



MULTIMEDIA UNIVERSITY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY

MULTIMEDIA UNIVERSITY

MALAYSIA

TCC3141 CLOUD COMPUTING

TRIMESTER 1, 2022/202

PROJECT REPORT

No	Student Name	Student ID	Programme
1	GRAYSON GOH JIN YI (GROUP LEADER)	1191101340	BIT ST
2	SIAH KAH CHUAN	1191100577	BIT ST
3	FOO HAW LIANG	1191101497	BIT ST
4	ALVIN KUA CHEE SHERN	1181103380	BIT ST
5	GOH JIN JET	1191101375	BIT ST

Table of Content

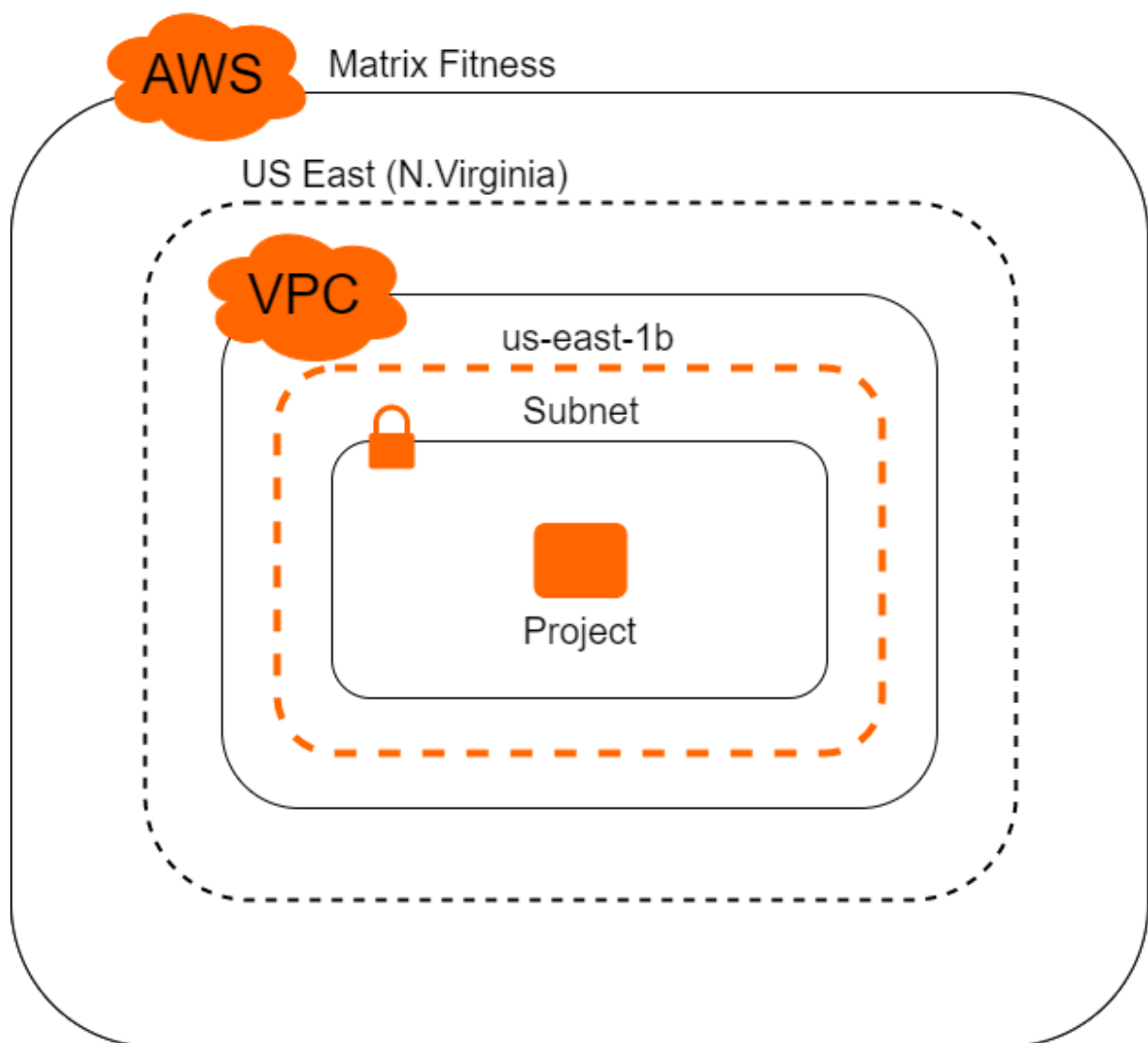
Table of Content	2
Phase 1	3
Introduction of Website	3
Layout Diagram of the networks	3
Phase 1 Implementation steps	4
Phase 2	12
Layout diagram of the revised network	12
Phase 2 Implementation steps	12
Phase 3	16
Layout diagram of the revised network	16
Phase 3 Implementation steps	17
Phase 4	19
Layout diagram of the revised network	19
Phase 4 Implementation steps	20

Phase 1

Introduction of Website

The website we implemented is mainly focused for the admin to do management on the products. The products are all related to fitness equipment. The company name that we applied to is Matrix Fitness. Admin will be required to insert the name and password in order to access the system. The website is designed to be ease of use, user-friendly and simple for admins to proceed any interaction and manipulation. The IP of the website is **projectlb-1004533238.us-east-1.elb.amazonaws.com**

