



## **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	Unsatisfactory	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
<b>Highly Available</b>	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.
<b>Secure</b>	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: <ul style="list-style-type: none"> <li>Amazon Elastic Compute Cloud (Amazon EC2) instances were locked down to port 80, but port 22 was accessible from anywhere.</li> <li>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.</li> </ul>	The solution followed security best practices: <ul style="list-style-type: none"> <li>All EC2 instances were locked down to port 80 and port 22.</li> <li>Access to EC2 instances through port 22 was limited to an IP range that AWS provided.</li> <li>The DB instance was in a private subnet.</li> <li>The web server instances could only access the DB on port 3306.</li> <li>The DB credentials were stored securely in Secrets Manager.</li> <li>The load balancer was protected, with security groups having access only on port 80.</li> </ul>
<b>Cost Optimized</b>	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
<b>High Performing</b>	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)**

<b>Student Name</b>	<b>Score</b>	<b>Total</b>
---------------------	--------------	--------------

- 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.





# Academy Lab Projects – Showcase

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

**Building a Highly Available, Scalable Web Application**

21 December 2025

# Table of contents

---

No.	Contents	Page
1.	Business Scenario Overview	4 - 8
2.	Architecture Diagram	10
	<b>Solution Overview</b>	
3.	Performance & Functionality	15 - 25
4.	Scalability	26 - 27
5.	High Availability	28 - 34
6.	Load Balancing	35 - 38
7.	Security	39 - 43
8.	Monitoring and Threat Detection	44 - 45
9.	Cost Optimization	46 - 52

# Table of contents

---

Lesson Learned	
16.	Challenges
17.	Helpful Resource
18.	New Skills
19.	Nest Steps
Individual Lab Scores	
20.	Guided and Challenge Lab Scores

# Business scenario overview

---

- **Problem Statement:**

Example University is experiencing performance and availability issues with their on-premises student records web application. During peak admission periods, when thousands of students are accessing the system simultaneously, the application often becomes slow or even entirely unavailable. This results in a poor user experience, with students facing long loading times or even being unable to access their records. Additionally, the admissions team is overwhelmed by complaints about the application's instability during these critical periods.

- **Opportunity Statement:**

As the university grows, this issue will only worsen unless action is taken. This presents an opportunity to modernize the university's IT infrastructure by migrating the student records application to the cloud. By leveraging cloud services, the university can improve the performance, scalability, availability, and security of the application, thereby improving the experience for both students and staff.

