Building a Highly Available, Scalable Web Application

Graduation Project - AWS Cloud Specialist Track

Prepared By

Hamza Abd-El-Baset El-Gebaly

Supervised By

Eng/ Ahmed Elhamy

Table of Contents

Building a Highly Available, Scalable Web Application

---

1. Overview

The primary objective of this project was to create a scalable, highly available web application on AWS to meet the needs of Example University's admissions department, which experiences peak traffic during admissions periods. The project leveraged various AWS services to ensure availability, load balancing, and security. The goal was to design a fault-tolerant, secure, and scalable infrastructure that could sustain heavy loads while optimizing costs.

Key Objectives:

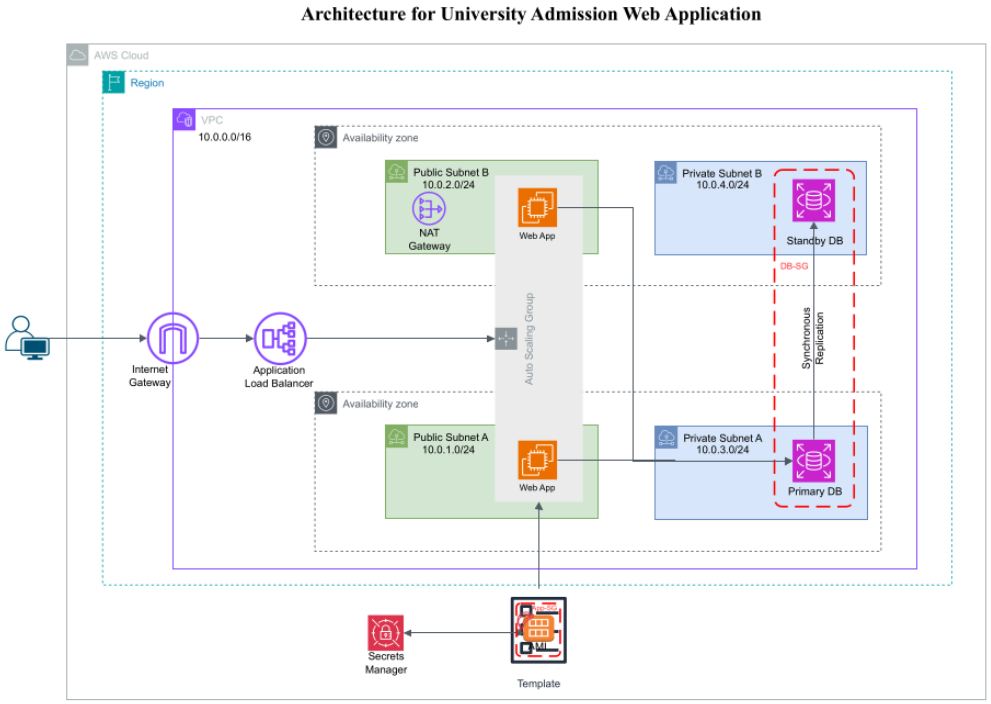
- **High Availability and Scalability**: The application should remain accessible and perform efficiently, even under high load.
- **Security**: Ensuring only authorized access to the database and web servers.
- **Cost Efficiency**: Designing a cost-effective solution by utilizing appropriate AWS services.

2. Implementation Phases

The project was implemented in four phases, following best practices in the AWS Well-Architected Framework.

Phase 1: Planning and Cost Estimation

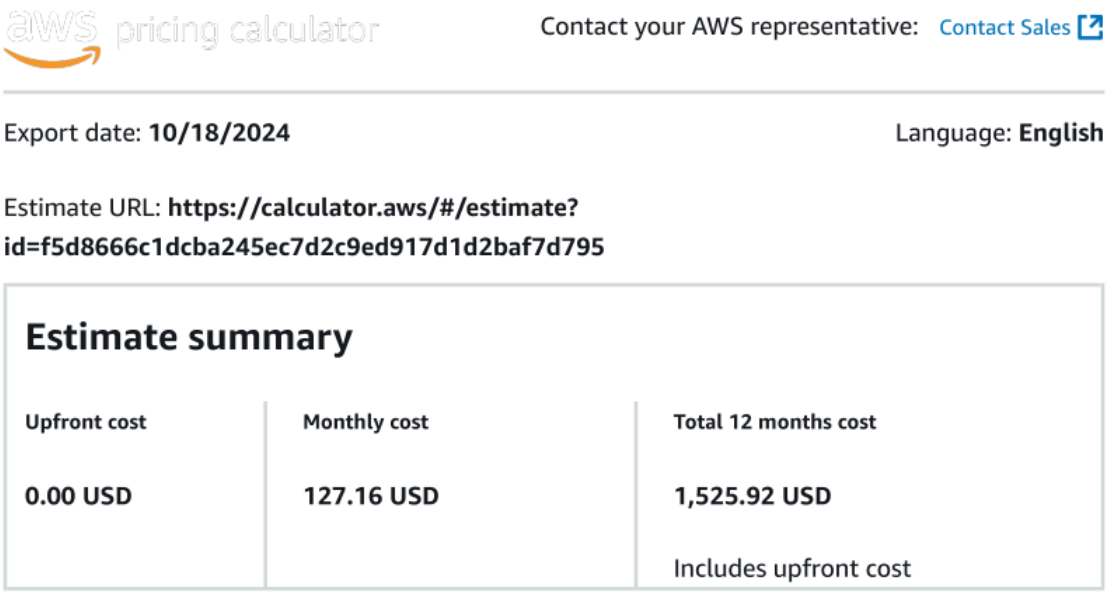
- **Architecture Diagram**:



The architecture included a Virtual Private Cloud (VPC) with two Availability Zones (AZs). Each AZ contained both public and private subnets:

- **Public Subnets**: Deployed for web servers, configured in an Auto Scaling Group to ensure load distribution and availability.
- **Private Subnets**: Used for Amazon RDS, employing a Multi-AZ MySQL deployment for redundancy. The RDS was accessible to the internet only through a NAT gateway in the public subnet.

- **Cost Estimation**:
  Used the AWS Pricing Calculator to estimate costs for 12 months in the us-east-1 region.



Details of the estimate can be found in the following link:
https://calculator.aws/#/estimate?id=f5d8666c1dcba245ec7d2c9ed917d1d2baf7d795

Phase 2: Creating a Functional Web Application

- **Networking**: Created a VPC and set up subnets across two AZs to improve redundancy.

