



CT097-3-3-CSVC

CLOUD INFRASTRUCTURE AND SERVICES

GROUP ASSIGNMENT

STUDENT NAME	STUDENT ID
SHAIK YASEEN	TP02920

GROUP NUMBER : CT097-3-3-CSVC-L-2

INTAKE CODE : APU4F2505CE

LECTURER NAME : [YOGESWARAN A/L NATHAN](#)

SUBMISSION DATE : 29th August 2025

TABLE OF CONTENTS

LIST OF FIGURES.....	ii
LIST OF TABLES	iii
1.0 INTRODUCTION.....	1
2.0 SOLUTION DIAGRAM ILLUSTRATION	3
3.0 DESIGN DECISIONS	5
3.1 DEPLOYMENT	5
3.1.1 VPC	5
3.1.2 SUBNET	5
3.1.3 INTERNET GATEWAY	6
3.1.4 NAT GATEWAY	7
3.1.5 ROUTE TABLE	7
3.2 SECURITY.....	10
3.2.1 NETWORK ACL.....	10
3.2.2 SECURITY GROUPS	11
3.2.3 PARAMETER STORE.....	13
3.2.4 SSH CONNECTION.....	14
3.3 HIGH AVAILABILITY AND SCALABILITY	16
3.3.1 AMAZON S3.....	16
3.3.2 LAUNCH TEMPLATE	17
3.3.3 LOAD BALANCER	19
3.3.4 AUTO SCALING GROUPS.....	21
3.3.5 AMAZON RDS	23
4.0 FINAL OUTCOME	26
5.0 ESTIMATED PROJECT PRICING	27
6.0 FUTURE ENHANCEMENTS.....	28
7.0 CONCLUSION	29
REFERENCES.....	30
APPENDICES.....	31
1) WORK BREAKDOWN STRUCTURE.....	31

LIST OF FIGURES

Figure 1: Design of Overall Architecture.....	4
Figure 2: Design of Overall Architecture.....	4
Figure 3: VPC.....	5
Figure 4: Subnet.....	6
Figure 5: Internet Gateway.....	7
Figure 6: NAT Gateway.....	7
Figure 7: Public Route Table	8
Figure 8: Private Route Table	8
Figure 9: Network ACL (Inbound rules).....	10
Figure 10: Network ACL (Outbound rules).....	11
Figure 11: Network ACL (Subnet associations)	11
Figure 12: Security Groups (Bastion-secg).....	12
Figure 13: Security Groups (Appelb-secg)	13
Figure 14: Security Groups (Apptier-secg).....	13
Figure 15: Security Groups (Db-secg)	13
Figure 16: Parameter Store.....	14
Figure 17: SSH Connection 1	15
Figure 18: SSH Connection 2	15
Figure 19: S3 Bucket.....	16
Figure 20: Public Launch Template	18
Figure 21: Private Launch Template.....	19
Figure 22: Target Group.....	20
Figure 23: Load Balancer.....	21
Figure 24: Public Auto Scaling Groups	23
Figure 25: Private Auto Scaling Groups	23
Figure 26: RDS Private Subnet	24
Figure 27: RDS Configurations.....	25
Figure 28: Final Outcome	26
Figure 29: Estimated Project Pricing	27

LIST OF TABLES

Table 1: Subnet	6
Table 2: Route Table.....	9
Table 3: Security Groups	12
Table 4: Parameter Store.....	14
Table 5: Public Launch Template	17
Table 6: Private Launch Template	18
Table 7: Target Group.....	20
Table 8: Load Balancer	20
Table 9: Auto Scaling Groups.....	22
Table 10: RDS Private Subnet	24
Table 11: RDS Configurations	24

1.0 INTRODUCTION

Amazon Web Services (AWS) is a dynamic cloud computing platform that keeps evolving and is loved by millions of users and businesses around the world (Amazon, 2022i). AWS provides a wide range of cloud computing services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), along with hundreds of products and tools designed to help customers create the perfect cloud environment (Gillis, 2020).

IaaS is a go-to service for network architects who need to access virtualized computing resources like virtual storage and machines over the internet. PaaS is favored by app developers for its tools that facilitate the development and deployment of applications. Meanwhile, SaaS is popular among businesses and their customers, as it allows companies to host their software or websites, making them easily accessible to users (GeeksforGeeks, 2022).

Shirley Rodriguez, a researcher at the Example Social Research Organization, created a website to share her collected data with fellow researchers. She stored the data in a MySQL database, which could be accessed through the PHP website she developed. However, after launching the site with a less-than-reliable commercial hosting provider, she faced a slew of complaints as the site gained traction and traffic. Issues included slow website performance and attempts at ransomware attacks, which fortunately were thwarted. Consequently, she opted to temporarily migrate her website and database to AWS by running an EC2 instance with a MySQL setup in a public subnet.

To improve her website further, our team was brought in to implement a more secure and robust platform on AWS. We aimed to adhere to the latest best practices while staying within budget, ensuring the new site would be more resilient and secure, with the ability to return data from the query page. A detailed list was created to outline the requirements.

- Provide secure hosting of the MySQL database.
- Provide secure access for an administrative user.
- Provide anonymous access to web users.

- Run the website on a t2.small EC2 instance, and provide Secure Shell (SSH) access to administrators.
- Provide high availability to the website through a load balancer.
- Store database connection information in the AWS Systems Manager Parameter Store.
- Provide automatic scaling that uses a launch template.

2.0 SOLUTION DIAGRAM ILLUSTRATION

The first step in creating a solid cloud computing platform is to carefully design its architecture. This helps everyone involved have a clearer vision and a better grasp of what the platform needs to accomplish. Take a look at Figure 1, which illustrates the overall architecture crafted for the client using AWS architecture icons. As we mentioned earlier, this platform will be built on the AWS cloud. It's designed for one region, which will include two availability zones to reach a broader audience and population. We'll set up one Virtual Private Cloud (VPC) to allow users to access the client's website. Each availability zone will feature one public subnet and one private subnet. To enhance security for the private subnet, we'll implement a Network Address Translation (NAT) gateway, enabling instances in the private subnet to connect to the internet and fetch resources without needing an IP address. Additionally, we'll have Auto Scaling groups and an Application Load Balancer (ALB) in place to keep an eye on the system's health, ensuring it remains highly available and scalable.

Moving on to Figure 2, you'll see that we'll establish a total of four security groups. The website traffic is configured by setting up inbound and outbound rules, allowing data to flow from the first tier to the fourth tier. Users will gain access through the internet gateway to utilize the website's resources, and their access to additional resources will depend on their authority level. Anonymous users will have secure access to limited resources via Hypertext Transfer Protocol (HTTP), while admins will enjoy secure access to a broader range of resources through Secure Shell Protocol (SSH).

The first layer of security is known as the bastion tier, which includes bastion host instances that serve as the initial defense against cyber threats. Only administrators have access to this level. Next up is the app tier ALB, featuring an Application Load Balancer (ALB) that efficiently distributes users across two availability zones before directing them to the app instances. This load balancer plays a crucial role in managing traffic and ensuring the app instances remain healthy. Moving on to the third security level, we have the app tier, where users arrive to access the application after being sorted in the previous stage. Finally, we reach the data tier, which is exclusively accessible from the third tier by authorized personnel who need to interact with the database resources.

In the upcoming section, we'll dive into the AWS services and tools that were utilized to build the cloud computing platform for the client. The components we'll discuss next aren't listed in any specific order but are grouped into three categories: deployment, security, and high availability and scalability. When setting up the cloud platform in AWS, the process began with creating the VPC, followed by setting up subnets, an internet gateway, a NAT gateway, route tables, network ACLs, security groups, an Amazon S3 bucket, launch templates, a load balancer, auto-scaling groups, an Amazon RDS database, parameter store, and finally establishing the SSH connection.