# Business scenario overview

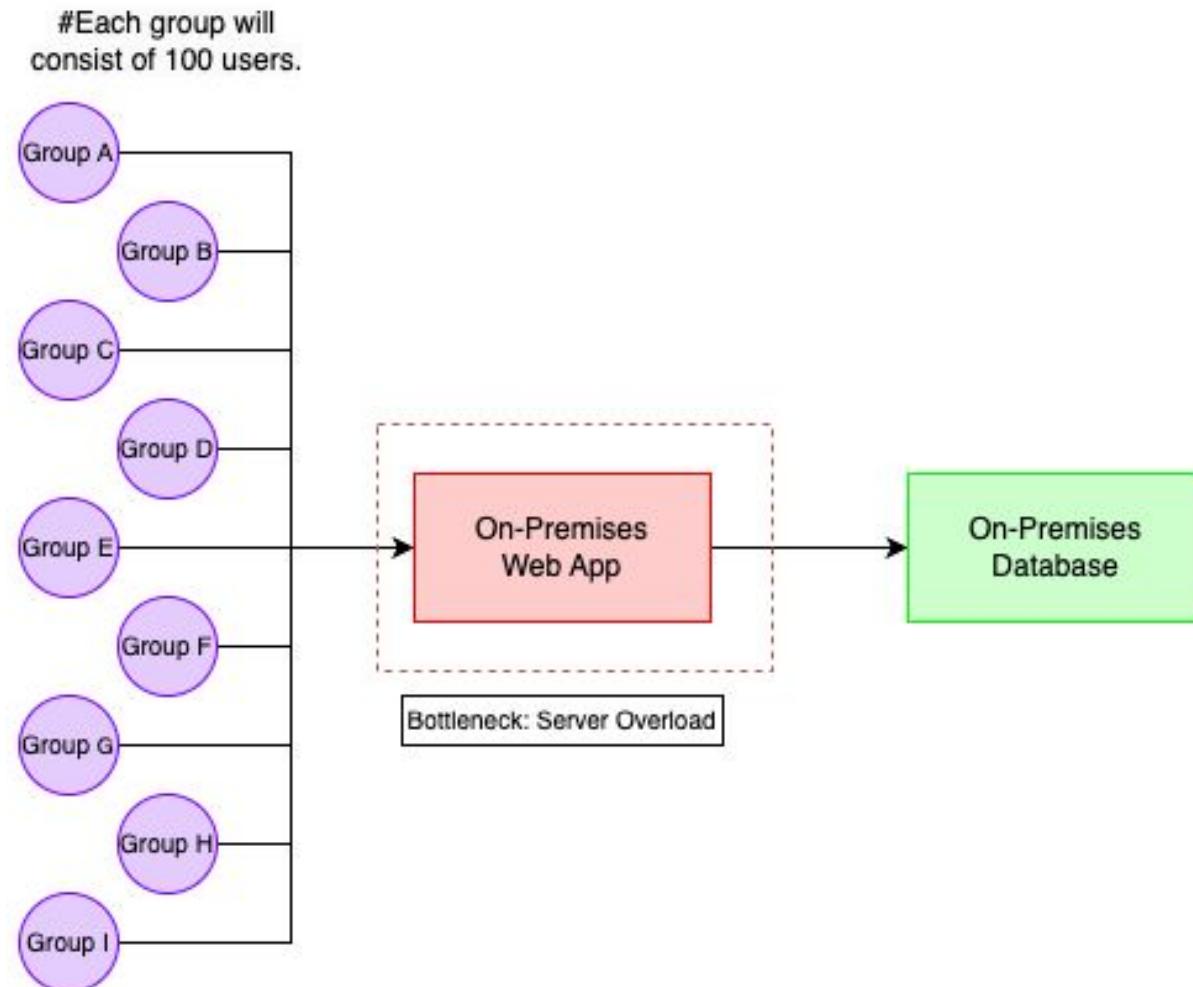


Figure 1: On-Premises Bottleneck During Peak Admissions

# Business scenario overview

---

- **Solution requirements:**

- 1. Performance & Functionality:**

- The system must be able to handle thousands of simultaneous users with no perceivable delay, even during peak admission periods. Users should be able to perform tasks such as viewing, adding, deleting, and modifying student records efficiently.

- 2. Scalability:**

- The new solution must automatically scale up or down based on traffic demand. During peak periods, it should quickly scale up to handle additional traffic and scale down during off-peak times to optimize costs.

- 3. High Availability:**

- The system must be architected for high availability, meaning that even if one server or data center becomes unavailable, the application remains accessible without downtime. This requires distributing resources across multiple Availability Zones (AZs).

# Business scenario overview

---

- Solution requirements (Cont):

## 4. Load Balancing:

- The application should use an Elastic Load Balancer (ELB) to distribute incoming traffic across multiple servers to ensure no single server becomes overloaded.

## 5. Security:

- The system must be secured against unauthorized access. Sensitive data, including student records, must be protected both in transit and at rest.
- The database should be located in a private subnet and not accessible directly from the public internet.
- Access to the database should be controlled using AWS Secrets Manager to store and manage credentials securely.

# Business scenario overview

---

- Solution requirements (Cont):

## **6. Monitoring and Threat Detection:**

- The new system should include monitoring and logging capabilities to track performance, resource usage, and security events.
- AWS GuardDuty should be implemented to detect and alert the team to any suspicious activity, ensuring the system remains secure.