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

(•) VPC only ( ) VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

University-VPC

IPv4 CIDR block Info

(•) IPv4 CIDR manual input

( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

You successfully created vpc-09f4255042dedede7 / University-VPC

VPC > Your VPCs > vpc-09f4255042dedede7

vpc-09f4255042dedede7 / University-VPC

Actions

Details Info

VPC ID vpc-09f4255042dedede7	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0b54f45c9a306deec	Main route table rtb-01df377157beef789	Main network ACL acl-080ee3ff9cd9bc656
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 725984196866	

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

Public Subnet A

The name can be up to 256 characters long.

Availability Zone **Info**

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block **Info**

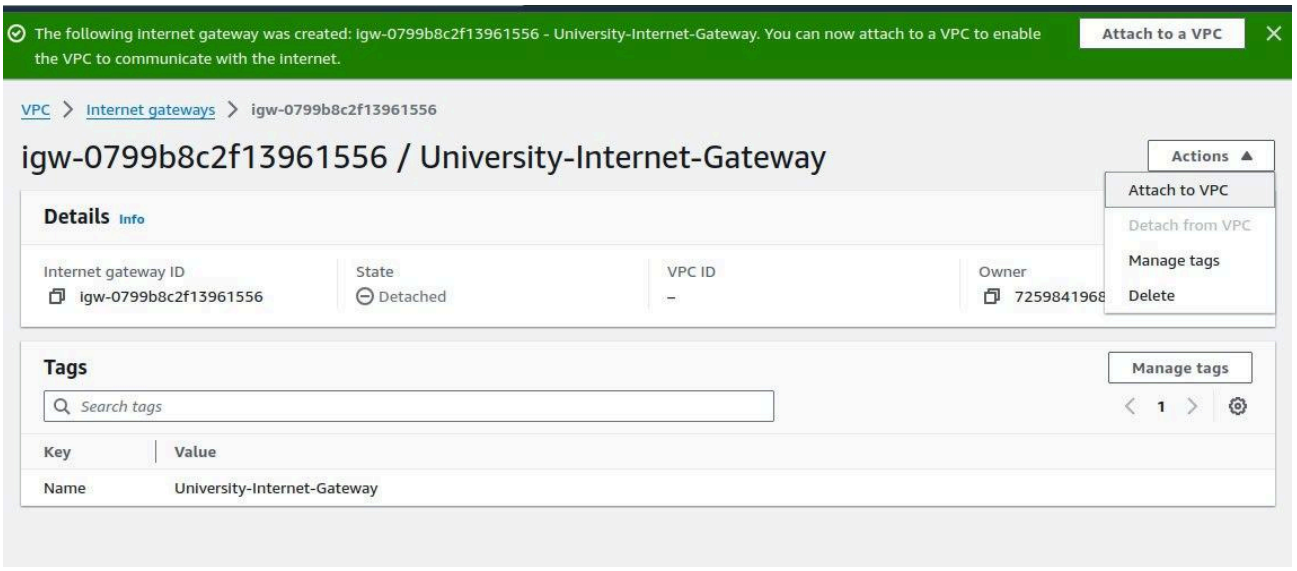
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