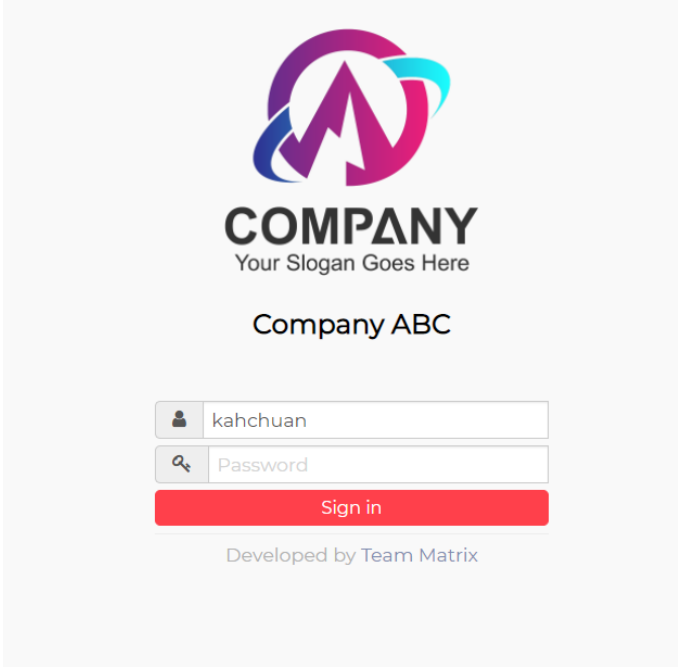
Layout Diagram of the networks



Phase 1 Implementation steps

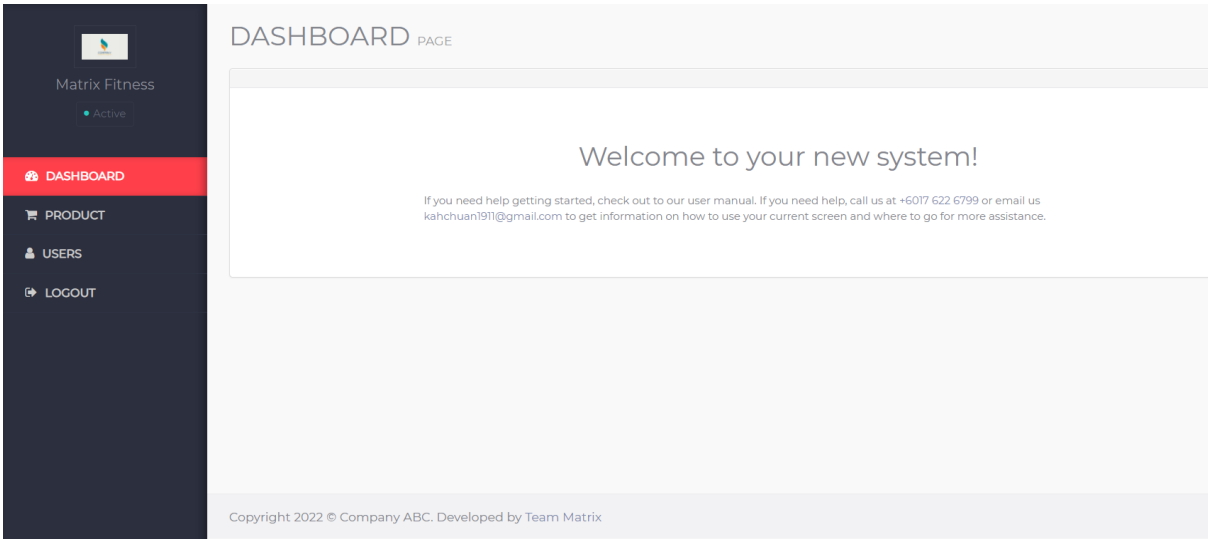
First, prepare the e-commerce website template as shown below:

Login Page




The login page features a central logo consisting of a stylized 'M' in blue and purple, with a teal swoosh. Below the logo, the text 'COMPANY' is displayed in a large, bold, black font, followed by the placeholder 'Your Slogan Goes Here' in a smaller font. Underneath, 'Company ABC' is written in a bold, black font. The login form includes a username field with a person icon and the text 'kahchuan', a password field with a magnifying glass icon and the text 'Password', and a red 'Sign in' button. At the bottom, it says 'Developed by Team Matrix'.

Home Page



The home page is a dashboard layout. On the left is a dark blue sidebar with a 'Matrix Fitness' logo and 'Active' status. The sidebar menu includes 'DASHBOARD' (highlighted in red), 'PRODUCT', 'USERS', and 'LOGOUT'. The main content area has a header 'DASHBOARD PAGE' and a large white box with the text 'Welcome to your new system!' and a paragraph of placeholder text. The footer contains the copyright notice 'Copyright 2022 © Company ABC. Developed by Team Matrix'.

Product Page


Matrix Fitness
Active

[DASHBOARD](#)


[PRODUCT](#)

[USERS](#)

[LOGOUT](#)

PRODUCT LIST

Add New

No.	Product Image	Product Name	Product Amount	Product Unit Price	Product Selling Price	Product Description	Action
1		siah	10005	20.00	30.00	def	Edit Delete

Copyright 2022 © Company ABC. Developed by Team Matrix

Next, launch the EC2 with the following setting:

- Name the instance as project
- Choose the Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
- For the instance type, choose the t2.micro
- For the Firewall (security group), choose create security group and with the following setting:

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&:{}!\$*

Description - *required* [Info](#)

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)

Protocol [Info](#)
TCP

Port range [Info](#)
22

Source type [Info](#)

Source [Info](#)

X

Add CIDR, prefix list or security

Description - *optional* [Info](#)

▼ Security group rule 2 (TCP, 443, 0.0.0.0/0)
Remove

Type
Info

HTTPS

Protocol
Info

TCP

Port range
Info

443

Source type
Info

Anywhere

Source
Info

Add CIDR, prefix list or security
0.0.0.0/0

Description - optional
Info

e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 80, 0.0.0.0/0)
Remove

Type
Info

HTTP

Protocol
Info

TCP

Port range
Info

80

Source type
Info

Anywhere

Source
Info

Add CIDR, prefix list or security
0.0.0.0/0

Description - optional
Info

e.g. SSH for admin desktop

Add security group rule

- For the key pair, choose create new key pair, name with project and key pair type select RSA, the private key file format select ppk and save the **project.ppk**.
- After all the settings are done, press Launch Instance. Below show the launched EC2 instances:

Instances (1/1)
Info
Find instance by attribute or tag (case-sensitive)
1

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
<input checked="" type="checkbox"/>	project	i-04b6a486c46a288a1	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-20

Instance: i-04b6a486c46a288a1 (project)

Details
Security
Networking
Storage
Status checks
Monitoring
Tags

▼ Instance summary
Info

Instance ID
i-04b6a486c46a288a1 (project)

IPv6 address
-

Hostname type
IP name: ip-172-31-18-52.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
18.207.115.236 [Public IP]

IAM Role
-

Public IPv4 address
18.207.115.236 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-18-52.ec2.internal

Instance type
t2.micro

VPC ID
vpc-03302ea6303db39a5

Subnet ID
subnet-0515f7e5a873ac3a0

Private IPv4 addresses
172.31.18.52

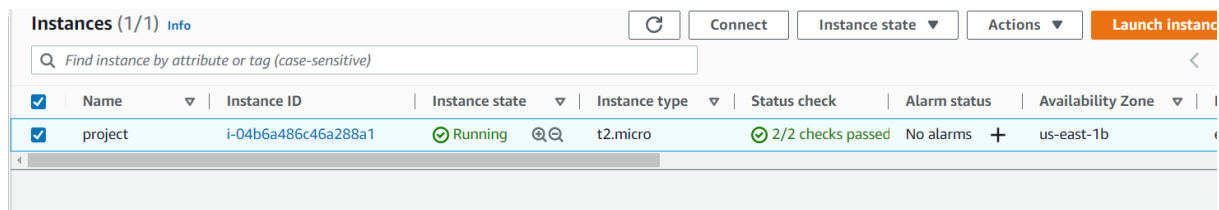
Public IPv4 DNS
ec2-18-207-115-236.compute-1.amazonaws.com | open address

Elastic IP addresses
-

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name
-

After launching the EC2 instances, choose the Connect button as shown below to move to the CLI.