## **7. Cost Optimization:**

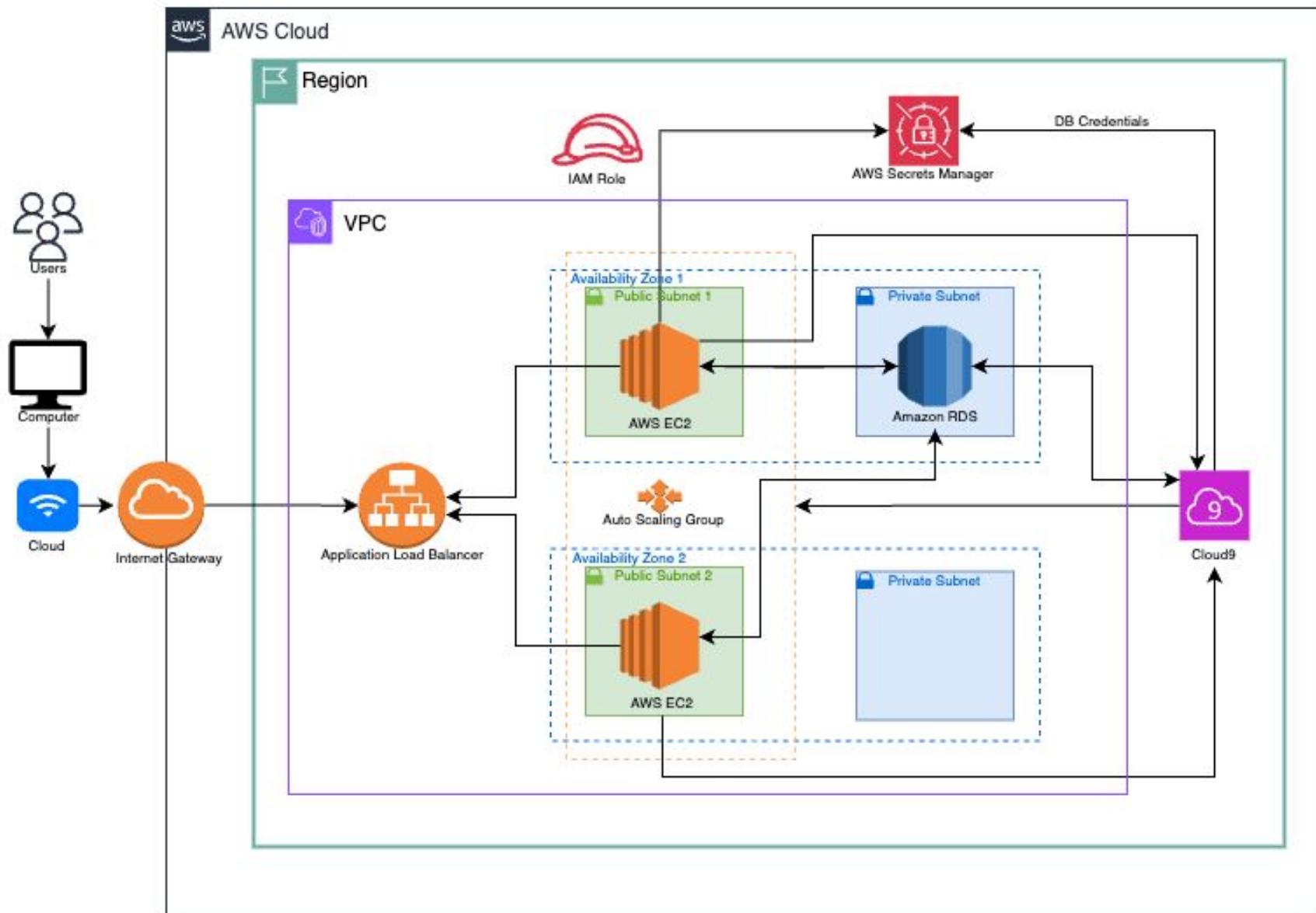
- The solution must be designed to minimize costs, with resources being allocated dynamically based on actual demand.
- Auto Scaling should be used to ensure that only the necessary number of servers are running at any given time.