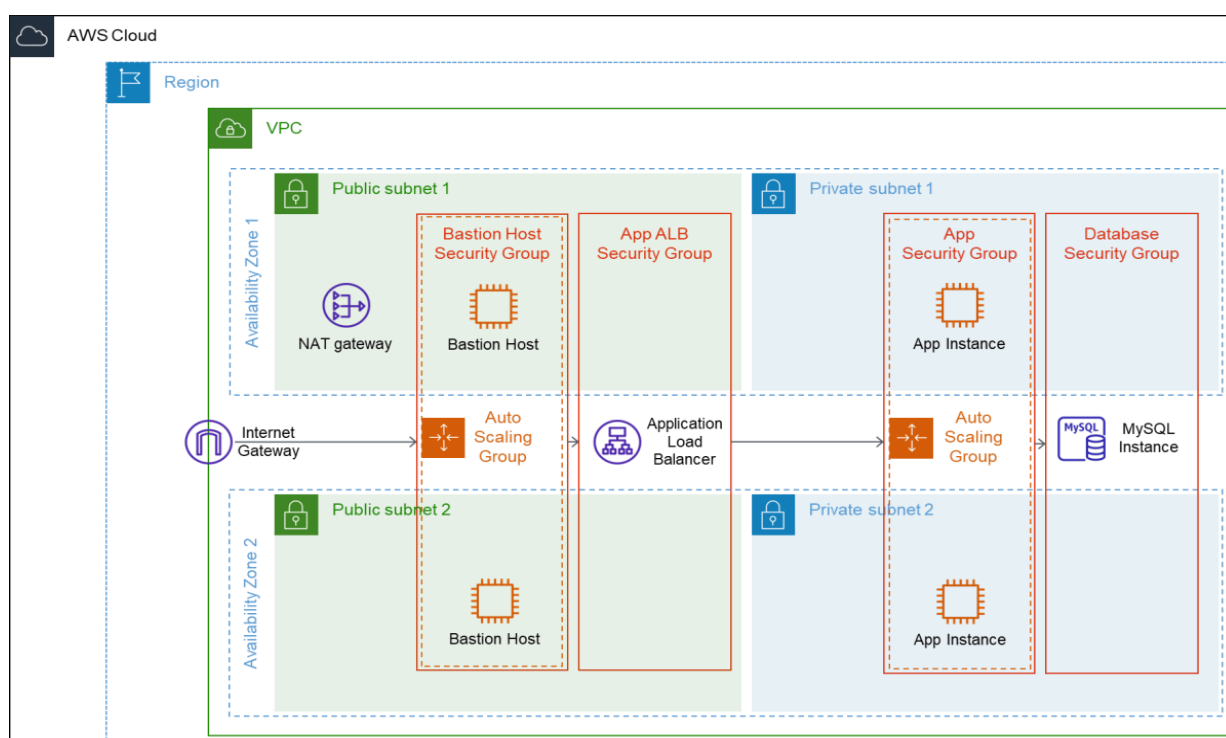


Figure 1: Design of Overall Architecture

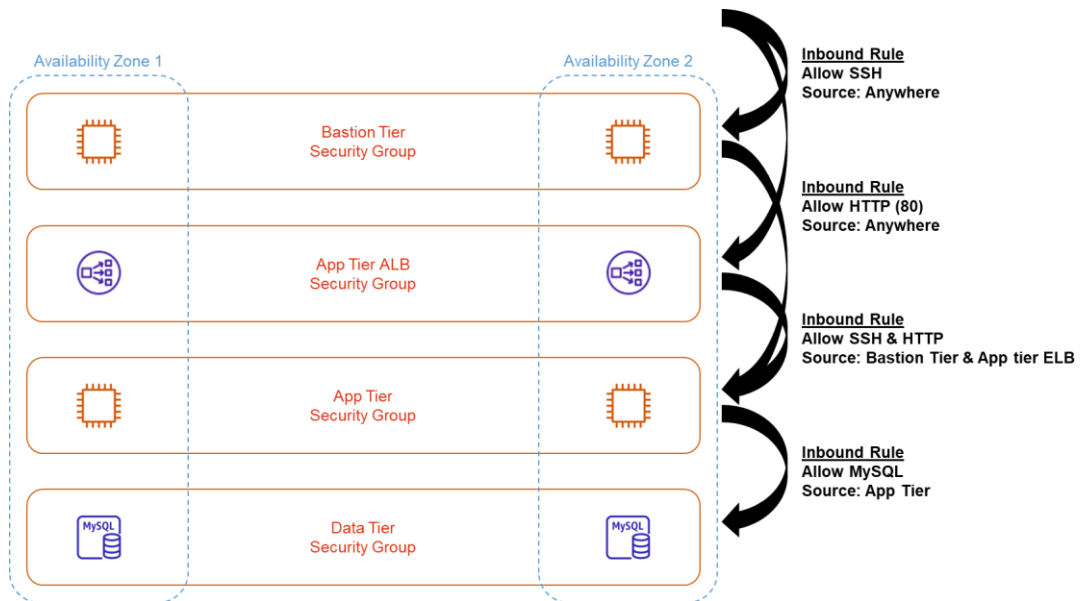


Figure 2: Design of Overall Architecture

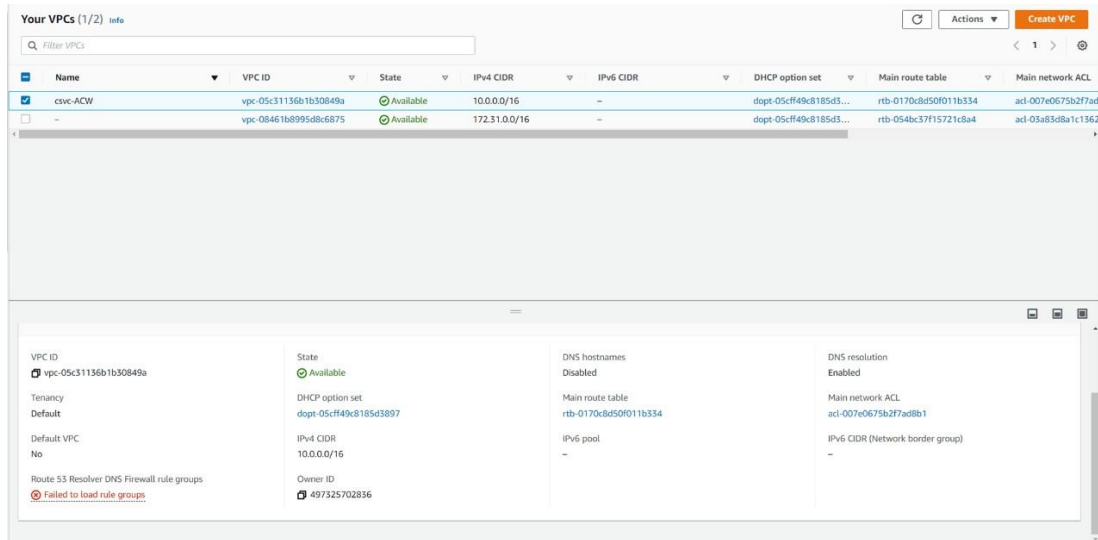
3.0 DESIGN DECISIONS

3.1 DEPLOYMENT

3.1.1 VPC

With the help of Virtual Private Cloud (VPC), users can establish a virtual private network area that can run or utilize Amazon's global resources within the AWS cloud infrastructure. Users can fully manage their resources by using VPC, including subnet creation, route tables, security, connection, and other aspects of the virtual network that are not publicly available online (Amazon Web Services, 2022).

In AWS, a VPC named "csvc-ACW" was established. To specify the 16-bit (65536) network IP address to be utilized, the Internet Protocol version 4 (IPv4) Classless Inter-Domain Routing (CIDR) setting was set to 10.0.0.0/16. The VPC developed for this task is displayed in Figure 3 below.



Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL
csvc-ACW	vpc-05c31136b1b30849a	Available	10.0.0.0/16	-	dopt-05c449c8185d3...	rtb-0170c8d50f011b334	acl-007e0675b2f7ad1
-	vpc-08461b8995d8c6875	Available	172.31.0.0/16	-	dopt-05c449c8185d3...	rtb-054bc37f15721c8a4	acl-05a83d8a1c1362

VPC ID	State	DNS hostnames	DNS resolution
vpc-05c31136b1b30849a	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-05c449c8185d3897	rtb-0170c8d50f011b334	acl-007e0675b2f7ad1
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	10.0.0.0/16	-	-
Route 53 Resolver DNS Firewall rule groups	Owner ID		
Failed to load rule groups	497325702836		

Figure 3: VPC

3.1.2 SUBNET

Virtual Private Clouds (VPC) can provide global resources while a subnet is only in a region subset of the VPC, this is to run AWS resources into very specific network. A subnet can be created by specifying one or a list of IP addresses within the VPC's IPv4 CIDR block. Unlike VPC, which can span in several availability zones, a subnet can only span one availability zone. A VPC can have one or more types of subnet, although for most applications only public subnet and private subnet will be used. public subnet provides route for outbound traffic to the public internet, while private subnet permits end users to access resources like a database (Amazon Web

Services, 2022). It can be said that the envelope of security offered by cloud infrastructure is greater the more that subnets multiply.

Four subnets were created where each public subnet and private subnet was created for one availability zone, as per Table 1 and Figure 4. Each subnet also has 256 (28) address in separate address range. To provide a greater level of security, two set of subnets were created, while still avoiding wastage of available IP addresses. Public subnet_1 and private subnet_1, were allocated to us-east-1a, while public subnet_2 and private subnet_2, were allocated to us-east-1d

Table 1: Subnet

Subnet Name	Subnet Type	Availability Zone	Subnet Address
Public Subnet_1	Public	us-east-1a	10.0.0.0/24
Public Subnet_2	Public	us-east-1d	10.0.10.0/24
Private Subnet_1	Private	us-east-1a	10.0.20.0/24
Private Subnet_2	Private	us-east-1d	10.0.30.0/24

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Availability Zone
Public Subnet_2	subnet-0459ecffec07e1d	Available	vpc-05c31136b1b30849a csw...	10.0.10.0/24	--	251	us-east-1d
Public Subnet_1	subnet-058c2cae9c16ca7be	Available	vpc-05c31136b1b30849a csw...	10.0.0.0/24	--	251	us-east-1a
Private Subnet_2	subnet-0faa1cb045e87b93f	Available	vpc-05c31136b1b30849a csw...	10.0.30.0/24	--	251	us-east-1d
Private Subnet_1	subnet-03afc764f6fb634fa	Available	vpc-05c31136b1b30849a csw...	10.0.20.0/24	--	251	us-east-1a

Figure 4: Subnet

3.1.3 INTERNET GATEWAY

Internet Gateway (IGW) is a critical part of the VPC to enable communication between a VPC and the internet. An IGW will allow two-way communications to either initiate communication from internet resources to the public subnet or provide public access to other public resources such as an instance. In other words, if there is no IGW in the VPC, no resources can be accessible via the internet. An IGW only exists in the VPC, and not necessarily in any availability zone. This means that for a subnet to become a public subnet, a route table must be associated with the subnet that directs the internet traffic through an IGW (Amazon Web Services, 2022).