After that, install the packages as shown as below:

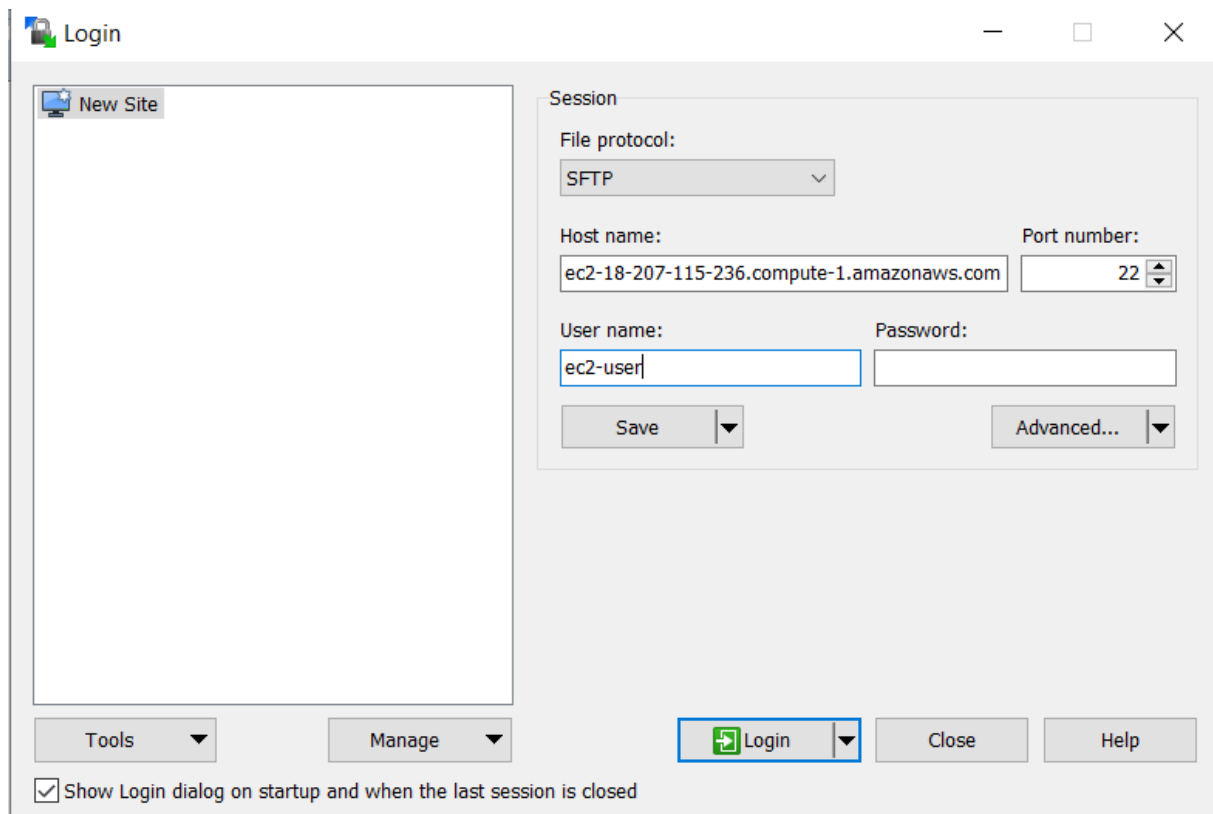
```
[ec2-user@ip-172-31-18-52 ~]$ sudo yum list installed httpd mariadb-server php-mysqld
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
httpd.x86_64                                     2.4.54-1.amzn2
mariadb-server.x86_64                           3:10.2.38-1.amzn2.0.1
php-mysqld.x86_64                               7.2.34-1.amzn2
[ec2-user@ip-172-31-18-52 ~]$
```

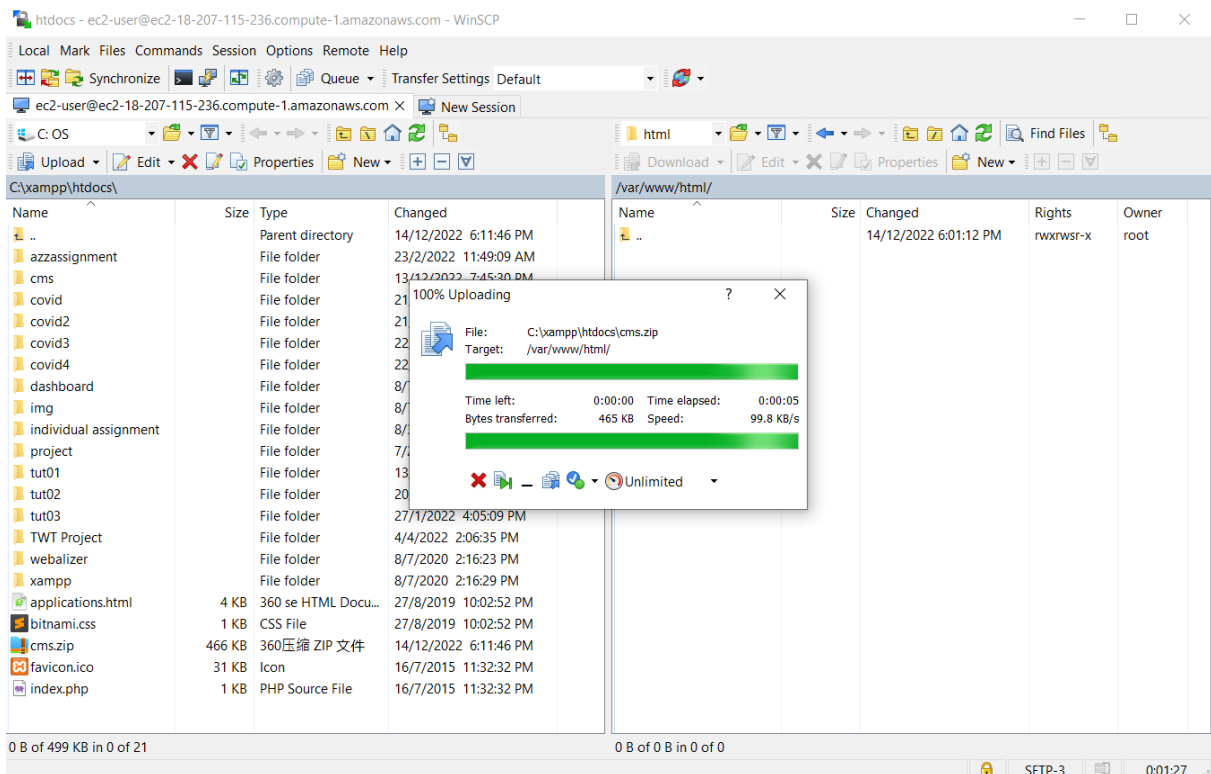
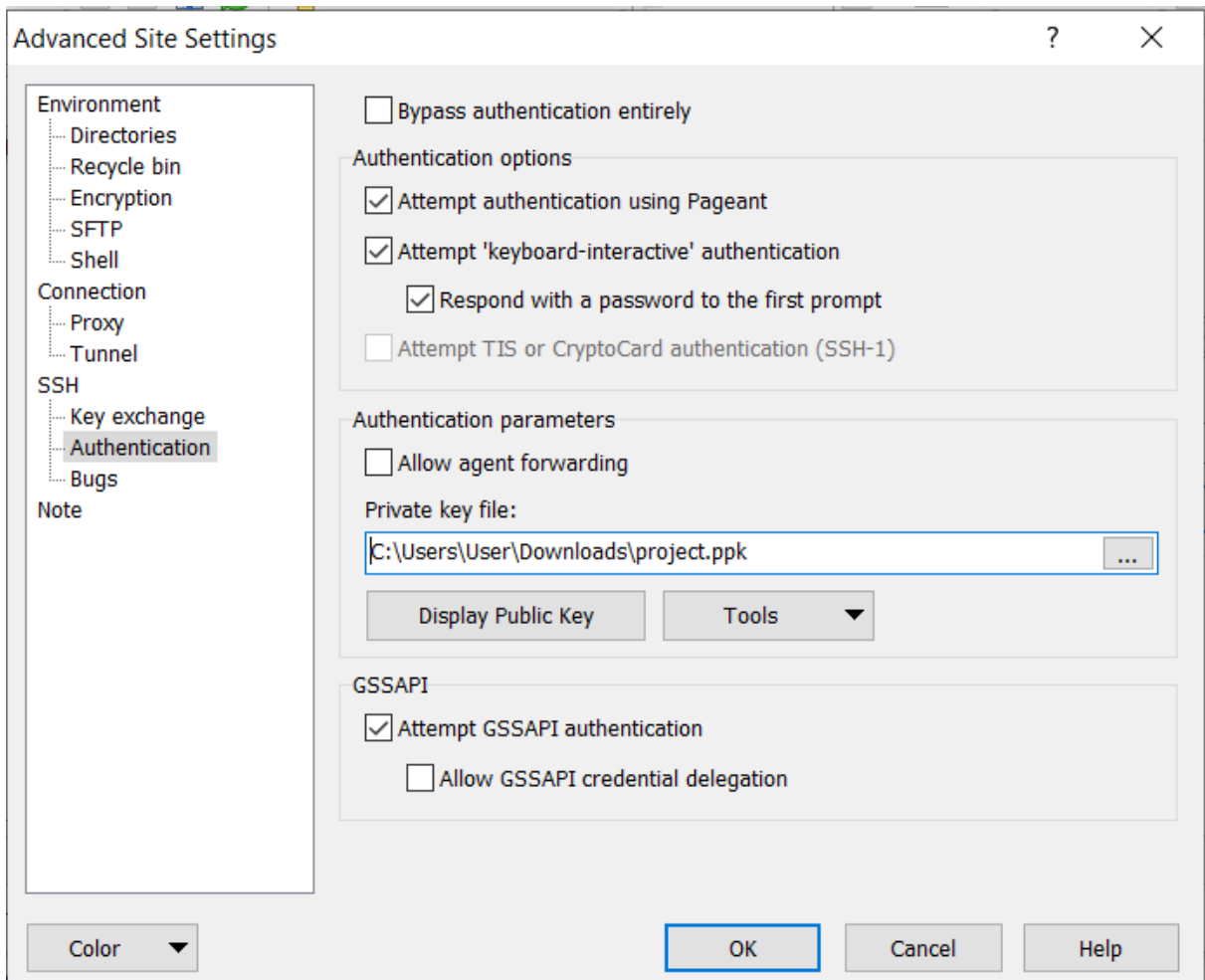
After installed all the above packages, enabled it by using the following command:

- `sudo systemctl start httpd`
- `sudo systemctl enable httpd`

After opening the httpd, we need to login into WinSCP using the private key **project.ppk** we have created just now. Then we need to transfer the web template to the EC2 instances by using WinSCP

The following steps have been shown below:





/var/www/html/				
Name	Size	Changed	Rights	Owner
..		14/12/2022 6:01:12 PM	rw-rw-r--x	ec2-user
cms		14/12/2022 3:45:32 AM	rw-rw-r--x	ec2-user

After transferring the web template to the EC2 instances, we move back to CLI to configure the database. The steps have been shown below:

```
[ec2-user@ip-172-31-18-52 ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 22
Server version: 10.2.38-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

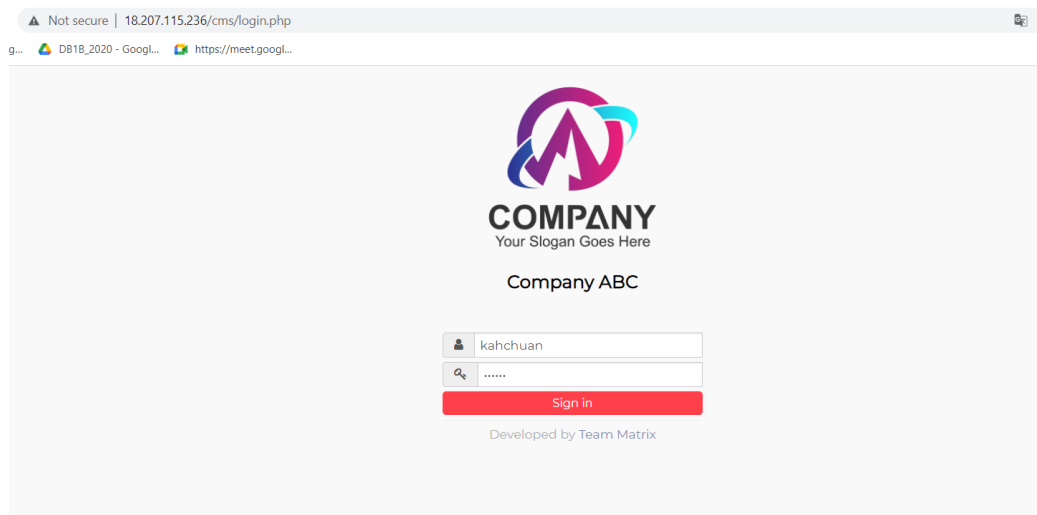
MariaDB [(none)]> CREATE DATABASE system_cms;
Query OK, 1 row affected (0.00 sec)
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-52 ~]$ cd /var/www/html/cms/Connect
[ec2-user@ip-172-31-18-52 Connect]$ sudo mysql -u root -p system_cms < system_user.sql
Enter password:
[ec2-user@ip-172-31-18-52 Connect]$ sudo mysql -u root -p system_cms < system_product.sql
Enter password:
[ec2-user@ip-172-31-18-52 Connect]$
```

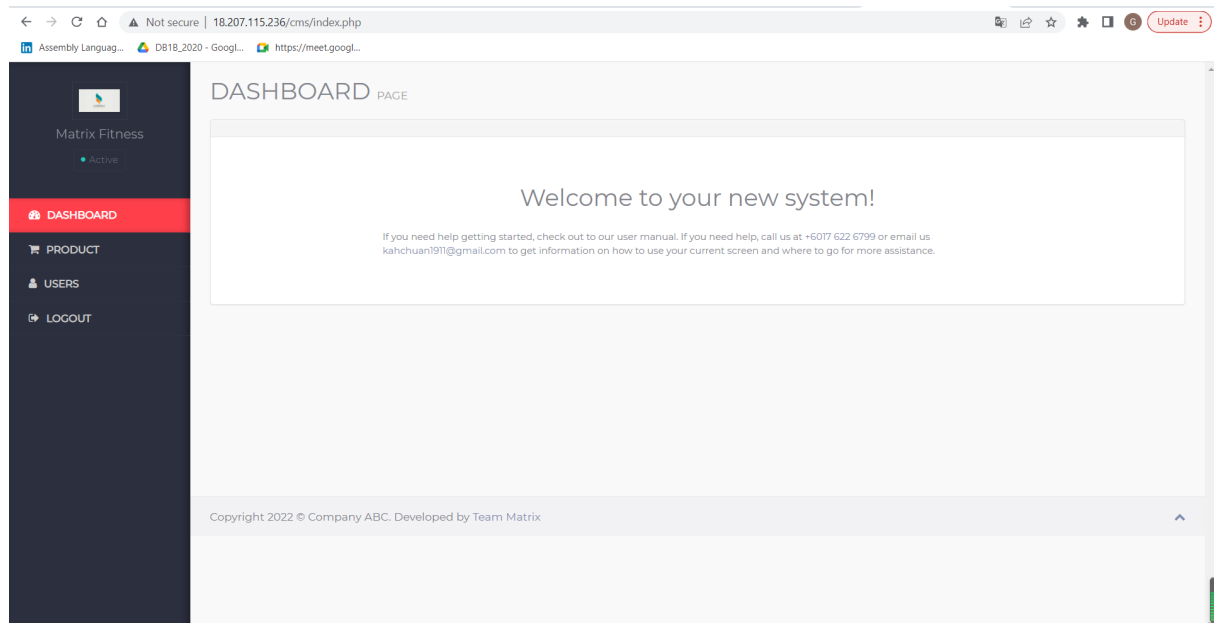
After all the set up the website can be fully functional.