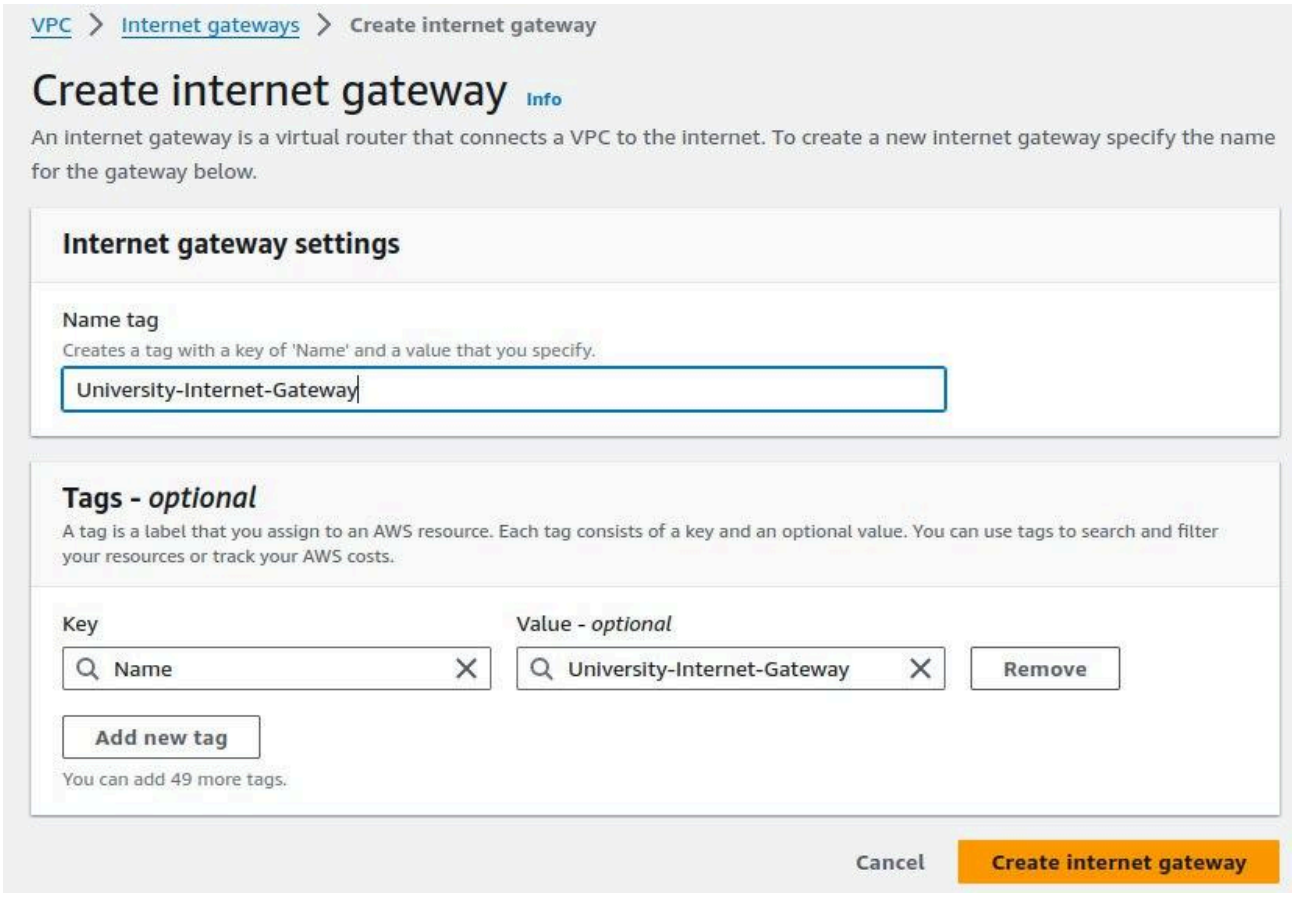
10.0.0.0/16

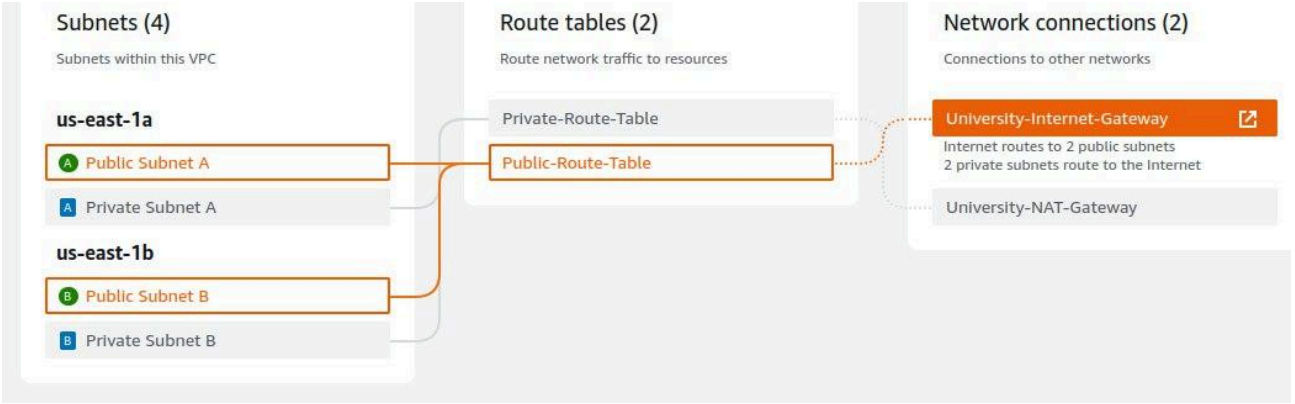
IPv4 subnet CIDR block

10.0.1.0/24 256 IPs

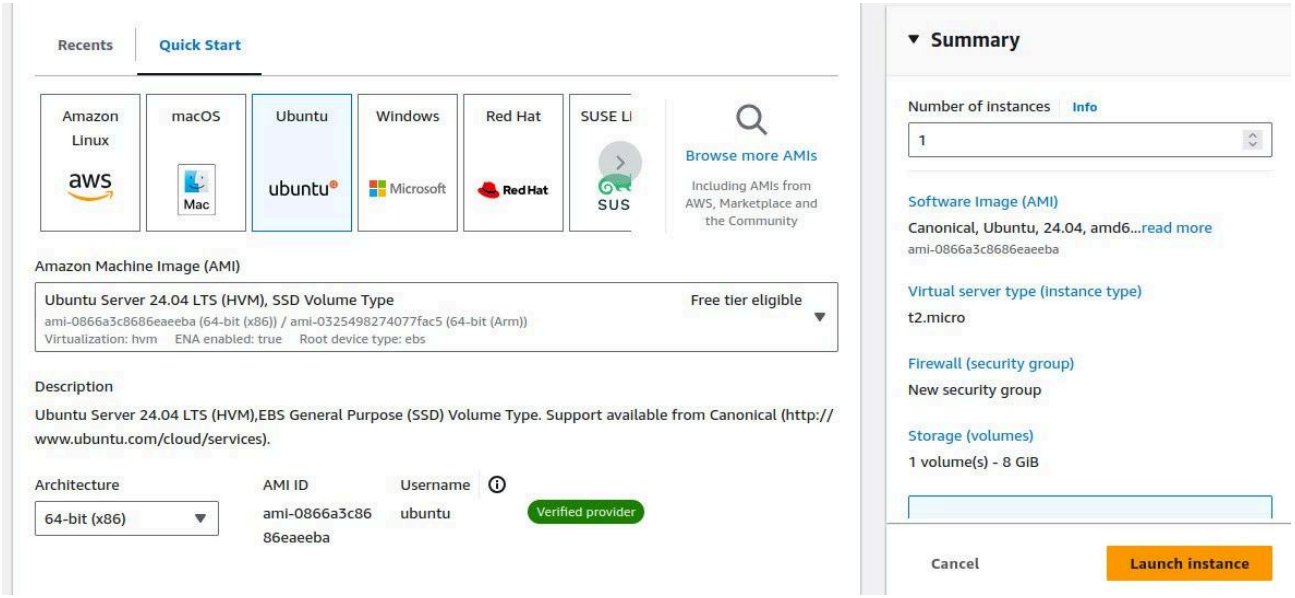
Name	Subnet ID	State	VPC	IPv4 CIDR
Public Subnet A	subnet-07de46381c79e0fe9	Available	vpc-09f4255042dedede7 | Univ...	10.0.1.0/24
Public Subnet B	subnet-0278e74d59f09db70	Available	vpc-09f4255042dedede7 | Univ...	10.0.2.0/24
Private Subnet A	subnet-070966c57fc2e1cc1	Available	vpc-09f4255042dedede7 | Univ...	10.0.3.0/24
Private Subnet B	subnet-0e7dea3d1e92dfaac	Available	vpc-09f4255042dedede7 | Univ...	10.0.4.0/24


The following internet gateway was created: igw-0799b8c2f13961556 - University-Internet-Gateway. You can now attach to a VPC to enable the VPC to communicate with the internet.
Attach to a VPC
VPC > Internet gateways > igw-0799b8c2f13961556
igw-0799b8c2f13961556 / University-Internet-Gateway
Actions
Attach to VPC
Detach from VPC
Manage tags
Delete
Details Info
Internet gateway ID
igw-0799b8c2f13961556
State
Detached
VPC ID
–
Owner
7259841968
Tags
Manage tags
Search tags
1
Key
Value
Name
University-Internet-Gateway