An IGW called “igw-ACW” was created and attached to VPC, as shown in Figure 5. The public route table was updated to allow the created subnet public access route to the internet through the IGW.

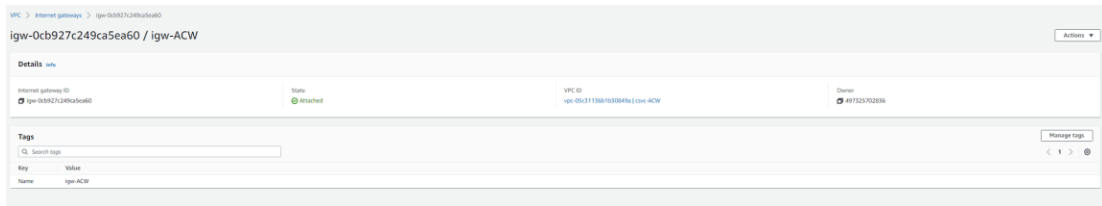


Figure 5: Internet Gateway

3.1.4 NAT GATEWAY

The purpose of a NAT Gateway (NGW) is fundamentally different from that of an Internet Gateway (IGW). It blocks internet access to instances with public IP addresses and instead provides one-way internet connectivity to instances in private subnets. NAT Gateways are created for every availability zone to improve VPC security (Amazon Web Services, 2022). The NGW for this assignment is outlined in Figure 6. Before the NAT Gateway (NGW) was created, an elastic IP address 52.7.104.95 was allocated to be associated with the public NGW named “ngw_ACW” in public_subnet_1. The purpose of creating “ngw_ACW” was to ensure the instances in private_subnet_1 have continuous internet connectivity to update and download critical security patches while ensuring no user from the internet can initiate any connections to these instances. After creating the “ngw_ACW”, the route table associated with the private_subnet_2 was updated.



Figure 6: NAT Gateway

3.1.5 ROUTE TABLE

Imagine that you are looking at a spreadsheet, but this one is quite specialized—it is known as a route table. In it, you find a set of instructions spelled out in routes. The spreadsheet guides you on the best path to take, whether it be from a gateway or subnet, enabling the flow of network traffic. Every route laid out in the route table needs to have a clear traffic forwarding destination and target, as explained in the AWS documentation (2022).

Public and private route tables can be viewed in Figures 7 and 8, respectively. The data compiled for both types of tables are organized and displayed in Table 2. With regard to the public route table, a route with the destination of 0.0.0.0/0 and the target `igw-0cb927c249ca5ea60` was created, sending all internet-bound traffic to the Internet Gateway. Similarly, the private route table has the default route of 0.0.0.0/0, but with the target `nat-02dd75c6d48041d4b`, directing internet-bound traffic to the NAT Gateway. These two routing rules are crucial to allow the subnets to have internet access through the IGW or NGW, respectively. The destination 0.0.0.0/0 is a wildcard notation referring to all IPv4 traffic directed to either the IGW or NGW. Both route tables share an identical entry for the local route, which facilitates communication within the defined VPC.

Details Info			
Route table ID <code>rtb-080269fed1a4e7bd7</code>	Main <code>No</code>	Explicit subnet associations <code>2 subnets</code>	Edge associations <code>-</code>
VPC <code>vpc-05c31136b1b30849a</code> <code>csvc-ACW</code>	Owner ID <code>497325702836</code>		

Routes	Subnet associations	Edge associations	Route propagation	Tags
--------	---------------------	-------------------	-------------------	------

Routes (2)				Edit routes
<input type="text" value="Filter routes"/>	<input type="text" value="Both"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Destination	Target	Status	Propagated	
0.0.0.0/0	<code>igw-0cb927c249ca5ea60</code>	Active	No	
10.0.0.0/16	local	Active	No	

Figure 7: Public Route Table

Details Info			
Route table ID <code>rtb-001dd5c029b72f408</code>	Main <code>Yes</code>	Explicit subnet associations <code>2 subnets</code>	Edge associations <code>-</code>
VPC <code>vpc-05c31136b1b30849a</code> <code>csvc-ACW</code>	Owner ID <code>497325702836</code>		

Routes	Subnet associations	Edge associations	Route propagation	Tags
--------	---------------------	-------------------	-------------------	------

Routes (2)				Edit routes
<input type="text" value="Filter routes"/>	<input type="text" value="Both"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Destination	Target	Status	Propagated	
0.0.0.0/0	<code>nat-02dd75c6d48041d4b</code>	Active	No	
10.0.0.0/16	local	Active	No	

Figure 8: Private Route Table

Table 2: Route Table

Route Table	Destination	Target
Public	0.0.0.0/0	igw-0cb927c249ca5ea60
	10.0.0.0/16	local
Private	0.0.0.0/0	nat-02dd75c6d48041d4b
	10.0.0.0/16	local

3.2 SECURITY

AWS's top priority is to provide secure cloud services to reduce the risk of security incidents and to eliminate vulnerabilities through well-defined access levels. Additionally, users can secure their data in AWS by managing rules, inbound and outbound traffic, and subnet associations. In this case, the customer is transferring the website and database to a public subnet. The use of Network ACLs, Security Groups, Parameters, and SSH connections will improve the security and durability of the website and database.

3.2.1 NETWORK ACL

A Network Access Control List (NACL) is an extra layer of protection that act as filter out to permit the visitors in or out of the subnet. The traffic inside times in a same subnet will not be filtered out as the NACL controls visitors out of doors of the subnet. NACL has rules to be described at the inbound and outbound regulations so one can manipulate the traffic. The described policies might be choose from lowest to the highest rule range in addition to the execution collection of the rules. Sometimes, few numbers of NACL are created to most effective allow specific site visitors to enter and go out at the precise subnet.

A NACL associated with the 4 created subnets was created and named as “acl-ACW” as proven in Figure nine, Figure 10, and Figure eleven. The regulations for each inbound and outbound policies of the NACL which contained a rule variety of one hundred and asterisk. Based on the defined guidelines, all kind of traffics had been allowed to bypass through all of the public and private subnets.

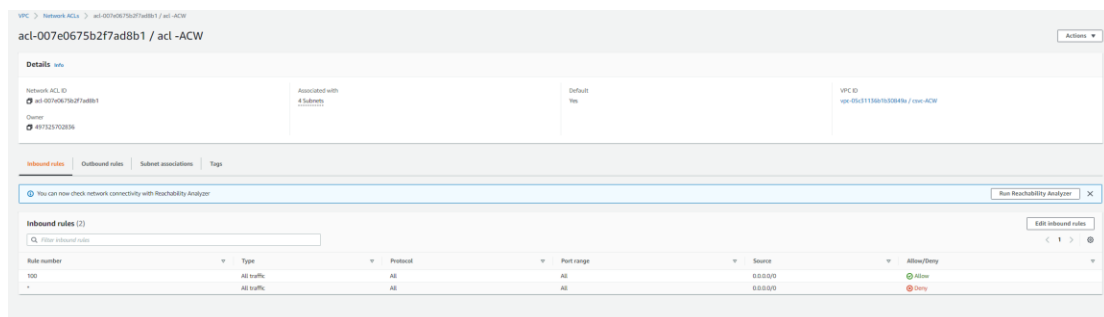


Figure 9: Network ACL (Inbound rules)

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
1	All traffic	All	All	0.0.0.0/0	Deny

Figure 10: Network ACL (Outbound rules)

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
Private Subnet_1	subnet-05a75a0e5b2f7ad8b1	aci-007e0675b2f7ad8b1 / acl -ACW	us-east-1a	10.0.20.0/24	-
Private Subnet_2	subnet-04b75a0e5b2f7ad8b1	aci-007e0675b2f7ad8b1 / acl -ACW	us-east-1b	10.0.10.0/24	-
Private Subnet_3	subnet-03a75a0e5b2f7ad8b1	aci-007e0675b2f7ad8b1 / acl -ACW	us-east-1c	10.0.30.0/24	-
Private Subnet_4	subnet-02a75a0e5b2f7ad8b1	aci-007e0675b2f7ad8b1 / acl -ACW	us-east-1d	10.0.40.0/24	-

Figure 11: Network ACL (Subnet associations)

3.2.2 SECURITY GROUPS

Functioning as a virtual firewall, the security group oversees both incoming and outgoing traffic for EC2 instances. Inbound and outbound rules regulate the movement of traffic to and from that instance. The main role of the AWS security group is to protect the cloud environment by managing the traffic permitted into the EC2 instances. At the instance level, security groups guarantee that all traffic, whether public or private, passes through specified ports and protocols.

