can select the existing security groups or create a new security group as we created for EC2 instance.
- Choose **Next: Configure Routing.**
- Type the name for the target group and choose **Next: Register targets.**
- Select the instances and click **add to registered**. Here we can select the number of instances as per our requirement. In our case we registered two instances.

	Group details	Targets	Monitoring	Tags

Registered targets (2)

	Instance ID	Name	Port	Zone	Status	Status details
☐	i-0e59c3bc25c4068d4	server1	80	us-east-1e	⊘ healthy	
☐	i-0fd5edebde654cbae	server 2	80	us-east-1e	⊘ healthy	

- Choose **Next: Review.**
- On the Review page, choose **Create**.
- Once it created, choose close.
- Post to the creation open the Load balancers under the LOAD BALANCING.
- Select the load balancer and under description, wait until the state goes to **active** from **provisioning**.
- Once the state is active, copy the DNS name and open it in the browser.

Load balancer: Loadbalancer

Description	Listeners	Monitoring	Integrated services	Tags

Basic Configuration

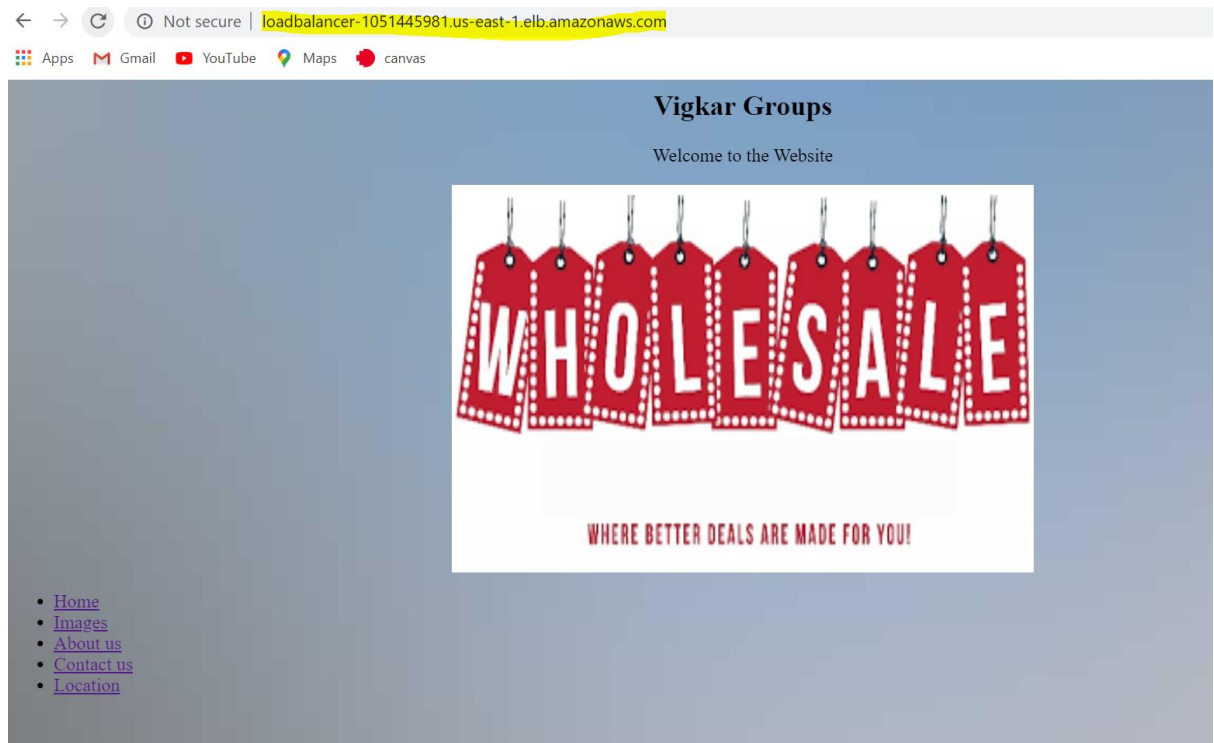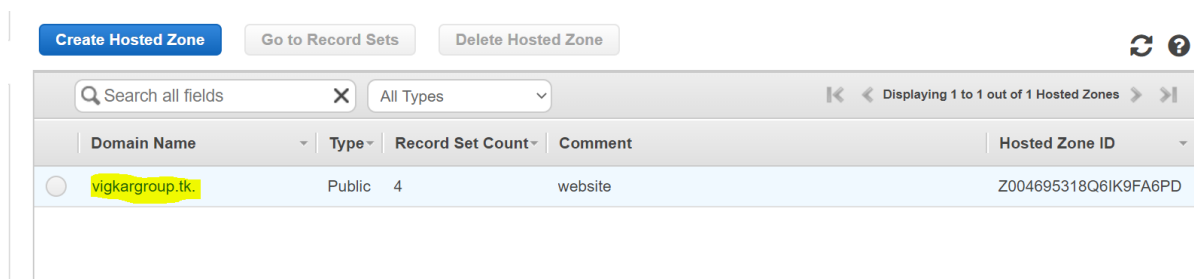|	
Name	Loadbalancer
ARN	arn:aws:elasticloadbalancing:us-east-1:570027386944:loadbalancer/app/Loadbalancer/f8a40cacfc64db15
DNS name	Loadbalancer-1051445981.us-east-1.elb.amazonaws.com
	(A Record)
State	active
Type	application