# Solution Overview

# Architecture diagram of the solution

	- Distribute incoming traffic across multiple targets
	- Server to host website
	- Control access to AWS resources
	- Save database credentials
	- Managed relational database service
	- Dynamically scale in/out instances
	- Cloud-based coding environment
	- Enable communication between VPC and Internet



# Solution overview

---

## • High-Level Description

The proposed solution is a cloud-based student records management system deployed entirely on Amazon Web Services(AWS). The system hosts a CRUD (Create,Read,Update,Delete) web application on EC2 instances, which are automatically managed by an Auto Scaling Group and distributed across two Availability Zones using an Application Load Balancer. Student records are stored securely in an Amazon RDS MySQL database placed in a private subnet prevent unauthorized external access. AWS Secrets Manager securely manages database credentials, while Cloud9 provides a development and testing environment for deploying the application and running load tests.

This design ensures the system is scalable, secure, and highly available, with optimized costs for a Proof of Concept (POC) deployment.

# Solution overview

---

- Design Considerations

1. Scalability – The system uses Auto Scaling to dynamically add or remove EC2 instances based on traffic, ensuring seamless performance during peak admission periods.
2. High Availability – By distributing resources across two Availability Zones, the solution minimizes down time and improves fault tolerance.
3. Security – Student records are secured through a private subnet database, encrypted data transfers, and credential management via AWS Secrets Manager. Only authorized EC2 instances can access the database.
4. Cost Optimization – Light weight instance types (t3.micro) and the Dev/Test RDS template are used to minimize costs, while scaling ensures resources are provisioned only when necessary.
5. Monitoring – Cloud9 load tests allow continuous performance monitoring, ensuring the system meets functional and non-functional requirements.

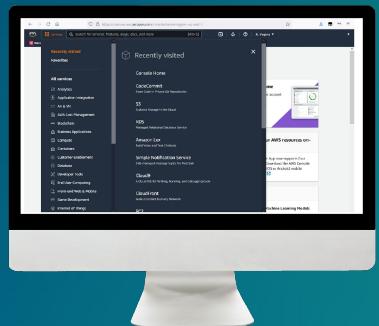
# Solution overview

---

- Use cases

- Students can view, add, delete, and modify their records seamlessly, even during peak traffic.
- Administrators can manage student records efficiently, with guaranteed uptime and performance.
- The system automatically scales to accommodate high traffic during admissions and reduces resource usage in off-peak times, optimizing costs.