Four security groups exist, each with distinct inbound rules that define their respective usages. The sole rules that are the same are found in the outbound rule. This collection of outbound rules allows all outgoing traffic.

The inbound rule in Bastion-secg permits only the administrator to access the EC2 bastion layer through SSH, configured to use port range 22, which is the standard SSH port. To enable application users to connect via HTTP, the Appelb-secg inbound rule is configured for HTTP on port 80. Apptier-secg will subsequently utilize Bastion-secg and Appelb-secg as its incoming source. This enables the administrator and/or application users to access EC2 from the ELB and bastion layers. A security group for the database is established with the name Db-secg

to permit an inbound connection. This incoming connection is MYSQL/Aurora to provide app-tier level access to the database.

Table 3: Security Groups

Security Group Name	Inbound type	Inbound Protocol	Inbound Port range	Inbound Source	Description
Bastion-secg	SSH	TCP	22	0.0.0.0/0	Allows SSH connection by admins
Appelb-secg	HTTP	TCP	80	0.0.0.0/0	Allows HTTP connection by application users
Apptier-secg	SSH HTTP	TCP	22 80	Bastion-secg Appelb-secg	Allows HTTP and SSH connections from Bastion Host and Elastic Load Balancer
Db-secg	MYSQL/Aurora	TCP	3306	Apptier-secg	Allows MYSQL/Aurora connection from application to database

Table 3: Security Groups

EC2 > Security Groups > sg-07fe25cf8dcddeeb - Bastion-secg

sg-07fe25cf8dcddeeb - Bastion-secg

Actions

Details

Security group name

Bastion-secg

Security group ID

sg-07fe25cf8dcddeeb

Description

Allows SSH connection by admins

VPC ID

vpc-05c31136b1b30849a

Owner

497325702836

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (1/1)

Manage tags

Edit inbound rules

Filter security group rules

< 1 >

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sgr-07567ffbf40fa450	IPv4	SSH	TCP	22	0.0.0.0/0	Allows SSH connection by admins

Figure 12: Security Groups (Bastion-secg)

EC2 > Security Groups > sg-0e8699adb76eb3313 - Appelb-secg

sg-0e8699adb76eb3313 - Appelb-secg

Actions

Details

Security group name

Appelb-secg

Security group ID

sg-0e8699adb76eb3313

Description

Allows HTTP connection by application users

VPC ID

vpc-05c31136b1b30849a

Owner

497325702836

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (1/1)

Manage tags

Edit inbound rules

Filter security group rules

< 1 >

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	Appelb-secg	sgr-017e966719044bc...	IPv4	HTTP	TCP	80	0.0.0.0/0	Allows HTTP connection by applic...

Figure 13: Security Groups (Appelb-secg)

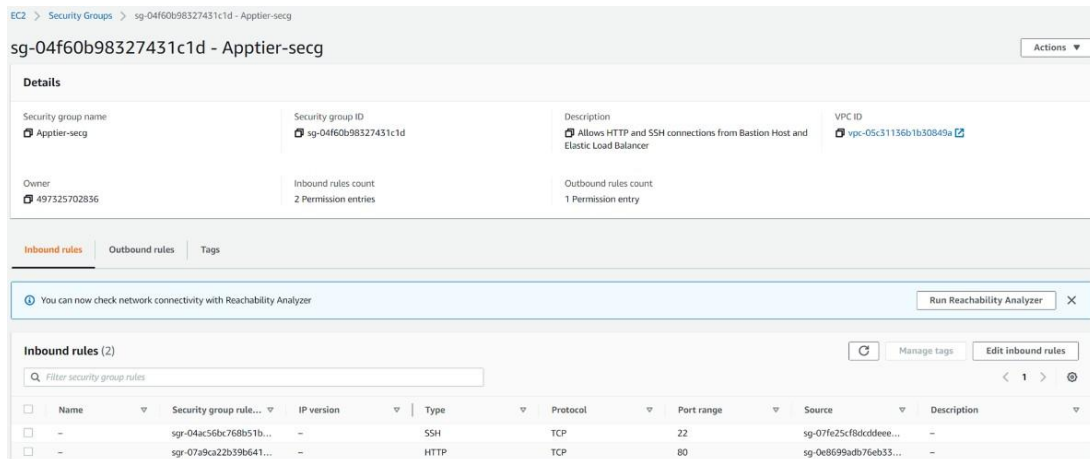


Figure 14: Security Groups (Apptier-secg)

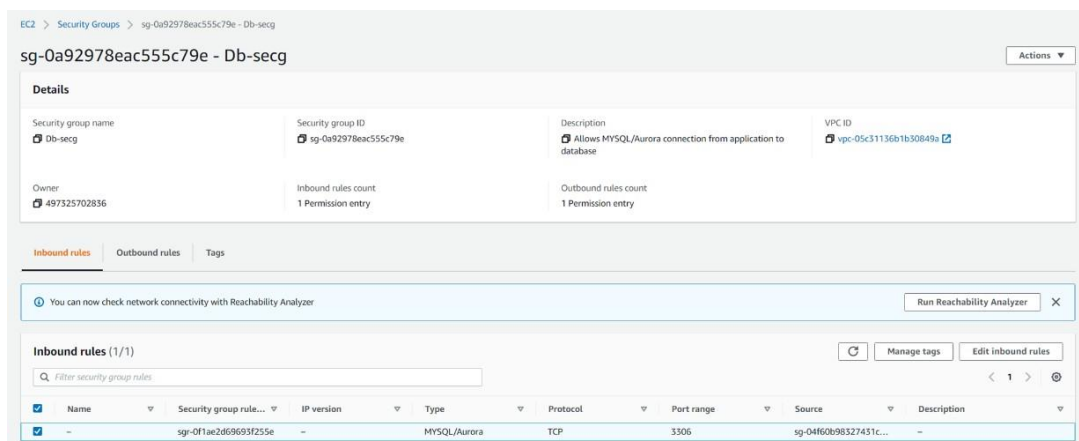


Figure 15: Security Groups (Db-secg)

3.2.3 PARAMETER STORE

The Parameter Store is a feature of the AWS Systems Manager that offers a secure and safe way to store and manage configuration data and secrets. The worth of the stored data can be recorded as a straightforward plain text, and it will be encrypted using a distinctive

designated name. The saved information can be accessed in scripts, commands, documents, etc., by utilizing the established unique identifier. The Parameter Store can offer security and encryption for the data stored, safeguarding against theft. (Amazon, 2022c) Table 4 and Figure 16 display the parameters that were saved in the Parameter

Warehouse. The parameters consist of the elements required to create a link between the PHP application on the EC2 app instances and the system's database, enabling admins to access the database.

Table 4: Parameter Store

Parameter	Type	Value
/example/endpoint	String	database-01.c4bzmp6l6ub1.us-east-1.rds.amazonaws.com
/example/username	String	admin
/example/password	SecureString	assignment
/example/database	String	csvc

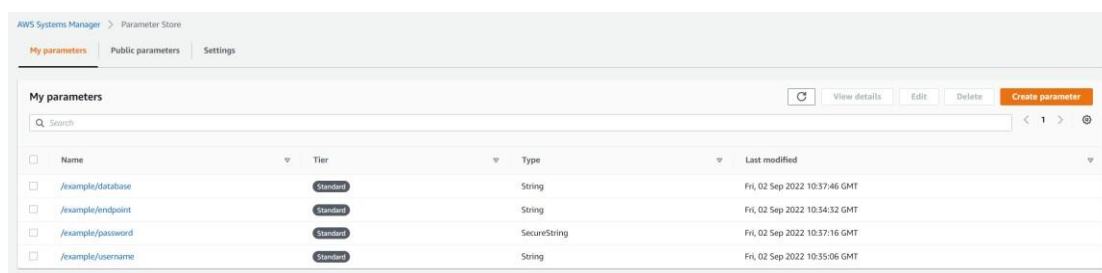


Figure 16: Parameter Store

3.2.4 SSH CONNECTION

A Secure Shell (SSH) is a protocol for linking two computers. The SSH connection secures the link between both computers for viewing, monitoring, and transferring data by demanding a keypair or password for authentication. The task specifies the use of EC2 t2.small instances that operate on Linux, but the actual computer is running a Windows Operating System (OS). Consequently, a Windows PowerShell was utilized to create an SSH connection from Linux to Windows to execute the instances on the physical machine. Figure 17 and Figure 18 presented below illustrate

certain outcomes of linking the AWS Linux instances to the PowerShell operating on a Windows OS physical machine via SSH command. The findings indicate that the SSH connection was successfully created since the terminal could access MySQL, located

within the private subnet.

```
Select ec2-user@ip-10-0-10-7:~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Aiman> cd downloads
PS C:\Users\Aiman\downloads> ssh -i assignment-acw.pem ec2-user@18.204.43.147
Last login: Sat Sep  3 08:41:42 2022 from 1sdn-mc-226-213.tn.net.my

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

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-10-7 ~]$ wget https://keypair0123.s3.amazonaws.com/assignment-acw.pem
--2022-09-03 09:06:53-- https://keypair0123.s3.amazonaws.com/assignment-acw.pem
Resolving keypair0123.s3.amazonaws.com (keypair0123.s3.amazonaws.com)... 52.217.168.225
Connecting to keypair0123.s3.amazonaws.com (keypair0123.s3.amazonaws.com)[52.217.168.225]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1674 (1.6K) [binary/octet-stream]
Saving to: 'assignment-acw.pem.5'
```

