TARUMT
TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Faculty of Computing and Information Technology

BUILDING A HIGHLY AVAILABLE, SCALABLE WEB APPLICATION

RDS3S3 & RDS3S2

Practical Assignment

BMIT3273 Cloud Computing (202509)

LECTURER: LOW CHOON KEAT

Name	Registration No.
Xavier Ngow Kar Yuen	24WMR12398
Heng Zheng Teck	22WMR11422
Cho Wei Bin	22WMR04086

Building a Highly Available, Scalable Web Application

Grading Rubric

You can use the following rubric to assess students' solutions.

Criteria	Uns$t&sf$(t)*+	Needs Improvement	Good	Excellent
Functional	The web server wasn't accessible from the internet.	The web server was accessible from the internet, but the application webpage didn't display.	The application was accessible from the internet, and all operational tasks (querying, adding, and removing records) could be performed successfully. Migrated data wasn't available.	The application was accessible from the internet, and all operational tasks (querying, adding, and removing records) could be performed successfully. Migrated data was available.
Load Balanced	Load balancing wasn't implemented.	Load balancing was implemented partially. Application traffic was directed to one Availability Zone instead of multiple Availability Zones.	The application traffic was distributed across multiple Availability Zones, but instances were carrying variable load.	The application traffic was distributed evenly across two Availability Zones and across the instances in those Availability Zones. All instances were evenly loaded at peak user load.
Scalable	Automatic scaling wasn't implemented at the application layer.	Automatic scaling was implemented at the application layer. However, scaling was delayed or didn't occur.	Automatic scaling was implemented at the application layer. However, the automatic scaling configuration could be improved by adjusting the scaling policy and the minimum and maximum capacities.	The automatic scaling configuration was optimal, staying within the scaling range, without reaching the maximum limit frequently. The application layer scaled in and out according to application demand and in a timely manner.

Criteria	Unsatisfactory	Needs Improvement	Good	Excellent
Highly Available	A single web server instance hosted the application and database. There was only one public subnet. Managed services, such as Elastic Load Balancing (ELB) and automatic scaling, weren't implemented.	Multiple web server instances were in one Availability Zone, and the application wasn't architected to be highly available.	Web server instances were spread across two Availability Zones. The architecture had public and private subnets in each Availability Zone. Managed services, such as AWS Secrets Manager and ELB, were used.	Web server instances were spread across two Availability Zones. The architecture had public and private subnets in each Availability Zone. Managed services, such as Secrets Manager and ELB, were used. Database backup was enabled for recovery.
Secure	The solution didn't consider any security best practices.	The solution didn't consider any security best practices explicitly, except for using Secrets Manager, as provided in the guidance.	The solution considered security best practices for only a few resources: • Amazon Elastic Compute Cloud (Amazon EC2) instances were locked down to port 80, but port 22 was accessible from anywhere. • The DB instance was in a private subnet, and the web server could access the DB. However, the DB security group wasn't restricted to database port 3306.	The solution followed security best practices: • All EC2 instances were locked down to port 80 and port 22. • Access to EC2 instances through port 22 was limited to an IP range that AWS provided. • The DB instance was in a private subnet. • The web server instances could only access the DB on port 3306. • The DB credentials were stored securely in Secrets Manager. • The load balancer was protected, with security groups having access only on port 80.
Cost Optimized	The Auto Scaling group size and instance sizing weren't considered explicitly, and only default sizing was used.	The Auto Scaling group size wasn't considered; for example, the minimum, maximum, and desired configurations were kept the same.	EC2 instances were oversized or undersized. The Auto Scaling group minimum configuration was high. The DB instance was oversized or undersized.	All EC2 instances were properly sized to handle the traffic. The Auto Scaling group minimum configuration supported the minimum expected traffic. The DB instance was optimally sized to handle normal and variable user load.

Criteria	Unsatisfactory	Needs Improvement	Good	Excellent
High Performing	The application was accessible intermittently under a normal user load. The application failed to serve users (timed out) for a higher user load.	The application was accessible and functional under a normal user load. However, the application failed to serve users (timed out) for a higher user load.	The application was accessible and functional under a normal user load and a higher user load without any deterioration in response time. However, response time deteriorated during the maximum load.	The application was accessible and functional under a normal user load and a maximum load without any deterioration in response. All operations (query, adding, and removing records) finished within the expected time.

Demonstration Session and Student Effort (30 marks)

Student Name	***Score***	***Total***

1. Xavier Ngow Kar Yuen
2. Heng Zheng Teck
3. Cho Wei Bin

Member Profiles

Name (Block Capital)	Contact Number	Profile Photo
Cho Wei Bin	0166639197	
HENG ZHENG TECK	01115152095	
Xavier Ngow Kar Yuen	0173466118	

Note:

- Each member must attach a passport-size photo (4 cm × 4 cm) with a plain background.

- Place this page after the cover page and before the Acknowledgement section.

- The lecturer will use this page for attendance verification and grading reference during presentation.

-