# Demo



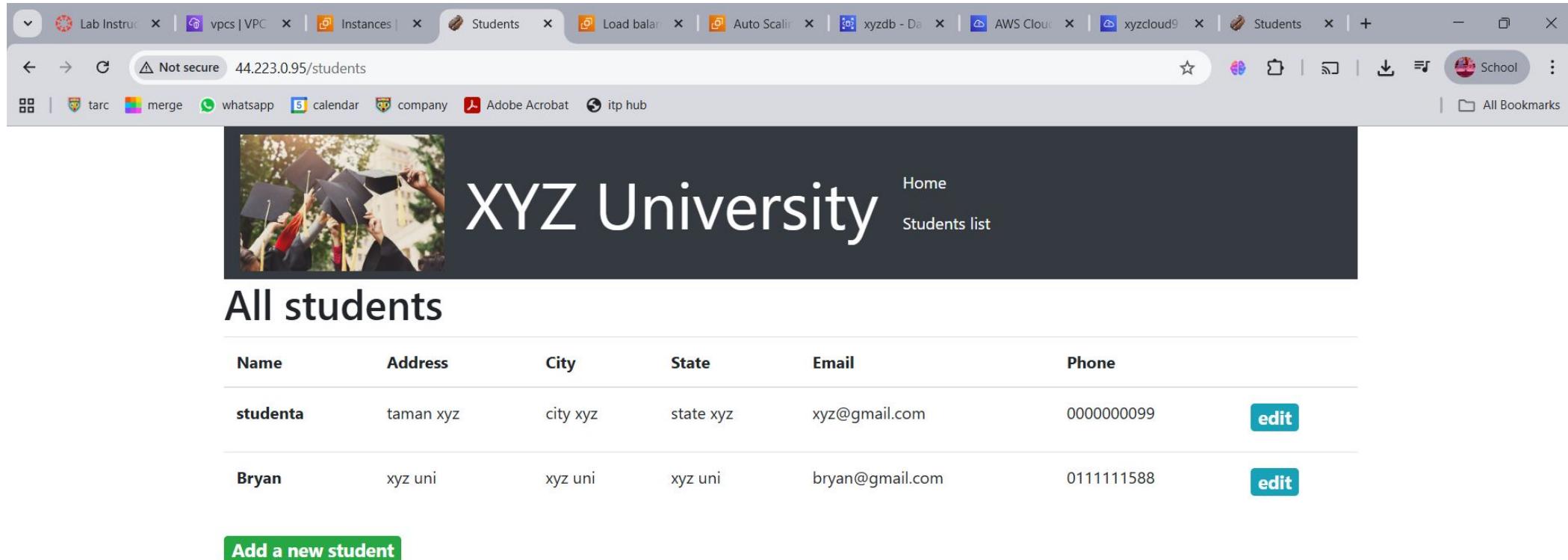
- Cho Wei Bin Presentation :  
[https://drive.google.com/file/d/1WjnMj7DgONU0FPbL\\_kHagzLvUpobqs-x/view?usp=sharing](https://drive.google.com/file/d/1WjnMj7DgONU0FPbL_kHagzLvUpobqs-x/view?usp=sharing)
- HENG ZHENG TECK  
<https://drive.google.com/file/d/1Yd9jj92MfseMGhcqXQIUlgOSp3u67ebc/view?usp=sharing>
- Xavier Ngow Kar Yuen  
<https://drive.google.com/file/d/1cPffbhl-MdQba509VD3DpgxJmezGD5ip/view?usp=sharing>

# Solution Overview:

## Performance & Functionality

# Performance & Functionality

## Operational tasks: adding record (part 1)



The screenshot shows a web browser window with multiple tabs open, including 'Students' and 'xyzcloud9'. The main content is the 'XYZ University' website, specifically the 'Students list' page. The page features a dark header with the university's name and navigation links for 'Home' and 'Students list'. Below the header is a photograph of graduates. The main area displays a table of student records:

Name	Address	City	State	Email	Phone
studenta	taman xyz	city xyz	state xyz	xyz@gmail.com	0000000099
Bryan	xyz uni	xyz uni	xyz uni	bryan@gmail.com	0111111588

At the bottom left, there is a green button labeled 'Add a new student'.