Figure 17: SSH Connection 1

```
Select ec2-user@ip-10-0-10-7:~
Length: 1674 (1.6K) [binary/octet-stream]
Saving to: 'assignment-acw.pem.5'

100%[=====>] 1,674  --.-K/s  in 0s

2022-09-03 09:06:53 (91.1 MB/s) - 'assignment-acw.pem.5' saved [1674/1674]

[ec2-user@ip-10-0-10-7 ~]$ chmod 400 assignment-acw.pem
[ec2-user@ip-10-0-10-7 ~]$ ssh -i assignment-acw.pem ec2-user@10.0.30.14
The authenticity of host '10.0.30.14 (10.0.30.14)' can't be established.
ECDSA key fingerprint is SHA256:EHNZmqZ5Sgwh8zA0gp0bDiBjcsIrNa0sgLxK154IEmM.
ECDSA key fingerprint is MD5:f4:b4:7b:a9:9a:b3:c3:9d:cd:4e:bf:64:0a:b9:77:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.30.14' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-30-14 ~]$ mysql -u admin -p --host database-01.c4bzmp6l6ub1.us-east-1.rds.amazonaws.com
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 294
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.
```

Figure 18: SSH Connection 2

3.3 HIGH AVAILABILITY AND SCALABILITY

Ensuring high availability and scalability is crucial when creating a cloud platform, as these factors are vital for maintaining the system's health and longevity. A system possesses high availability when it can operate continuously without interruption, while a system has high scalability when the Information Technology (IT) resources utilized to create it can be adjusted up or down according to demand. The tools and services described in this section will enable the system developed to achieve high availability and scalability.

3.3.1 AMAZON S3

Amazon Simple Storage Service (S3) is a service that enables the storage of data as objects within containers referred to as buckets. The items held are regarded as the private possessions of the bucket owner, and others may access them only with permission, enhancing the security of the system in place. The Amazon S3 service provides significant availability and scalability since the S3 bucket can hold an unlimited number of objects, which can be accessed anytime and from anywhere via the internet. (Amazon, 2022f)

As illustrated in Figure 19, a bucket called “keypair0123” was established, and 3 objects were saved inside it. The initial item is the “assignment-acw.pem,” which holds the private key for SSH access. The second item is the “Countrydatadump.sql,” which is the SQL dump file of the MySQL database that includes sample data provided by the client. The third item is “Example.rar,” which holds the PHP: Hypertext Preprocessor (PHP) application source code and image files for the client’s website, set to be deployed in the EC2 instance within the App Tier security group.

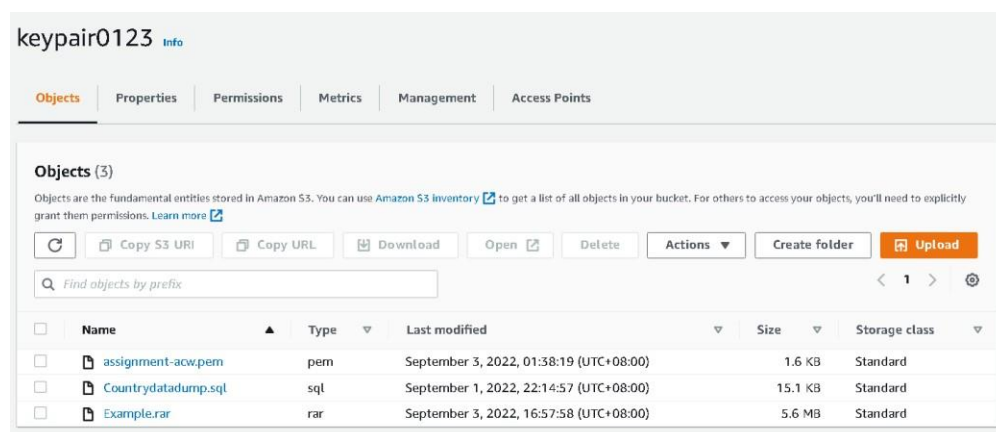


Figure 19: S3 Bucket

3.3.2 LAUNCH TEMPLATE

A launch template is a template that includes details about the instance settings. The launch template enables the initiation of several EC2 instances according to the specifications outlined in the template. It serves as a helpful resource for easily generating various editions of the same launch template. This tool enables scalability within the system since a template can be easily adjusted, allowing for the launch of multiple instances according to the needs. (Amazon, 2022g)

According to the created architecture, two separate EC2 instances need to be initiated, with one designated for the public subnets and the other for the private subnets. Beginning with the launch template for the instances within the public subnets, the launch template titled “bastion-template-ACW” was established as illustrated in Table 5 and Figure.

20. This template will be utilized to deploy 2 bastion host instances in both public subnets to serve as the initial layer of defense for the system. The t2.small instance will be utilized for the EC2 as required. Using the specified user data, the initiated instance will obtain the private key file from the S3 bucket and configure it for read access. This will grant the administrative personnel secure access to the system through SSH.

Table 5: Public Launch Template

Launch Template	bastion-template-ACW
Resource tag	Key: Name
	Value: bastion
	Resource type: Instances
Amazon machine image (AMI)	Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volute Type 64-bit (x86)
Instance type	t2.small
Key pair name	Assignment-acw
Storage volume	Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))
Network interface	Security Group: Bastion-sg
	Public IP auto-assign enabled
User data	<pre>#!/bin/bash wget https://keypair0123.s3.amazonaws.com/assignment.pem chmod 400 assignment.pem</pre>

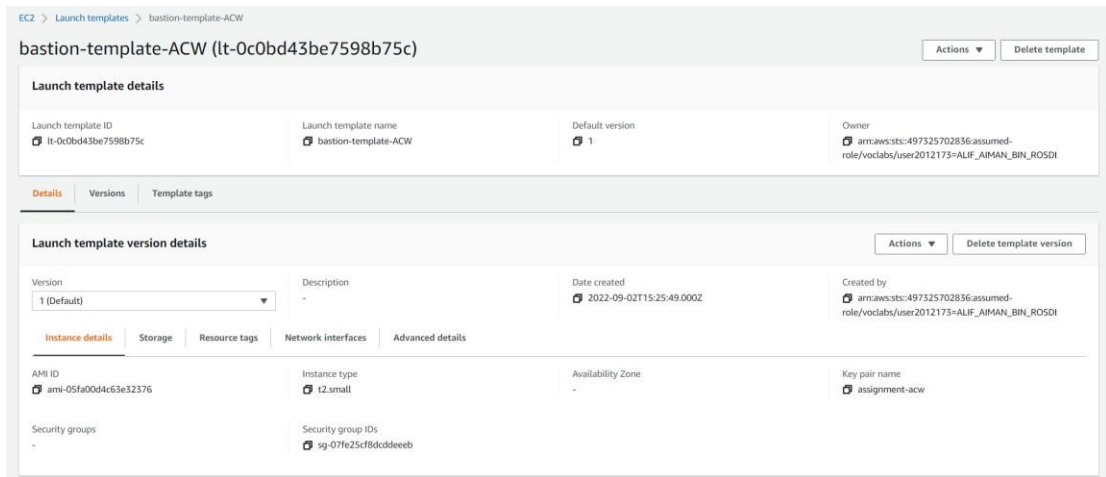


Figure 20: Public Launch Template

The launch template designated for initiating instances in the private subnets is the “ec2-template-ACW,” which was established as illustrated in Table 6 and Figure 21. This template will be utilized to initiate 2 app instances for each private subnet, enabling users to access the application. The EC2 instance was configured as t2.small, as per the requirement. According to the specified user data, the initiated instance will begin by installing the Apache Web Server and PHP, followed by fetching the PHP application source code from the S3 bucket, then installing the AWS SDK for PHP, and finally activating the web server by configuring the host for the PHP application.