- The web content which we created using EC2 instance will be accessed from the DNS name of the load balancer.

Steps to configure the Domain name through the Route 53 service

- Select Route 53 service from Amazon console.
- Choose **Create hosted zone** consecutively for three times.
- In the Domain name field, type the registered domain and choose **Create**.

- From the created hosted zone copy all values for the type **NS.**

- Mention the name server values for the domain where we registered the domain name. Mentioning the name server for the domain will be explained in the presentation video please check it for more reference.
- Choose **Create Record Set.**
- In the name field type "**www**" and in the value field mention the public Ip address of the Load balancer and choose **Create**.

- For finding the public IP address of the Load balancer, open EC2 and in navigation pane under LOAD BALANCING, open **load balancer.** Copy the name of the load balancer that we have created. In navigation pane, under NETWORK & SECURITY choose Network Interfaces. In network interfaces, past the name in search bar and filter it. From the filtered results select a network interface and in details tab we can find the public IP.

- Back to Route 53 create hosted zone, Choose **Create Record Set.**
- Select yes for Alias and in the Alias target field, paste the name of the previously created record set. The name will start with www.domainname. and choose **Create.**

- Now open the website with the registered domain name in a browser.
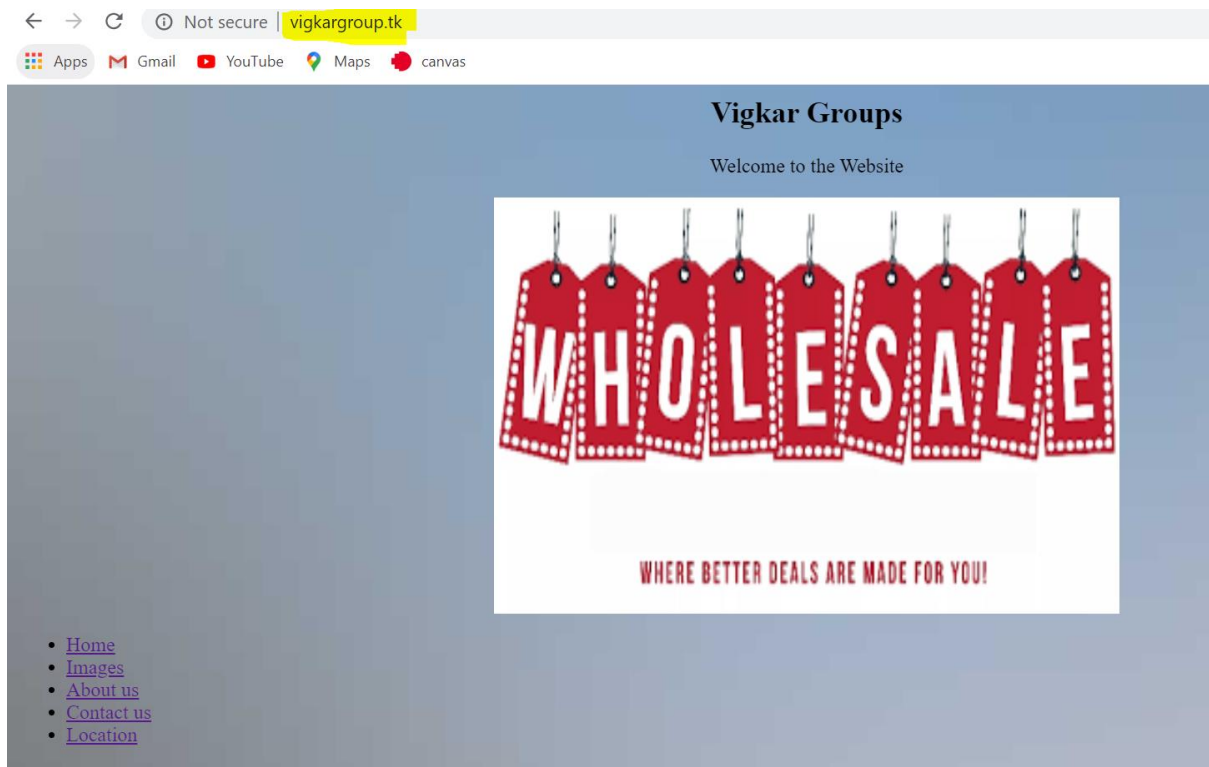
Implementation:

- First the S3 bucket is created and the necessary files are uploaded into the bucket. The image and video source of the website is used from S3 bucket.
- The object URL of the placed content in the S3 bucket is used in the HTML files as the image/ video source.
- We are using S3 bucket to store the content of the website and accessed by the EC2 instance.
- Next two EC2 instance is created and installed httpd service to make the instance to act as webserver.
- Once it is converted to webserver, HTML files are placed inside the HTML folders of the instance. The file placing can be done through the command prompt. The HTML source files are attached for the reference.
- Now we can access the website using the DNS of the instance.

- To maintain the network traffic and application load, Elastic load balancer is created, and two instances is registered to the load balancer.
- When more users trying to access the website, load balancer will split the user and assign different server based on the load.
- Once the state of the load balance become active, we can check with the DNS name of the load balancer to access the website.
- As we need a proper domain name instead of randomly generated Domain name for our website, we need to purchase and register the domain name.
- Then by creating a hosted zone in Route 53 which will help in map the registered domain name to the DNS of the load balancer.
- Whenever we search with the registered domain in the browser it will automatically map and retrieve the data from DNS of load balancer.
- By these steps we can create a dynamic website and access the website through http://www.vigkargroup.tk/

Technical Flow:
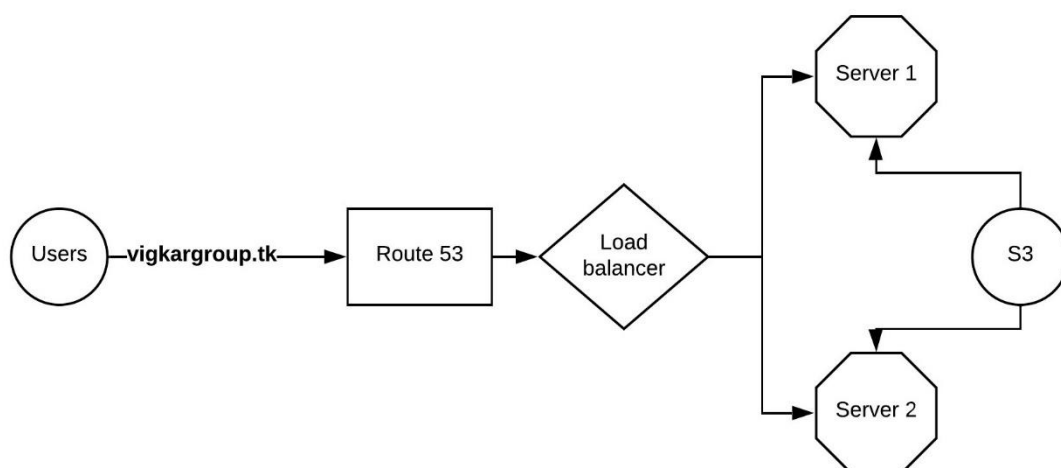
- When the Users try to access the vigkargroup.tk or www.vigkargroup.tk, the route 53 service will map the vigkargroup.tk DNS name to the DNS name of the Load balancer.
- The Application Load balance has two registered instances, once it gets a request through Route 53, it will check for the availability zone and available instance and send the request to the available EC2 instance.
- The EC2 server will process the request accordingly and display the content of the website.
- The images and videos used in the website is accessed directly from the S3 bucket.

Cost Estimates:

Services	Estimates (Monthly)
Route 53 Hosted Zone	0.50 USD
EC2 Instance	12.11 USD
Elastic Load Balancer	16.43 USD
S3 Bucket	0.07 USD

The Non-functional requirements are checked for the built Infrastructure. The services used to build the prototype is much cheaper compared to the other services offered by the Amazon web services. The main conflicting NFR's performance versus cost is managed properly as we used the load balancer, server's performance will not be impacted as well as it is cost effective. The failure can be easily recovered.

Error Handling:

The Major Error occurred while building the website.

1. If we try to access the website, the site is not accessible.
   For this error we must restart the httpd service. Connect to the titan server and connect to the EC2 instance through command prompt and run the below command.
   - ➢ sudo service httpd restart

2. If we are trying to create a new instance through the snapshot of first instance i.e., through AMI image. Then the new sever will have all the properties of the old server but old server will not act as a webserver. Again, we must restart the httpd service of old server.

Note:

The reason for choosing the AWS services, error handling, functionality of the services, Scalability and its reliability are explained and show in the video presentation.