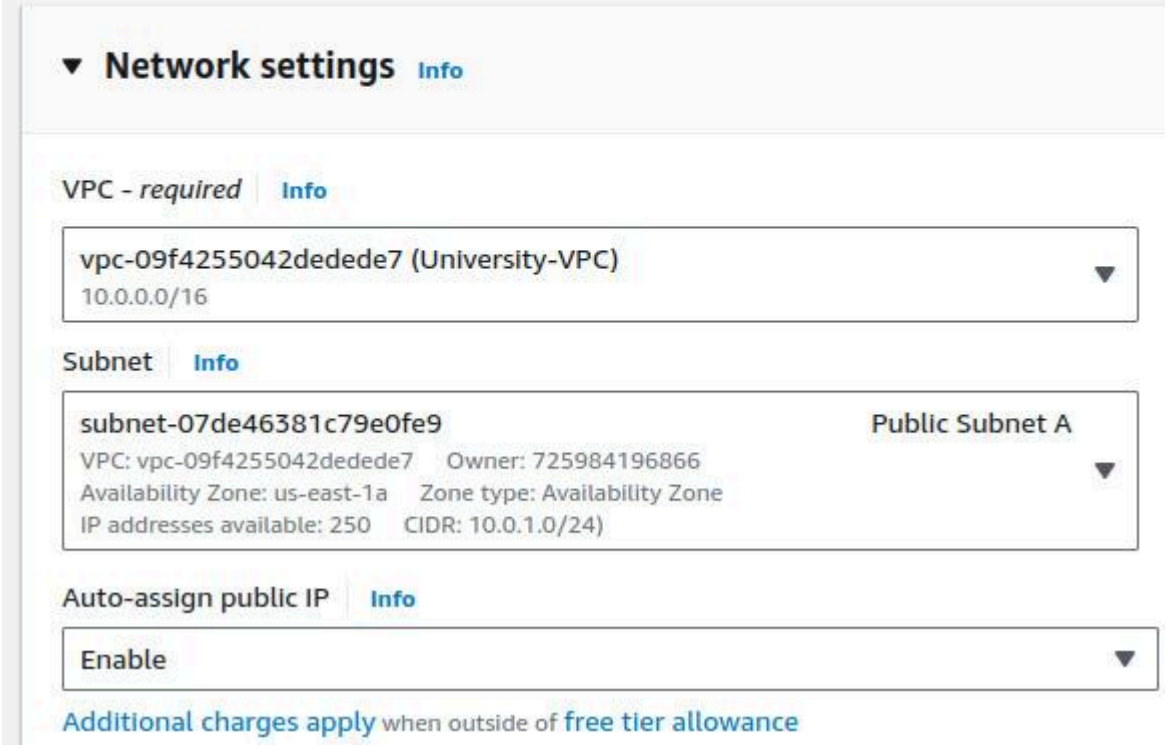

VPC > Internet gateways > Create internet gateway
Create internet gateway Info
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.
Internet gateway settings
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
University-Internet-Gateway
Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Key
Name
Value - optional
University-Internet-Gateway
Remove
Add new tag
You can add 49 more tags.
Cancel
Create internet gateway



- **Web Server**: Deployed an EC2 instance in a public subnet with a web application.


Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0866a3c8686eaeeba (64-bit (x86)) / ami-0325498274077fac5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Architecture
64-bit (x86)
AMI ID
ami-0866a3c8686eaeeba
Username
ubuntu
Verified provider
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0866a3c8686eaeeba
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance


Key pair name - required
vockey


Network settings Info
VPC - required Info
vpc-09f4255042dedede7 (University-VPC)
10.0.0.0/16
Subnet Info
subnet-07de46381c79e0fe9
Public Subnet A
VPC: vpc-09f4255042dedede7 Owner: 725984196866
Availability Zone: us-east-1a Zone type: Availability Zone
IP addresses available: 250 CIDR: 10.0.1.0/24)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance

Security group name - *required*

App-SG

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

Security group for web app EC2 instance

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 41.37.205.68/32) Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Name Info	Description - *optional* Info
My IP	Add CIDR, prefix list or security	e.g. SSH for admin desktop
	41.37.205.68/32 ✕	

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) Remove

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Source type Info	Source Info	Description - *optional* Info
Anywhere	Add CIDR, prefix list or security	e.g. SSH for admin desktop
	0.0.0.0/0 ✕	