Table 6: Private Launch Template

Launch Template	ec2-template-ACW
Resource tag	Key: Name
	Value: app-tier
	Resource Type: Instances
Amazon machine image (AMI)	Amazon Linux 2 AMI (HVM) – Kernel 5.10, SSD Volute Type 64-bit (x86)
Instance type	t2.small
Key pair name	Assignment-acw
Security Group	App-tier-sg
Storage volume IAM Instance Profile	Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))
	LabInstanceProfile (LabRole)
User data	#!/bin/bash # Install Apache Web Server and PHP yum install -y httpd mysql

	<pre>amazon-linux-extras install -y php7.2 # Download Lab files wget https://keypair0123.s3.amazonaws.com/Example.zip unzip Example.zip-d /var/www/html/ # Download and install the AWS SDK for PHP wget https://github.com/aws/aws-sdk- php/releases/download/3.62.3/aws.zip unzip aws -d /var/www/html # Turn on web server chkconfig httpd on service httpd start</pre>
--	--

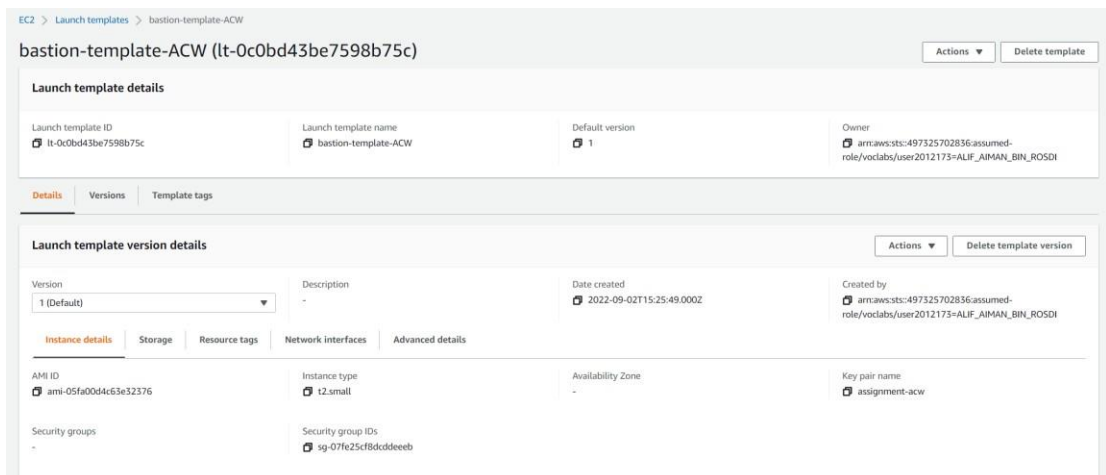


Figure 21: Private Launch Template

3.3.3 LOAD BALANCER

Elastic Load Balancing (ELB) is a service that automatically distributes incoming traffic among various target groups in one or more Availability Zones. ELB is extremely beneficial for maintaining the high availability of the system as it can track and redirect traffic to other healthy targets, ensuring an even workload distribution throughout the system, preventing any target from becoming overloaded and keeping high availability intact. Additionally, the ELB is scalable since it automatically adjusts the load balancer according to the volume of incoming traffic and the consequent workload. (Amazon, 2022h)

An Application Load Balancer (ALB), labeled “App-elb-ACW”, was established to oversee the EC2 instances within the App Tier since these instances will run the application for users to reach the resources, which are anticipated to experience significant user traffic. The load balancer can oversee the condition of the

instances, allocate traffic and workloads properly to guarantee that both

Instances are in a good state. Table 7, Figure 22, Table 8, and Figure 23 display the target audience and settings for the developed load balancer.

Table 7: Target Group

Target group	app-tier
Listener	HTTP-80

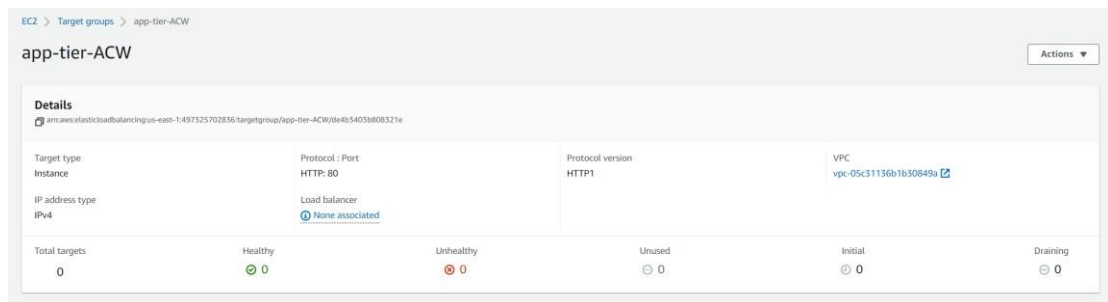


Figure 22: Target Group

Table 8: Load Balancer

Name	App-elb-ACW
Scheme	Internet-facing
IP Address Type	ipv4
Listener	HTTP:80
Availability Zones	us-east-1a
ipv4	us-east-1d
Security Group	App-elb-sg

Basic Configuration	
Name	App-elb-ACW
ARN	arn:aws:elasticloadbalancing:us-east-1:497325702836:loadbalancer/app/App-elb-ACW/356c9fe24c85a25c 🔗
DNS name	App-elb-ACW-77765318.us-east-1.elb.amazonaws.com 🔗 (A Record)
State	Provisioning
Type	application
Scheme	internet-facing
IP address type	ipv4
	Edit IP address type
VPC	vpc-05c31136b1b30849a 🔗
Availability Zones	subnet-0459ecffcee07e1d - us-east-1d 🔗 IPv4 address: Assigned by AWS
	subnet-058c2cae8c16ca7be - us-east-1a 🔗 IPv4 address: Assigned by AWS
	Edit subnets
Hosted zone	Z35SXDOTRQ7X7K
Creation time	September 1, 2022 at 11:36:09 PM UTC+8
Security	
Security groups	sg-0e8699adb76eb3313, Appelb-secg • Allows HTTP connection by application users
	Edit security groups

Figure 23: Load Balancer

3.3.4 AUTO SCALING GROUPS

Auto Scaling groups are designed to automatically manage a collection of EC2 instances by keeping an eye on their health and adjusting the number of instances as needed. Essentially, the Auto Scaling group ensures that the right number of instances are running based on their health status. For instance, if an instance is found to be unhealthy, it will be terminated and replaced with a new, healthy one. This way, the Auto Scaling group helps meet the demands for high availability and scalability. (Amazon, 2022b)

These Auto Scaling groups will be utilized to launch the EC2 bastion host and application instances using the launch templates that were created earlier. Table 9 outlines the configurations for the Auto Scaling groups set up for both types of EC2 instances, while Figures 24 and 25 illustrate the created Auto Scaling groups. The group named “Bastion-asg-ACW” is specifically for the bastion host instance. It employs the “bastion-template-ACW,” and the instances will be deployed in the public subnets. No load balancer is necessary here since only administrators will access these instances, and the anticipated load is relatively low.

On the other hand, the group called “App-asg-ACW” is designated for the application instance. It utilizes the “ec2-template-ACW,” and these instances will be set up in the private subnets. The previously created load balancer will be employed to manage traffic and ensure that the instances remain healthy, as these instances will be accessed by many users, leading to a high expected load. Health checks will be conducted on all components: for the bastion host, checks will focus on the EC2 instance, while for the app, checks will be performed on both the instance and the load balancer to guarantee high availability.

For the group size, both types of instances are set to the default of 2. However, to accommodate the heavy traffic expected for the app instances, the maximum capacity has been increased to 4, ensuring both high availability and scalability.

Table 9: Auto Scaling Groups

Name		Bastion-asg-ACW	App-asg-ACW
Launch template		bastion-template-ACW	ec2-template-ACW
Subnets		Public Subnet 01 Public Subnet 02	Private Subnet 01 Private Subnet 02
Load Balancing		No load balancer	Attach to app-tier ELB
Health checks		EC2	EC2 ELB
Health check grace period		300 seconds	300 seconds
Group size	Desired capacity	2	2
	Min capacity	2	2
	Max capacity	2	4
Scaling policies		None	None
Tags		Name: bastion	Name: App-tier