IP of the website: <http://18.207.115.236/cms/login.php>

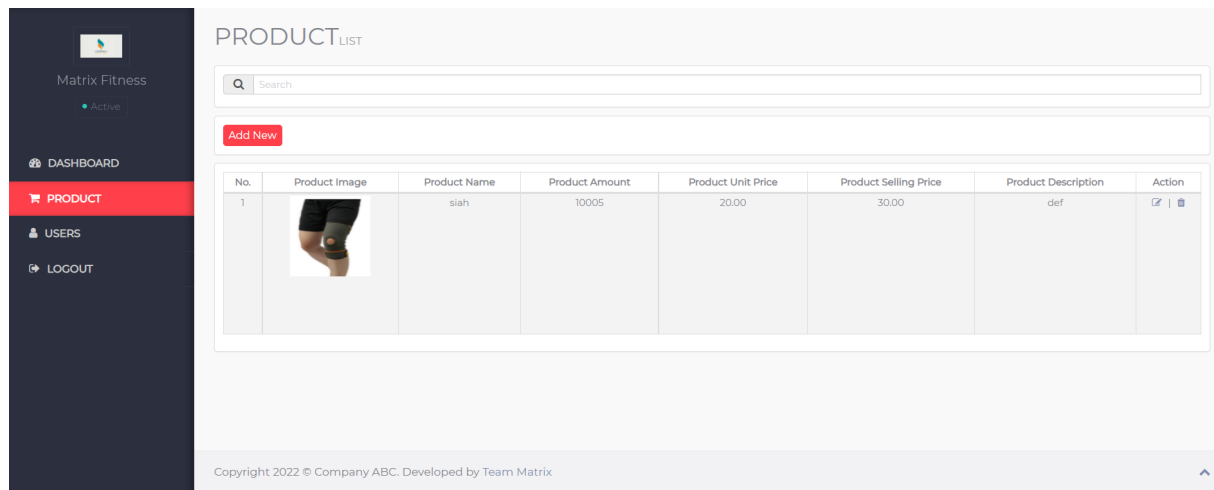
Login Page



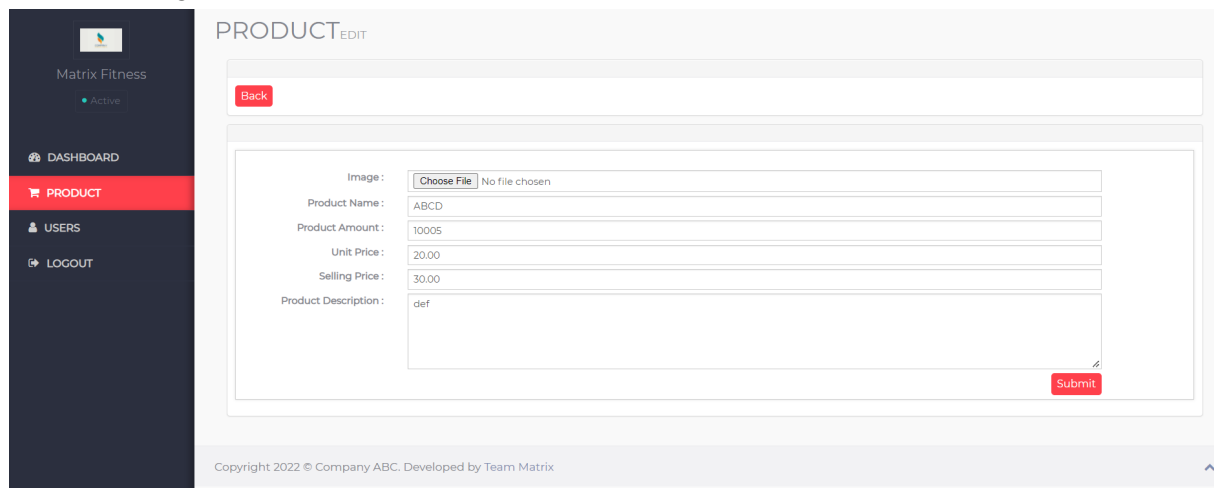
Home Page




Product Page



Product Editing



User Admin Page



Matrix Fitness

● Active

DASHBOARD

PRODUCT

USERS

LOGOUT

USERS

LIST

Search

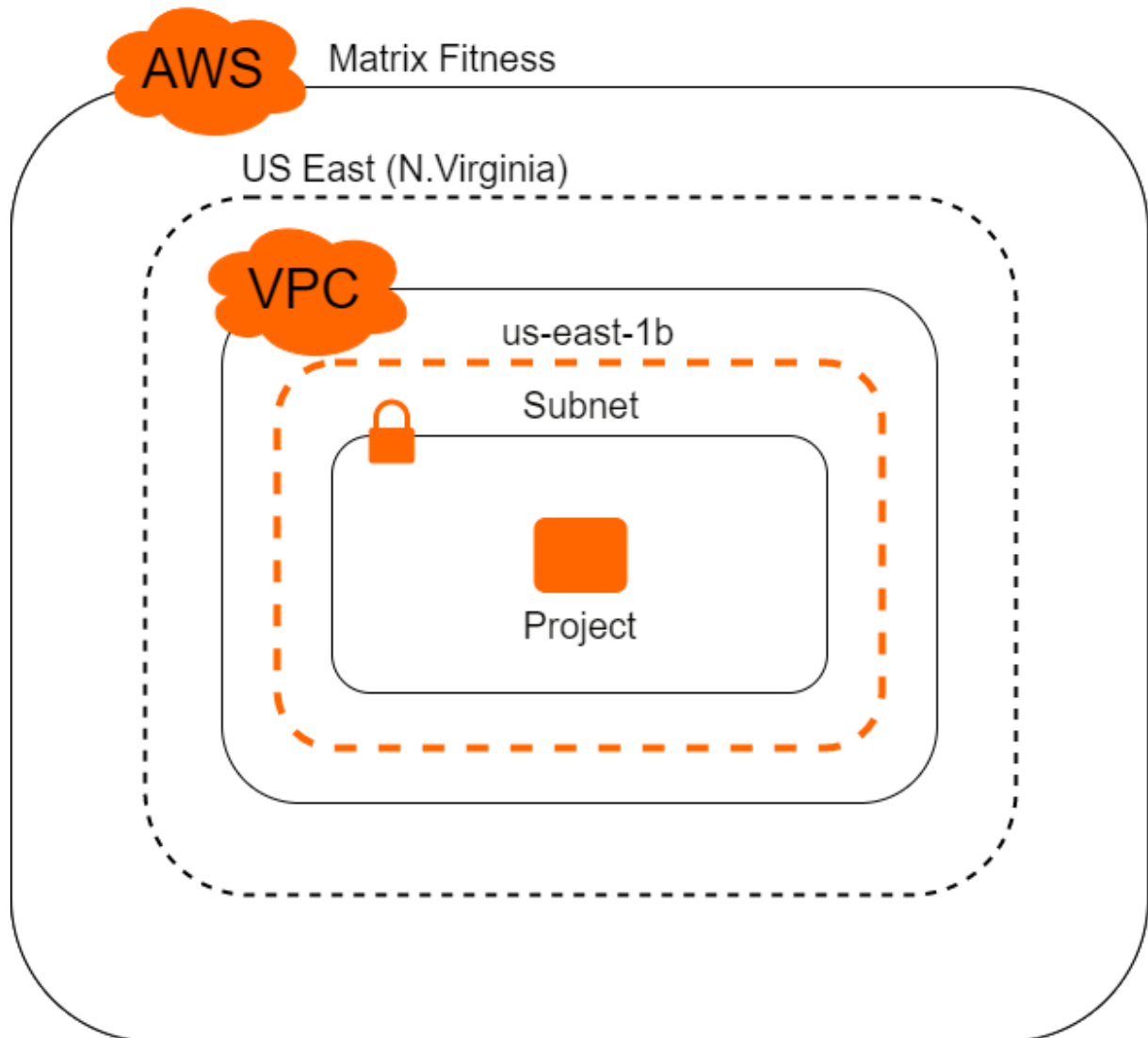
Add New

No.	Username	Full Name	Contact No.	Email Address	Action
1	kahchuan	Siah Kah Chuan	+6017 622 6799	kahchuan1911@gmail.com	<div><div></div><div></div></div>

Copyright 2022 © Company ABC. Developed by Team Matrix

Phase 2

Layout diagram of the revised network



Phase 2 Implementation steps

In phase 2, we need to separate the database from the web application EC2 instance we have created in phase 1. The first step is to open a new RDS instance. With the following set up

- Database creation method choose **standard create**
- Engine option choose **MySQL**
- Template choose **free tier**
- Name the RDS instances as **system_cms**
- And for the connectivity, choose **Connect to an EC2 compute resource and select the EC2 instance we have create before**
- And press the **Create Database**

Connectivity [Info](#)

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☐ Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

☒ Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 Instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-04b6a486c46a288a1

project

Some VPC settings can't be changed when a compute resource is added
Adding an EC2 compute resource automatically selects the VPC, DB subnet group, and public access settings for this database. To allow the EC2 instance to access the database, a VPC security group rds-ec2-X is added to the database and another called ec2-rds-X to the EC2 instance. You can remove the new security group for the database only by removing the compute resource.

Virtual private cloud (VPC) [Info](#)

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-03302ea6303db39a5)

Below shows the created RDS,

☑ Successfully set up a connection between systemcms and EC2 instance i-04b6a486c46a288a1 [🔗](#)

RDS > Databases > systemcms

systemcms

Modify Actions

Summary

DB identifier systemcms	CPU <div>2.00%</div>	Status <div>Available</div>	Class db.t3.micro
Role Instance	Current activity <div>0 Connections</div>	Engine MySQL Community	Region & AZ us-east-1b

Connectivity & security				
Endpoint & port Endpoint systemcms.chulshacmix3.us-east-1.rds.amazonaws.com Port 3306		Networking Availability Zone us-east-1b VPC vpc-03302ea6303db39a5 Subnet group rds-ec2-db-subnet-group-1 Subnets subnet-04c48ba639d441600 subnet-03ec3e884228175cf subnet-016a790b2bfa7fe7d subnet-0ff1aed1ddebfb1b5 subnet-0ba07800f50b6c50f subnet-043af157ed29a0344 Network type IPv4		
		Security VPC security groups launch-wizard-1 (sg-06aca57da016f68d7) Active rds-ec2-1 (sg-03156a444c9dcd9d8) Active Publicly accessible No Certificate authority rds-ca-2019 Certificate authority date August 23, 2024, 01:08 (UTC+08:00)		

Connected compute resources (1) Info				
Connections to compute resources that were created automatically by RDS are shown here. Connections to compute resources that were created manually aren't shown.				
<input type="text" value="Filter by compute resources"/>				
Resource identifier 🔗	Resource type	Availability zone	RDS security group 🔗	Compute resource security group 🔗
i-04b6a486c46a288a1	EC2 Instance	us-east-1b	rds-ec2-1	ec2-rds-1

After creating the RDS database, we move to the EC2 CLI to log in to the RDS database and import the MYSQL dump file.

```
[ec2-user@ip-172-31-18-52 ~]$ mysql -h systemcms.chulshacmix3.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.


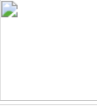
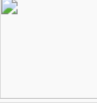
MySQL [(none)]> CREATE DATABASE system_cms
-> ;
Query OK, 1 row affected (0.01 sec)
```

```
[ec2-user@ip-172-31-18-52 Connect]$ mysql -h systemcms.chulshacmix3.us-east-1.rds.amazonaws.com -u admin -p system_cms < system_product.sql
Enter password:
[ec2-user@ip-172-31-18-52 Connect]$ mysql -h systemcms.chulshacmix3.us-east-1.rds.amazonaws.com -u admin -p system_cms < system_user.sql
Enter password:
```

And now we have the separate RDS database

(**endpoint:systemcms.chulshacmix3.us-east-1.rds.amazonaws.com**) connected with the EC2 instances . For testing purposes, we try to add a new product to the website and compare it with the RDS database.

We have added a testing product to our website and check whether it updates to our RDS database.