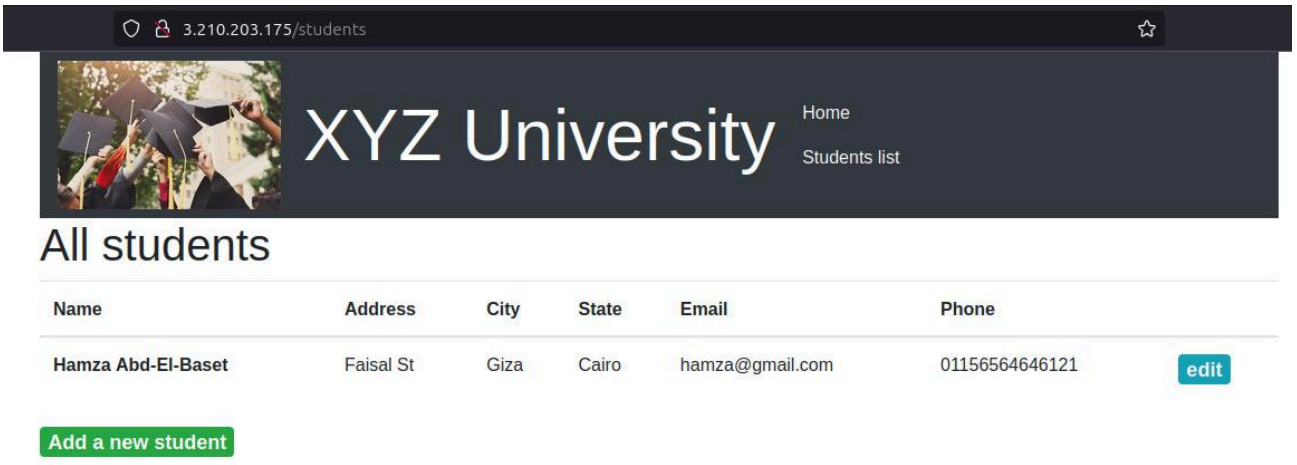
User data - *optional* Info

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash -xe
apt update -y
apt install nodejs unzip wget npm mysql-server -y
#wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCAP1-1-DEV/code.zip -P /home/ubuntu
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCAP1-1-91571/1-lab-capstone-project-1/code.zip -P /home/ubuntu
cd /home/ubuntu
unzip code.zip -x "resources/codebase_partner/node_modules/*"
cd resources/codebase_partner
npm install aws aws-sdk
mysql -u root -e "CREATE USER 'nodeapp' IDENTIFIED WITH mysql_native_password BY 'student12'";
mysql -u root -e "GRANT all privileges on *.* to 'nodeapp'@'%';"
mysql -u root -e "CREATE DATABASE STUDENTS;"
```

- **Testing**: Verified functionality by performing basic application tasks (view, add, delete, and modify student records).



Phase 3: Decoupling Application Components

- **Database Setup**:
  Provisioned an Amazon RDS MySQL database with Single-AZ as the project limitations prevented Multi-AZ deployment.

- **Development Environment Configuration**:
  Configured a development environment on AWS Cloud9 for ease of creating secrets and database migration.

AWS Cloud9 > Environments > Create environment