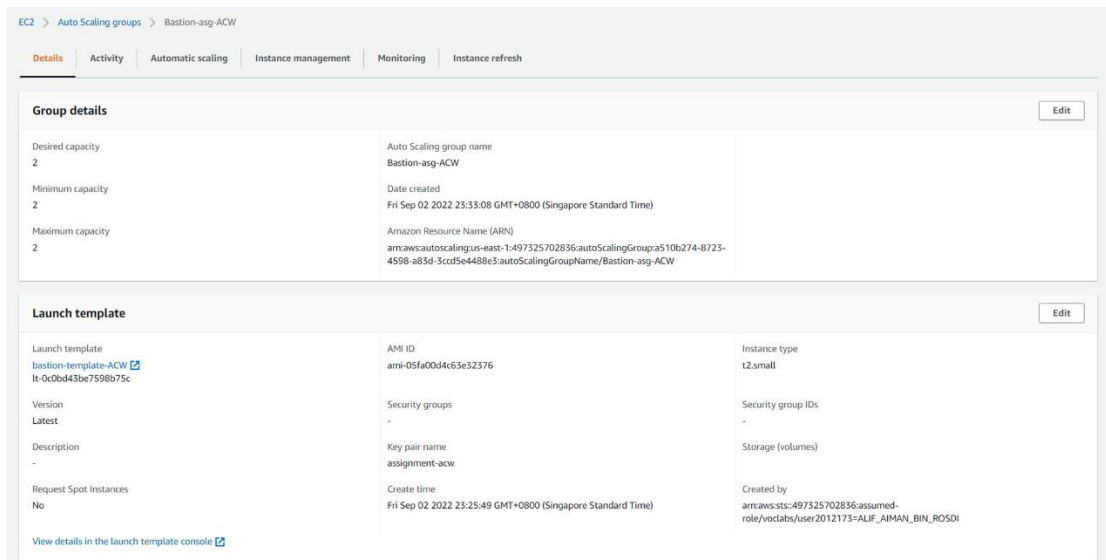


Figure 24: Public Auto Scaling Groups

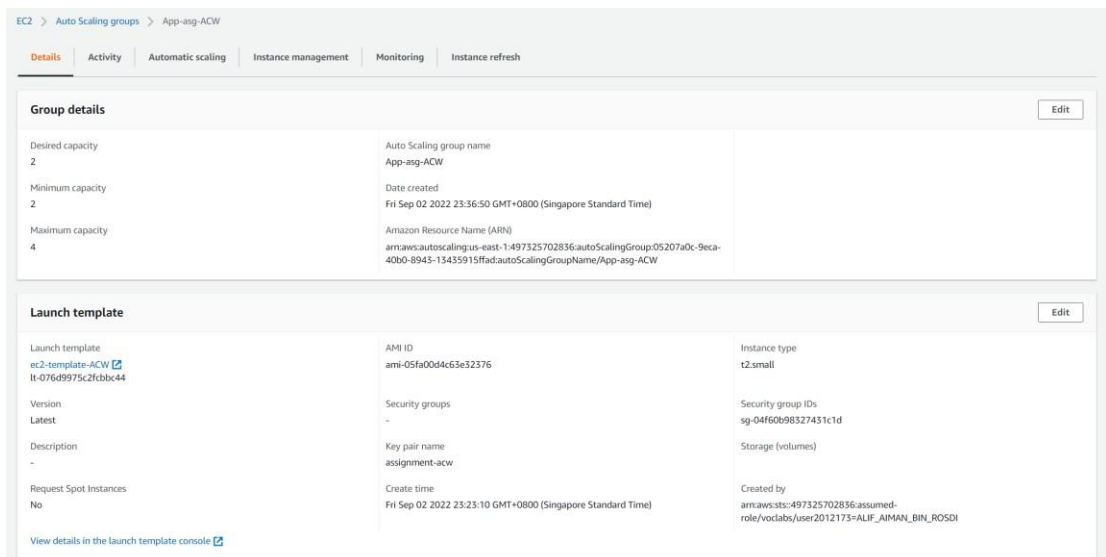


Figure 25: Private Auto Scaling Groups

3.3.5 AMAZON RDS

Amazon Relational Database Service (RDS) comprises various services that facilitate the straightforward establishment, operation, management, and scaling of cloud databases. Amazon RDS offers a budget-friendly solution for database management and migration, ensuring high availability, scalability, and throughput for applications. Amazon RDS supports 7 widely used engines, including Amazon Aurora compatible with MySQL, Amazon Aurora compatible with PostgreSQL, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server. (Amazon, 2022a)

In accordance with the assignment's requirements, the MySQL engine will operate the

database. For optimal security in hosting the database, it will be situated in the private subnets illustrated in Table 10 and Figure 26; more precisely, it will reside in the ultimate security layer known as “db-sg.” The database is protected by being designated as not publicly available and necessitates password authentication for access permission. Consequently, access to the database will be limited to the admins only. Additionally, high availability and scalability are considered, with storage scaling being facilitated, and the designated storage set at 20GiB, allowing a maximum limit of 1000GiB. The setup for the established database is illustrated in Table 11 and Figure 27.

Table 10: RDS Private Subnet

Name	db-sg-acw	
Description	database subnet privates	
VPC	csvc-ACW	
Availability Zone	us-east-1a	Private Subnet 2
	us-east-1d	Private Subnet 1

Subnet group details		
VPC ID	vpc-05c31136b1b30849a	
ARN	arn:aws:rds:us-east-1:497325702836:subgrp:db-sg-acw	
Supported network types	IPv4	
Description	database subnet privates	

Subnets (2)		
Availability zone	Subnet ID	CIDR block
us-east-1d	subnet-0faa1cb045e87bf5f	10.0.30.0/24
us-east-1a	subnet-03afc764ff6b634fa	10.0.20.0/24

Figure 26: RDS Private Subnet

Table 11: RDS Configurations

Engine	MySQL Version 8.0.28	
Production	Dev/Test	
DB identifier	database-01	
Credentials	Master username	admin
	Master password	Assignment-acw
DB instance class	Burstable class: db.t3.micro	

Storage type	General Purpose SSD (gp2)
Allocated storage	20 GiB
Storage autoscaling	Enabled
Maximum storage threshold	1000 GiB
Multi-AZ deployment	Standby instance not created
Subnet group	db-sg-acw
Security group	db-sg
Certificate Authority	rds-ca-2019
Public access	Not publicly accessible
Database port	3306
Database authentication	Password authentication
Enhanced Monitoring	Disabled

RDS > Databases > database-01

database-01 Modify Actions

Summary

DB identifier database-01	CPU 4.09%	Status Available	Class db.t3.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1d

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port Endpoint database-01.c4bzmp6i6ub1.us-east-1.rds.amazonaws.com Port 3306	Networking Availability Zone us-east-1d VPC csvc-ACW (vpc-05c31136b1b30849a) Subnet group db-sg-acw Subnets subnet-03afc764f6fb634fa subnet-0faa1cb045e87bf5f Network type IPv4	Security VPC security groups db-secg (sg-0a92978eac555c79e) Active Public accessibility No Certificate authority rds-ca-2019 Certificate authority date August 23, 2024, 01:08 (UTC+08:00)
--	---	--

Figure 27: RDS Configurations

4.0 FINAL OUTCOME

The results indicate that the website is easily accessible and open to the public. The website's contents are available to the public, allowing them to acquire the information they need. When the traffic is heavy, a new instance is swiftly created through an auto-scaling group. Certain users accessing the website or preparing to do so are redirected to a different site. All the material is the same, and the address will stay unchanged. Only the administrator has the authority to temporarily or permanently shut down the website for maintenance or to move it to a different area. When this occurs, the user or public accessing the website will see an APACHE test page.

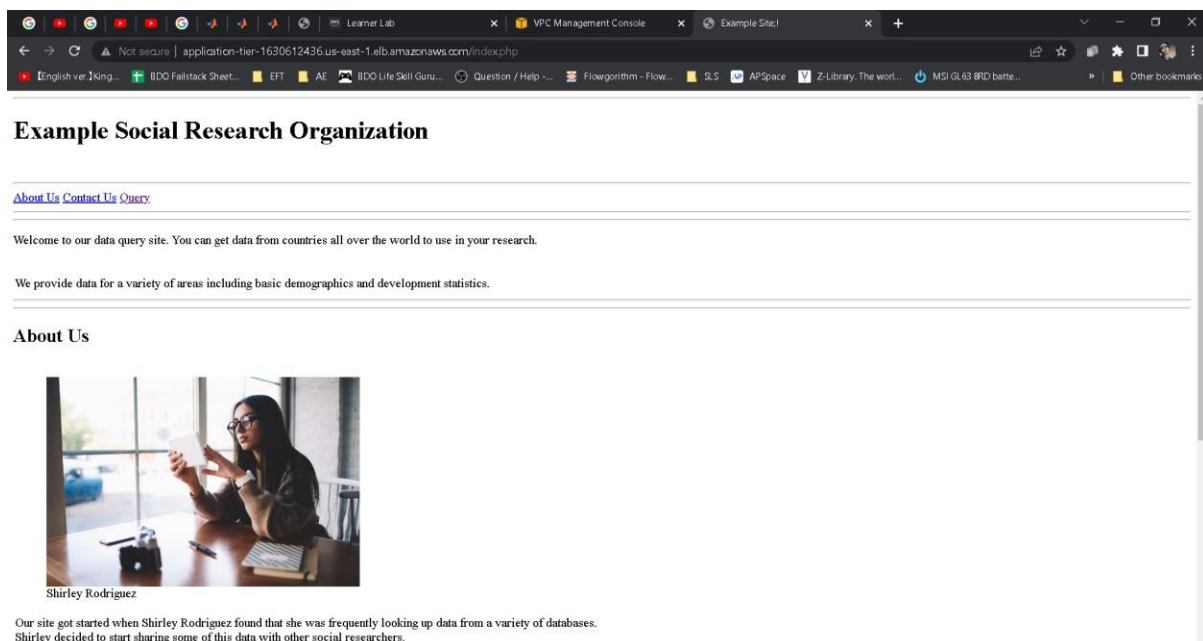


Figure 28: Final Outcome

5.0 ESTIMATED PROJECT PRICING

The total monthly expense of the created system can be determined and acquired by utilizing the AWS Pricing Calculator. The overall monthly expense reached 89.84 USD, and the expenses for AWS services, arranged from highest to lowest, include VPC, Amazon EC2, ELB, Amazon RDS with MySQL, and Amazon S3. A budget of 100 USD was allocated to create the AWS cloud computing platform, and the objective has been met.