Add New							
No.	Product Image	Product Name	Product Amount	Product Unit Price	Product Selling Price	Product Description	Action
1		siah	10005	20.00	30.00	def	✎ ✕
2		dfdf	123	123.00	555.00	ffff	✎ ✕
3		product testing	11	15.00	20.00	Testing the RDS database	✎ ✕

```
[ec2-user@ip-172-31-18-52 Connect]$ mysql -h systemcms.chulshacmix3.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 50
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> USE system_cms
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

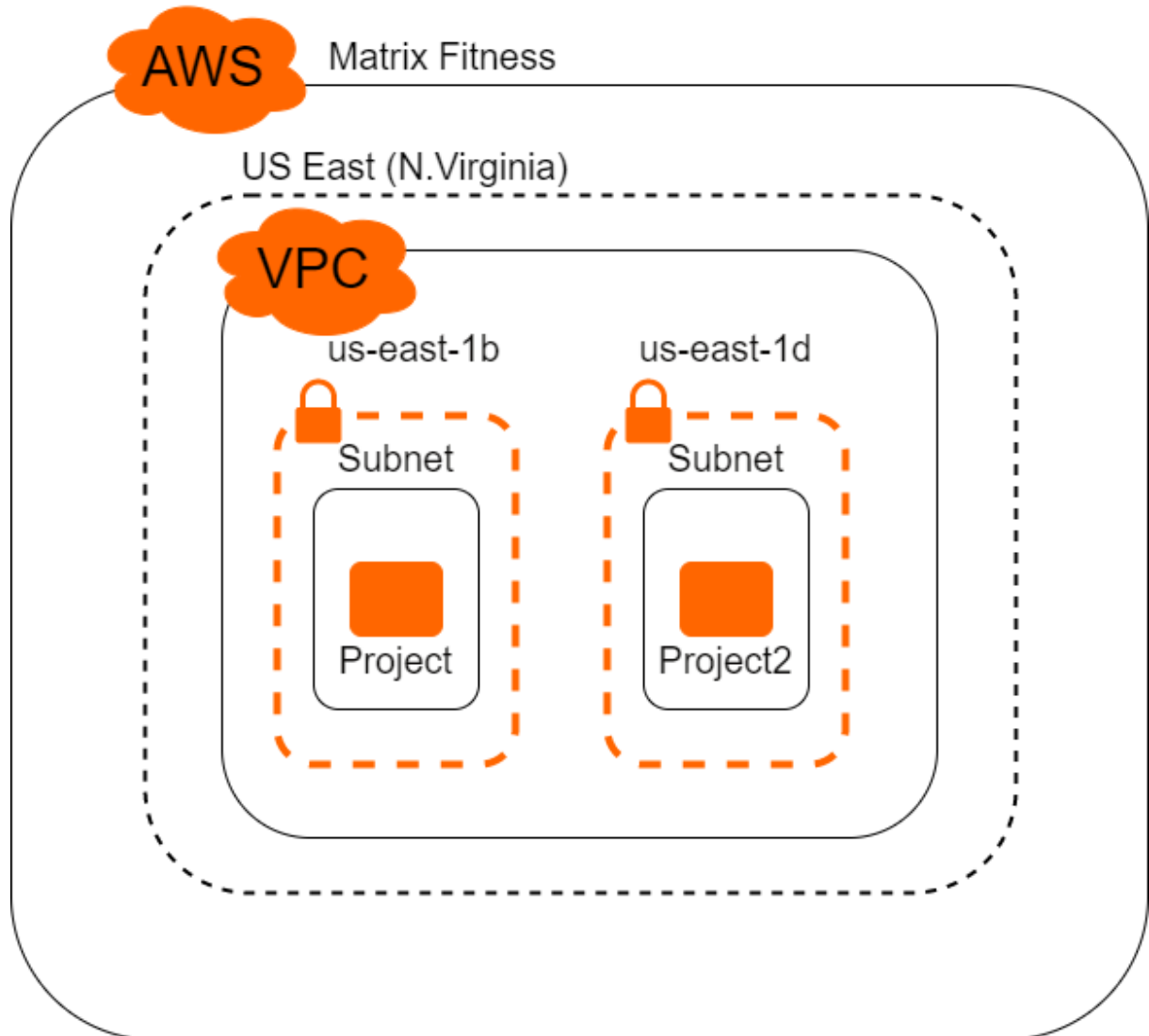
Database changed
```

```
MySQL [system_cms]> SELECT * FROM system_product;
+-----+-----+-----+-----+-----+-----+-----+-----+
| product_id | product_name | product_pic | product_amt | product_unit_price | product_sell_price | product_description | product_tras |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | siah | P2.png | 10005 | 20.00 | 30.00 | def | 1 |
| 2 | siah | P2.png | 10005 | 20.00 | 30.00 | def | 0 |
| 3 | dfdf | NULL | 123 | 123.00 | 555.00 | ffff | 0 |
| 4 | product testing | NULL | 11 | 15.00 | 20.00 | Testing the RDS database | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

With this, we have completed the Phase 2 implementation.

Phase 3

Layout diagram of the revised network



Phase 3 Implementation steps

1) Create a second ec2 instance with different availability zone, and different key pair

- For a quicker creation, can go to your first EC2 instance, click image and template and select launch more like this and then change the key pair and availability zone
- After that, open up the cli under the connect option
- Perform the same setting you used during first EC2 setup
- Go to /var/www/html folder to unzip the cms.zip
- Test your second EC2 instance

Instances (3) Info							
Find instance by attribute or tag (case-sensitive)							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	project	i-04b6a486c46a288a1	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b
<input type="checkbox"/>	project2	i-086ce86f7de494d3b	Terminated	t2.micro	-	No alarms	us-east-1b
<input type="checkbox"/>	project2.0	i-039a5b4616df6fb8c	Running	t2.micro	Initializing	No alarms	us-east-1a

2) Load balancing

a) Create a target group

- Go to "Target Groups" at the sidebar of EC2 dashboard
- Click create target groups
- Keep instance target groups selected
- Create a new target group name
- Keep default setting for the rest of section
- Click next
- Select the two EC2 instance that you have created earlier