Create environment Info

Details

Name

Web-App-Env

Limit of 60 characters, alphanumeric, and unique per user.

Description - *optional*

Limit 200 characters.

Environment type Info

Determines what the Cloud9 IDE will run on.

- (•) New EC2 instance
  Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.
- ( ) Existing compute
  You have an existing instance or server that you'd like to use.

New EC2 instance

Instance type Info

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

- (•) t2.micro (1 GiB RAM + 1 vCPU)
  Free-tier eligible. Ideal for educational users and exploration.
- ( ) t3.small (2 GiB RAM + 2 vCPU)
  Recommended for small web projects.
- ( ) m5.large (8 GiB RAM + 2 vCPU)
  Recommended for production and most general-purpose development.
- ( ) Additional instance types
  Explore additional instances to fit your need.

Platform Info

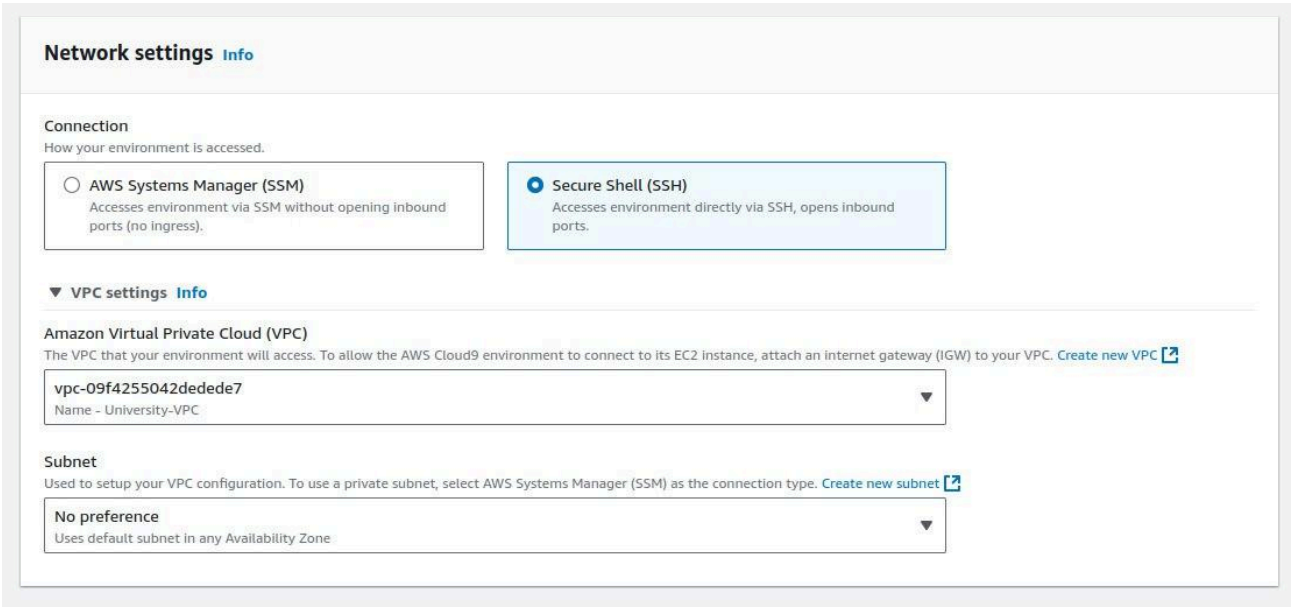
This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2023 ▼

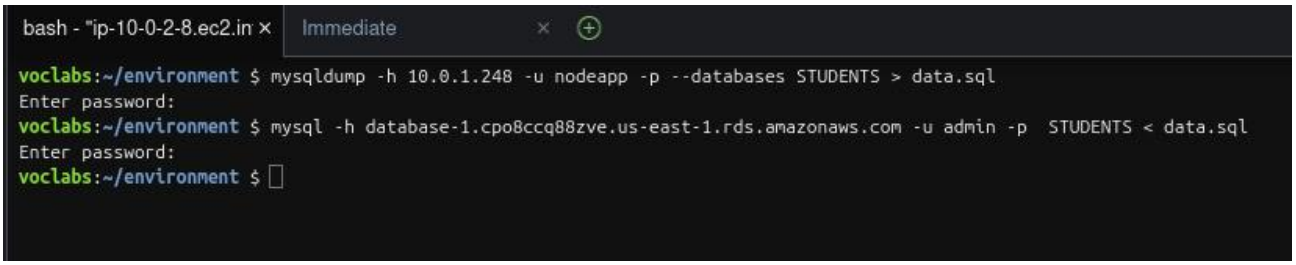
Timeout

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

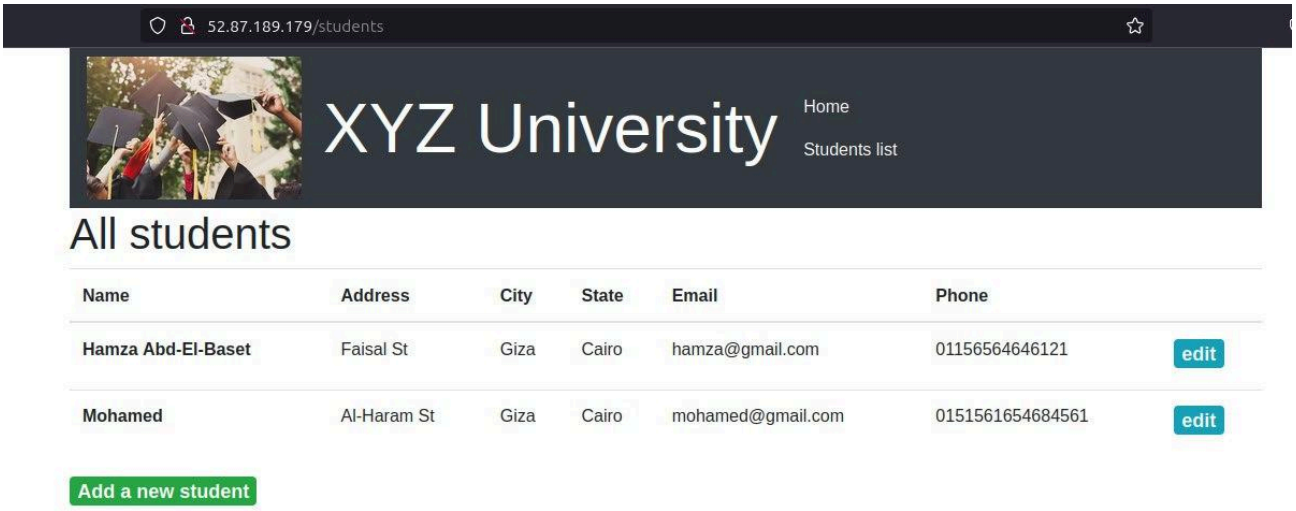
30 minutes ▼



- **Secrets Management**:
  Configured AWS Secrets Manager to securely manage database credentials.