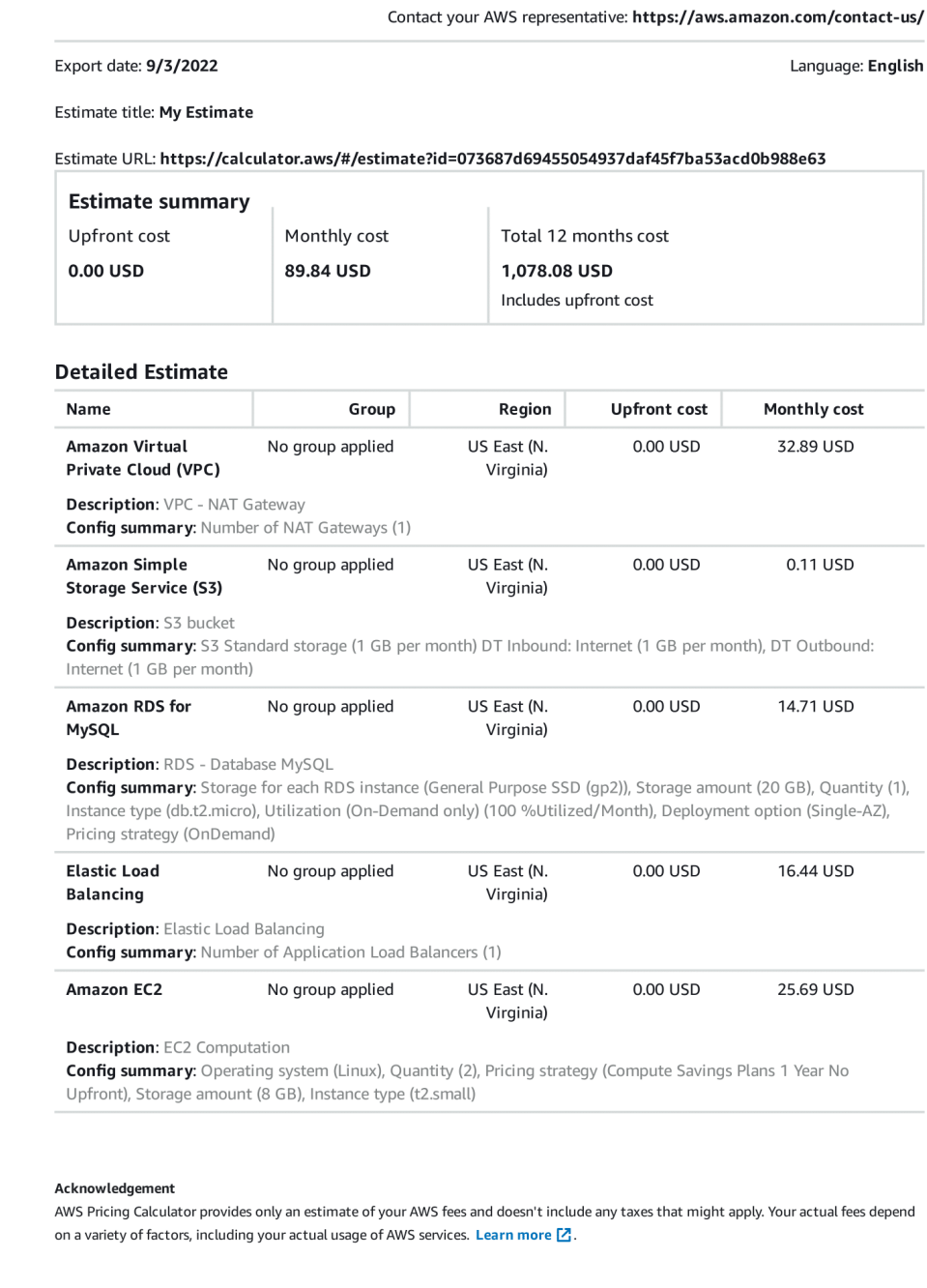


Figure 29: Estimated Project Pricing

6.0 FUTURE ENHANCEMENTS

To keep costs down, we came up with a budget-friendly solution for the client, as we discussed earlier. But with a bit more funding, there's definitely room for some upgrades and enhancements.

When it comes to ensuring high availability and scalability, the system we designed needs to avoid any Single Points of Failure (SPOF). We can tackle this by using a scalable load balancer or setting up an active-standby pair (Amazon, 2022d). Since we already have a load balancer in place, we might want to consider the active-standby approach. For instance, our current setup includes just one NAT gateway in one of the Availability Zones, which is shared between two zones. If that NAT gateway goes down, both private subnets in those zones would lose internet access, making the instances unreachable. To prevent this, we could add another NAT gateway in the other Availability Zone to boost our high availability.

Another strategy for achieving high availability and scalability is to implement Disaster Recovery (DR) plans for the cloud system, which can help protect against data loss or corruption during disasters. We have two options here: the active/passive strategy, which is simpler and more cost-effective, and the active/active strategy, which is more complex and pricier. For the active/passive approach, we'd use a backup and restore method, with Amazon RDS snapshots being the recommended service. This would allow us to replicate database data from the active region to a standby region, ready for recovery in case of regional disasters. On the other hand, the active-active strategy involves multiple active regions that handle traffic in their respective areas. If one region fails, the other active region can take over by redirecting traffic from the failed region until everything is back to normal.

7.0 CONCLUSION

To wrap up the work we did for the assignment, our team was given the task of using AWS to create a secure platform for hosting the client's website. We needed to ensure that the site was not only highly secure against cyber-attacks but also capable of handling a lot of traffic with high availability and scalability. We set up the AWS system in one region, which included a VPC to access the website. Within that VPC, we created one public subnet and one private subnet. The Internet gateway was put in place to let users access the public subnet, while the NAT gateway allowed access to the private subnet. We also established a Route Table to manage the traffic directions for both the Internet and NAT gateways.

On the security front, we implemented a Network ACL to clearly define the inbound and outbound rules for each Security group. We set up four Security groups: the first tier was for the bastion instance, the second for the app ALB, the third for the app instance, and the fourth for the database. The Parameter Store added an extra layer of security by encrypting the parameter data needed for the app instance to connect to the database. Additionally, we established an SSH connection to ensure that only administrators could access the bastion host and the database.

To ensure high availability and scalability, we duplicated the public and private subnets to create two Availability Zones, which helps manage heavy incoming traffic. We also created an Amazon S3 bucket to store the client's website PHP source code and data, along with the SSH private key. Two Launch Templates were set up to launch the bastion host and app instances through Auto Scaling groups. A Load Balancer was created to efficiently manage the traffic directed to the app instances by distributing it across the two Availability Zones. Finally, we set up an Amazon RDS database using the MySQL engine to store the client's website data. In the end, the migration was a success, and the client's website was accessible. The total project cost came to \$89.94.

REFERENCES

- Amazon. (2022a). *Amazon RDS*. Retrieved from AWS:
[https://aws.amazon.com/rds/#:~:text=Amazon%20Relational%20Database%20Service%20\(Amazon,scale%20databases%20in%20the%20cloud.](https://aws.amazon.com/rds/#:~:text=Amazon%20Relational%20Database%20Service%20(Amazon,scale%20databases%20in%20the%20cloud.)
- Amazon. (2022b). *Auto Scaling groups*. Retrieved from AWS:
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-groups.html>
- Amazon. (2022c). *AWS Systems Manager Parameter Store*. Retrieved from AWS:
<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html>
- Amazon. (2022d). *Disaster recovery options in the cloud*. Retrieved from AWS:
<https://docs.aws.amazon.com/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-options-in-the-cloud.html#backup-and-restore>
- Amazon. (2022e). *High availability and scalability on AWS*. Retrieved from AWS:
<https://docs.aws.amazon.com/whitepapers/latest/real-time-communication-on-aws/high-availability-and-scalability-on-aws.html>
- Amazon. (2022f). *How Amazon S3 works*. Retrieved from AWS:
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#CoreConcepts>
- Amazon. (2022g). *Launch templates*. Retrieved from AWS:
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/launch-templates.html>
- Amazon. (2022h). *What is an Application Load Balancer?* Retrieved from AWS:
<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>
- Amazon. (2022i). *What is AWS?* Retrieved from AWS: <https://aws.amazon.com/what-is-aws/>
- Amazon Web Services. (2022). *Amazon Virtual Private Cloud User Guide*. Retrieved from Amazon Web Services: <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ug.pdf#what-is-amazon-vpc>
- GeeksforGeeks. (2022). *Difference between IAAS, PAAS and SAAS*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/difference-between-iaas-paas-and-saas/#:~:text=IAAS%20gives%20access%20to%20the,access%20to%20the%20end%20user.&text=It%20is%20a%20service%20model,computing%20resources%20over%20the%20internet.>
- Gillis, A. S. (2020). *Amazon Web Services (AWS)*. Retrieved from Serach AWS:
<https://www.techtarget.com/searchaws/definition/Amazon-Web-Services>