- Include as pending below



Target groups (1/1) Info

Actions ▾

Create target group

Q Search or filter target groups

< 1 >

<input checked="" type="checkbox"/>	Name ▾	ARN ▾	Port ▾	Protocol ▾	Target type ▾	Load balancer ▾
<input checked="" type="checkbox"/>	projecttargetgroup	 arn:aws:elasticloadbalancing...	80	HTTP	Instance	 None associated

b) Create a load balancer

- Go to the sidebar and find load balancer and create load balancer
- Choose application load balancer
- Create a new load balancer name
- Keep default setting for scheme, ip address type and VPC
- Choose the availability zone for your two EC2 instance
- Select the security groups as what you applied for your EC2 instance
- Keep default setting for the rest, select the target group created before and create load balancer
- After creation, We can use the Elastic Load Balancer DNS to access the system.
- The DNS name is shown below:

Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter by property or value

search: ProjectLoadBalancer

Clear filters

<1>

	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	ProjectLoadBalancer	ProjectLoadBalancer-1713316753.us-east-1.elb.amazonaws.com	Provisioning	vpc-03302ea6303db39a5	2 Availability Zones	application

Matrix Fitness

Active

DASHBOARD

PRODUCT

USERS

LOGOUT

DASHBOARD PAGE

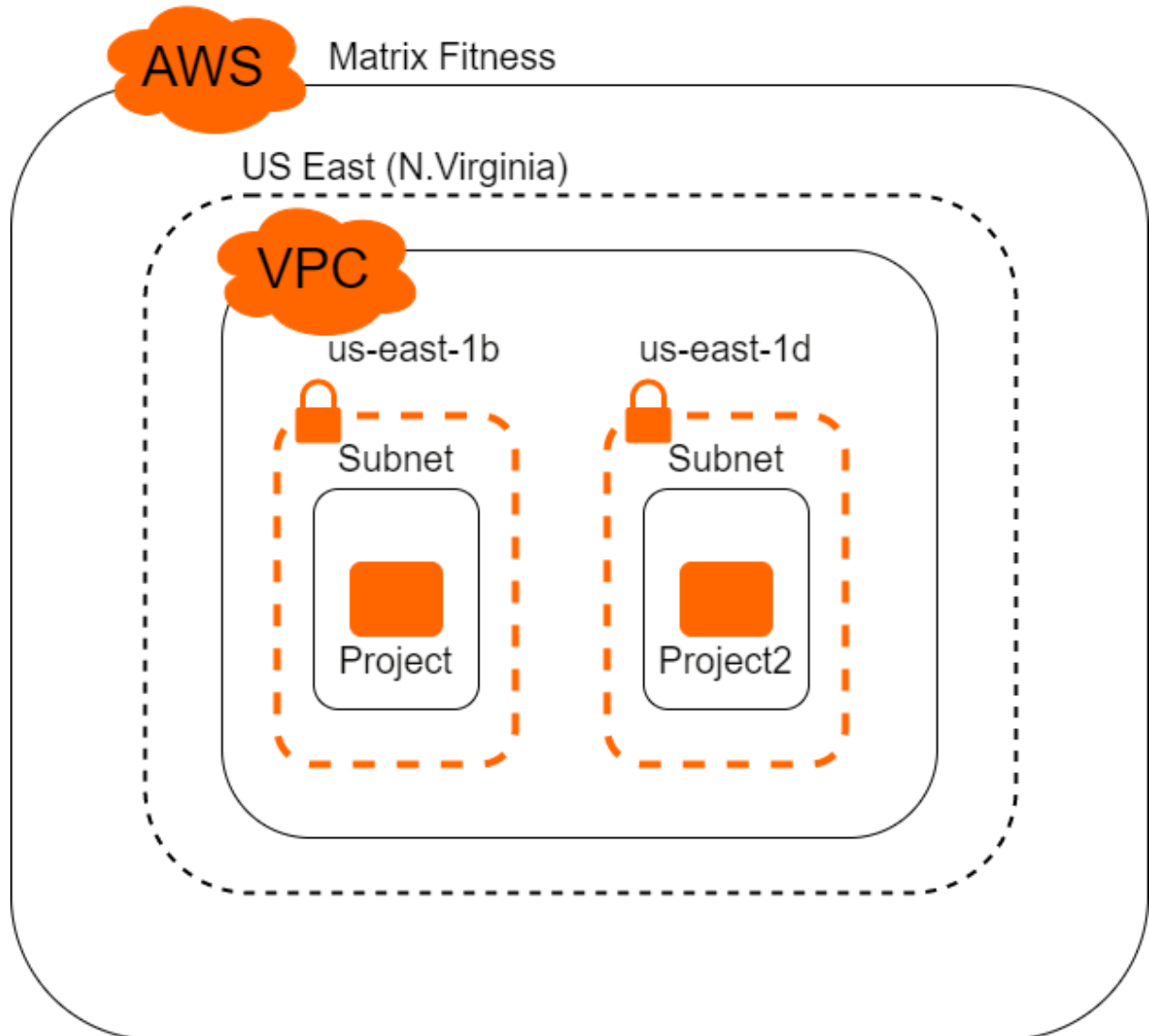
Welcome to your new system!

If you need help getting started, check out to our user manual. If you need help, call us at +6017 622 6799 or email us kahchuan91@gmail.com to get information on how to use your current screen and where to go for more assistance.

Copyright 2023 © Company ABC. Developed by Team Matrix

Phase 4

Layout diagram of the revised network



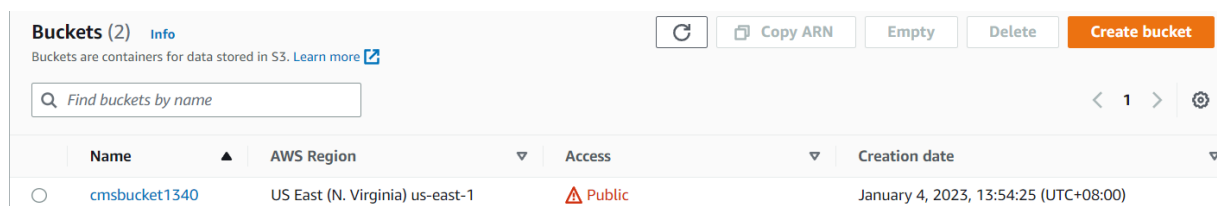
Phase 4 Implementation steps

1) First, we need to create a S3 bucket to store our web template

- Go to S3 bucket and choose create bucket
- Give the bucket a new name
- Keep default settings for ACLs disabled
- Untick the block all public access option
- Keep the default settings for the rest of section
- Press create bucket option
- After the bucket has been created
- Go to the details page and upload file
- Upload the "cms.zip" file into S3 bucket
- After that go to permission tab and scroll down to bucket policy and edit it
- Paste the following command:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::files-str/*" <- (change this to your
      own arn)
    }
  ]
}
```