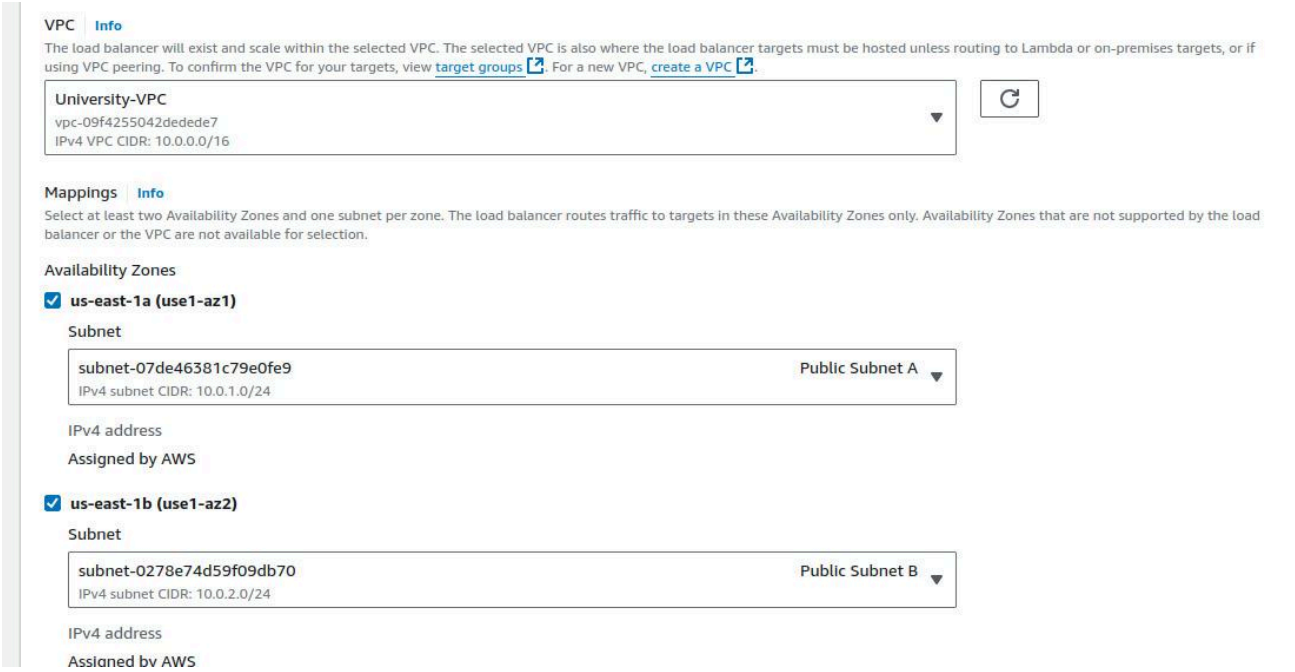


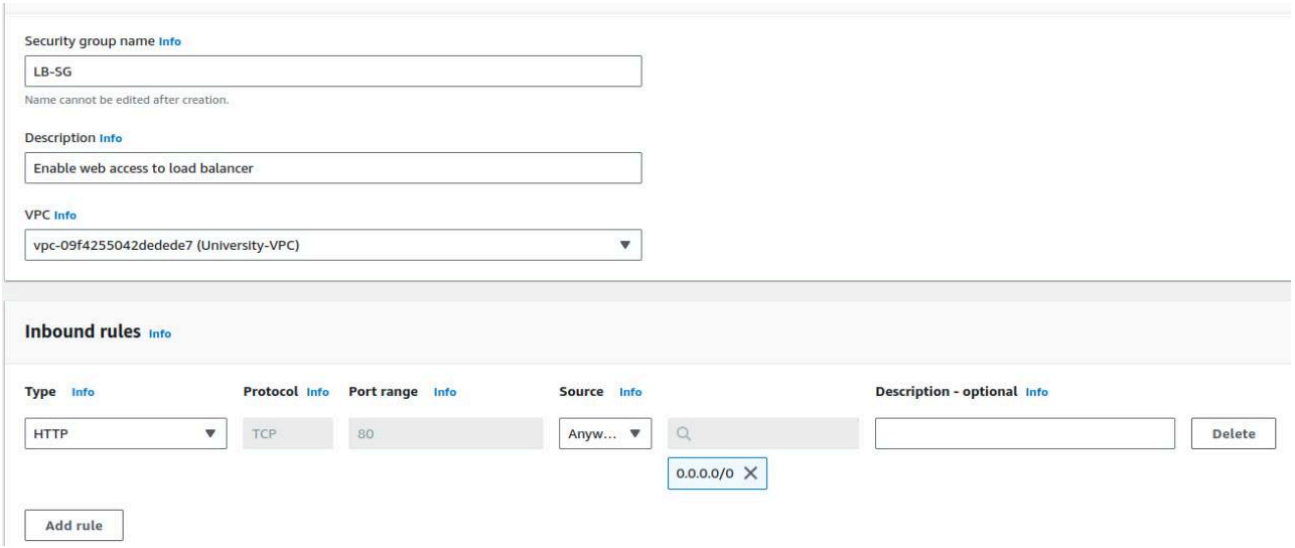
- **Testing after decoupling**:
  Verified functionality by performing basic application tasks (view, add, delete, and modify student records).

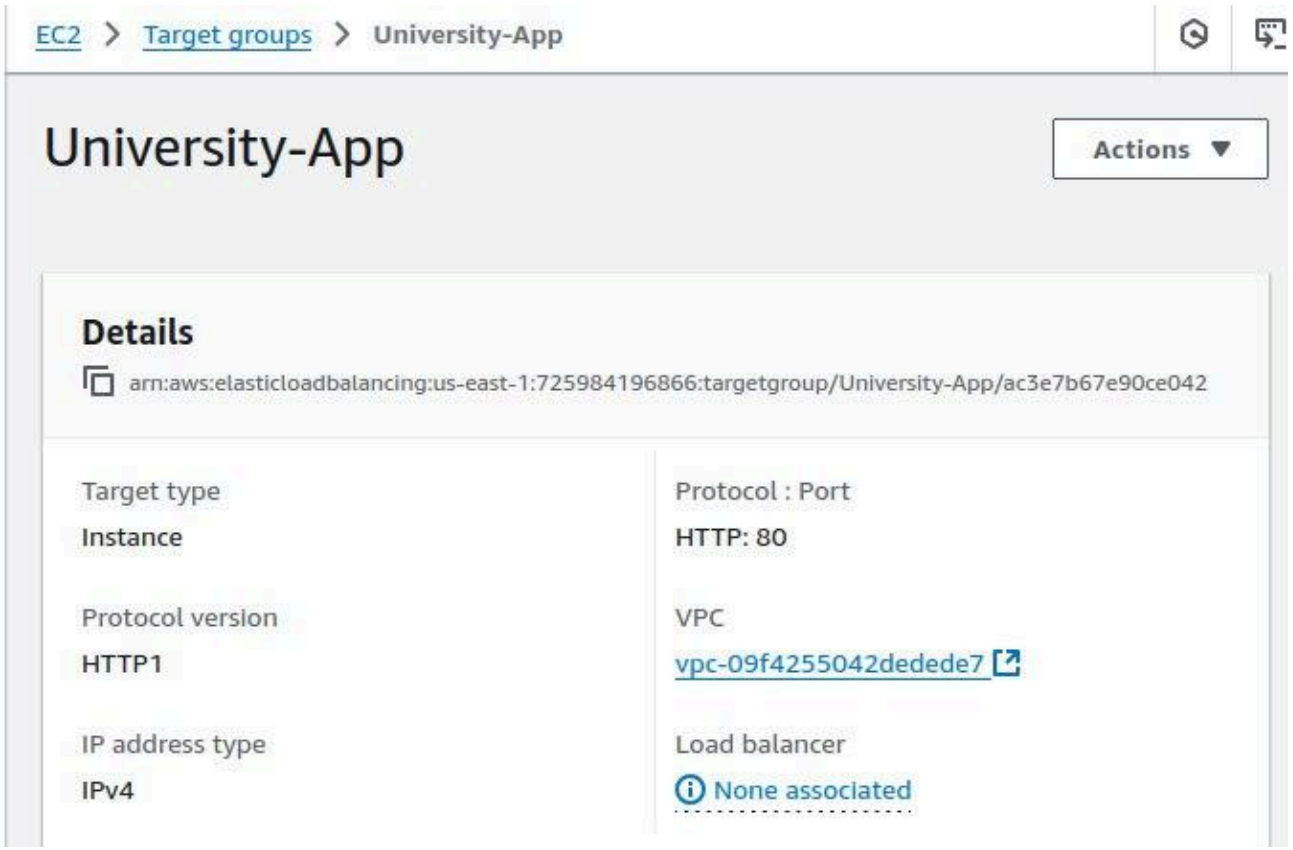


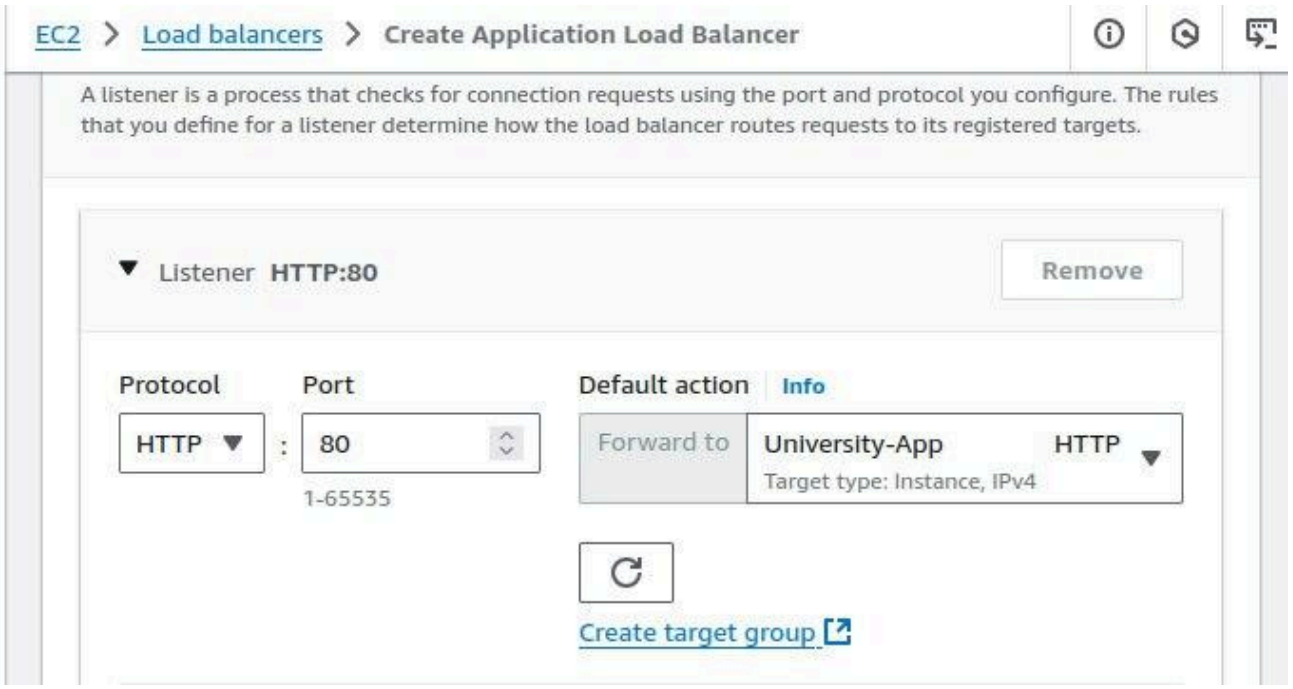
Phase 4: Enhancing High Availability and Scalability

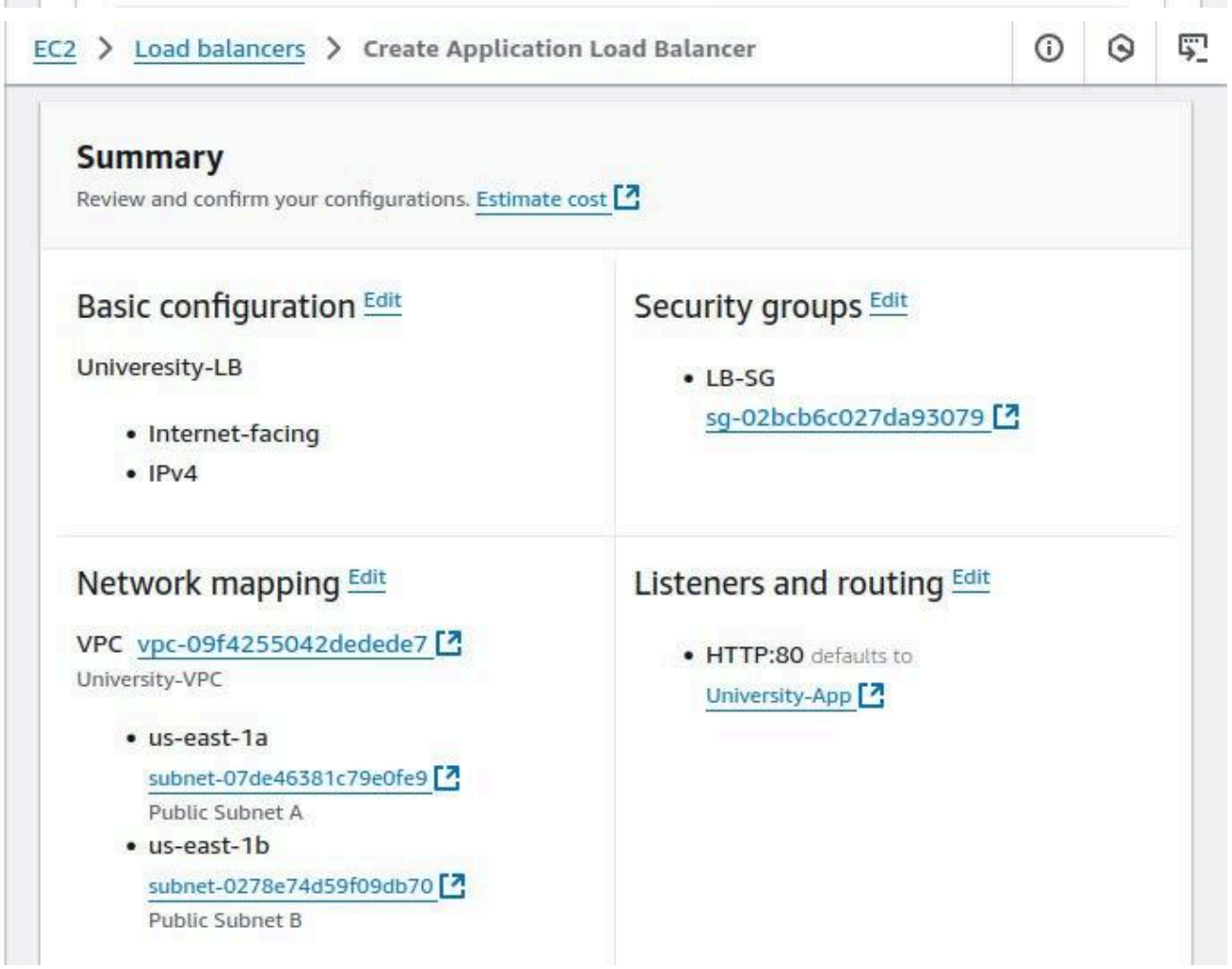
- **Load Balancing**:
  Deployed an Application Load Balancer (ALB) to manage incoming traffic across multiple web servers.






EC2 > Target groups > University-App
University-App
Actions
Details
arn:aws:elasticloadbalancing:us-east-1:725984196866:targetgroup/University-App/ac3e7b67e90ce042
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-09f4255042dedede7
IP address type
IPv4
Load balancer
None associated


EC2 > Load balancers > Create Application Load Balancer
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.
Listener HTTP:80
Remove
Protocol
Port
Default action
Info
HTTP
80
1-65535
Forward to
University-App
HTTP
Target type: Instance, IPv4
Create target group


EC2 > Load balancers > Create Application Load Balancer
Summary
Review and confirm your configurations. Estimate cost
Basic configuration Edit
Univeresity-LB
Internet-facing
IPv4
Security groups Edit
LB-SG
sg-02bcb6c027da93079
Network mapping Edit
VPC vpc-09f4255042dedede7
University-VPC
us-east-1a
subnet-07de46381c79e0fe9
Public Subnet A
us-east-1b
subnet-0278e74d59f09db70
Public Subnet B
Listeners and routing Edit
HTTP:80 defaults to
University-App

- **Auto Scaling**:
  Configured an Auto Scaling Group (ASG) for the web server instances, enabling scaling between 2 to 4 instances as demand fluctuates.
- **Load Testing**:
  Conducted load tests to ensure scaling worked as expected under heavy loads, verifying system performance and availability.

3. Conclusion

This AWS project achieved the intended objectives of building a scalable, highly available, and secure infrastructure for the admissions web application. The solution provides:

- **Enhanced User Experience**:
  Reliable access and quick responses to user requests, even during peak times.
- **Cost Optimization**:
  The use of Auto Scaling and on-demand resources helps balance performance with cost efficiency.
- **Security**:
  Network segmentation, private database access, and AWS Secrets Manager for credential storage improve overall security.

This architecture can be further optimized for production environments by introducing HTTPS access.