- After that press save changes



2) After creating the S3 bucket, we need to setup the launch template

- Goto launch templates
- Choose create launch template
- Create a new launch template name
- Give brief description for template version description (optional)
- Tick on auto scaling guidance
- Choose your favourite application and os images (mine is amazon linux 2 x86)
- Choose t2.micro for instance type

- Choose the security groups for the one you applied for your EC2 instance
- Open up advanced detail, go to user data and fill up the command you use in your EC2 and create the launch template
- The following command:

```
#!/bin/bash
yum update -y
amazon-linux-extras install php8.0 mariadb10.5
yum -y install httpd
systemctl enable httpd
systemctl start httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
systemctl restart httpd
wget https://cmsbucket1340.s3.amazonaws.com/cms.zip -P /var/www/html
unzip /var/www/html/cms.zip -d /var/www/html/
echo "<?php header('Location:cms/index.php');?>" > /var/www/html/index.php
```

Launch template ID	Launch template name	Default version	Latest version
lt-0946991bf3c4433f3	project	1	1

3) Lastly, we need to setup the auto scaling groups

- Go to auto scaling groups
- Choose create auto scaling groups
- Create a new auto scaling groups name
- Select the template we created earlier
- Choose VPC and the two availability zone for the two EC2 we created earlier
- Choose attach to an existing load balancer which you created earlier
- Keep default setting for the rest of the page
- Keep desired capacity as 2, minimum capacity as 1, maximum capacity as 4
- Choose target tracking scaling policy select average cpu utilization for give 80 (or any number you like) for target value and create the auto scaling group
- After that go to the EC2 instance list and click on the EC2 instance respectively and select action, go to instance setting and click attach to auto scaling groups
- Later on, go to the list of the auto scaling groups and select the auto scaling groups created just now
- Go to the instance management and select the two EC2 instances that you created and click on action to set scale in protection to avoid the EC2 instances getting deleted.

Auto Scaling groups (1/1) Info

🔄

Edit

Delete

Create an Auto Scaling group

🔍 Search your Auto Scaling groups

<1>⚙️

<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availa...
<input checked="" type="checkbox"/>	projectAutoScaling	project Version Default	2	-	2	1	3	us-east...

projectAutoScaling

Details

Activity

Automatic scaling

Instance management

Monitoring

Instance refresh

Instances (4)

🔄

Actions

🔍 Filter instances

<1>⚙️

<input type="checkbox"/>	Instance ID	Lifecycle	Instanc...	Weichte...	Launch ...	Availabi...	Health s...	Protecte...
<input type="checkbox"/>	i-039a5b4616df6fb8c	InService	t2.micro	-		us-east-1a	Healthy	Scale in
<input type="checkbox"/>	i-04b6a486c46a288a1	InService	t2.micro	-		us-east-1b	Healthy	Scale in
<input type="checkbox"/>	i-08782415cf1f15534	Terminating	t2.micro	-	project Vers	us-east-1a	Healthy	
<input type="checkbox"/>	i-0e77a4b55b1636c34	InService	t2.micro	-	project Vers	us-east-1b	Healthy	

Instances (4) Info

🔄

Connect

Instance state

Actions

Launch instances

🔍 Find instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

<1>⚙️

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	project	i-04b6a486c46a288a1	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-215-166
<input type="checkbox"/>	-	i-0e77a4b55b1636c34	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-163-52-
<input type="checkbox"/>	project2.0	i-039a5b4616df6fb8c	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-92-185-6
<input type="checkbox"/>	-	i-08782415cf1f15534	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-84-209-