# Performance & Functionality

## Operational tasks: adding record (part 2)

Adding a new Students which is 'kfc' and the details

All fields are required

Name  
kfc

Address  
taman uni xyz

City  
xyz city

State  
uni state

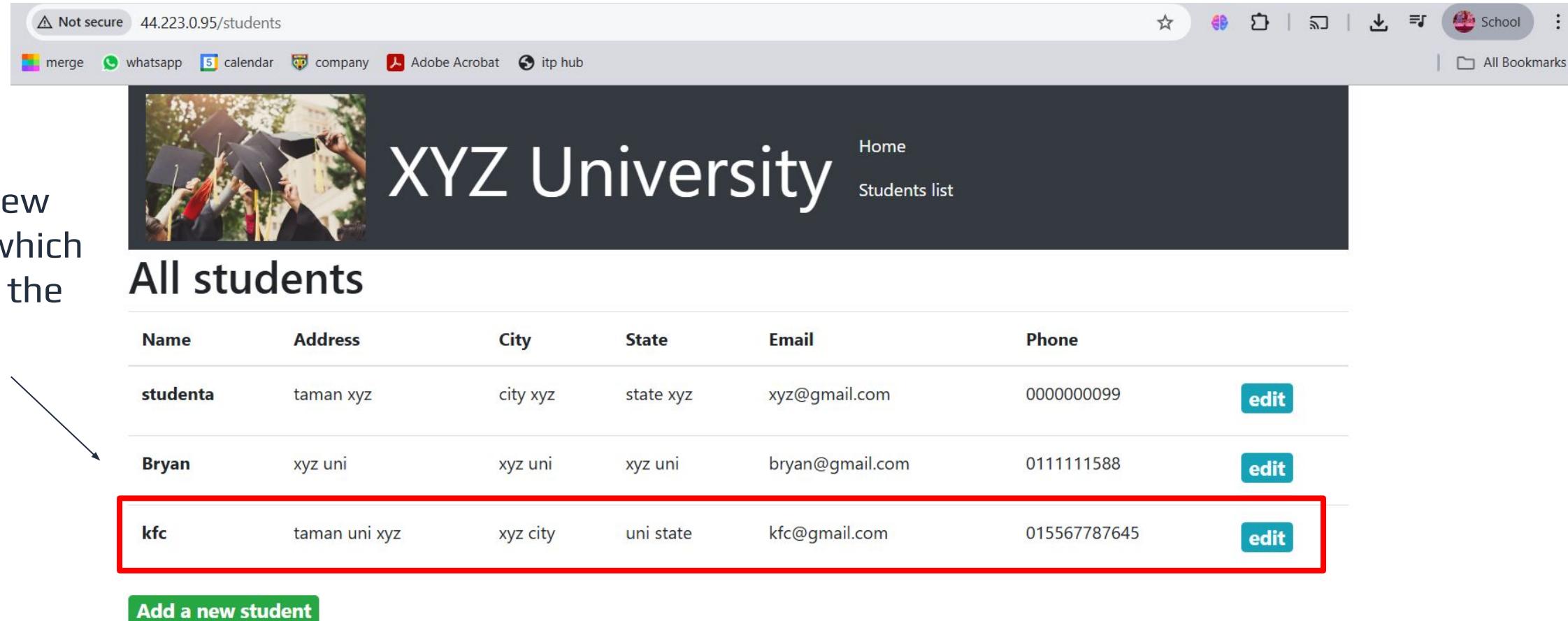
Email  
kfc@gmail.com

A screenshot of a web browser showing a form for adding a new student record. The browser has multiple tabs open, including 'Students', 'Load balanc', 'Auto Scal', 'xyzdb - Da', 'AWS Cloud', 'xyzcloud9', and another 'Students' tab. The main content area shows a header for 'XYZ University' with links to 'Home' and 'Students list'. Below the header is a banner image of graduates. The form itself has a light blue header bar stating 'All fields are required'. It contains five input fields: 'Name' (containing 'kfc'), 'Address' (containing 'taman uni xyz'), 'City' (containing 'xyz city'), 'State' (containing 'uni state'), and 'Email' (containing 'kfc@gmail.com'). A red arrow points from the text 'Adding a new Students which is 'kfc' and the details' to the 'Name' input field.

# Performance & Functionality

## Operational tasks: adding record (part 3)

Adding a new Students which is 'kfc' and the details



The screenshot shows a web browser window with the URL `44.223.0.95/students`. The page title is "XYZ University". The main content area is titled "All students" and contains a table with the following data:

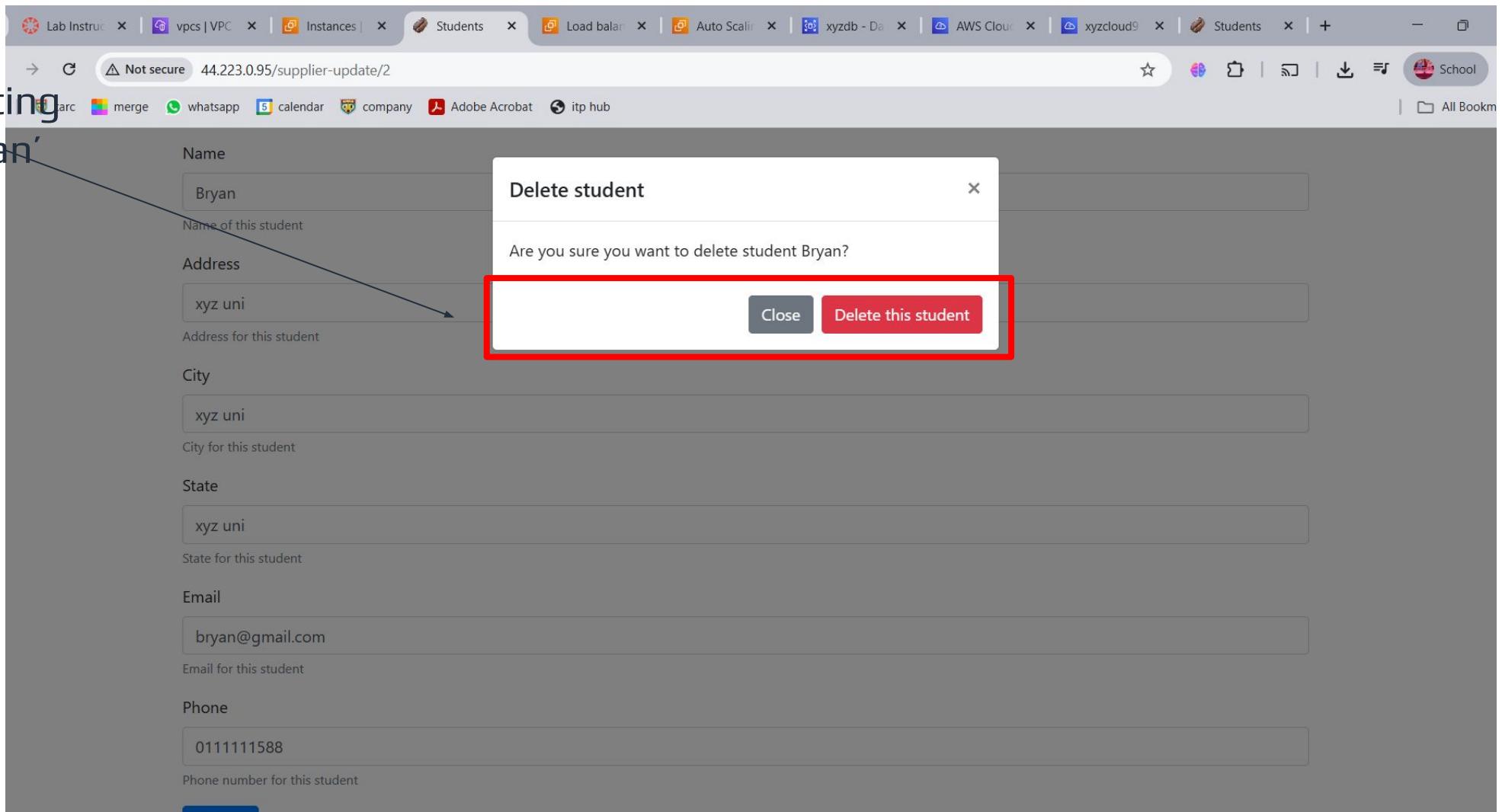
Name	Address	City	State	Email	Phone	Action
studenta	taman xyz	city xyz	state xyz	xyz@gmail.com	0000000099	<a href="#">edit</a>
Bryan	xyz uni	xyz uni	xyz uni	bryan@gmail.com	0111111588	<a href="#">edit</a>
kfc	taman uni xyz	xyz city	uni state	kfc@gmail.com	015567787645	<a href="#">edit</a>

[Add a new student](#)

# Performance & Functionality

## Operational tasks: removing record (part 1)

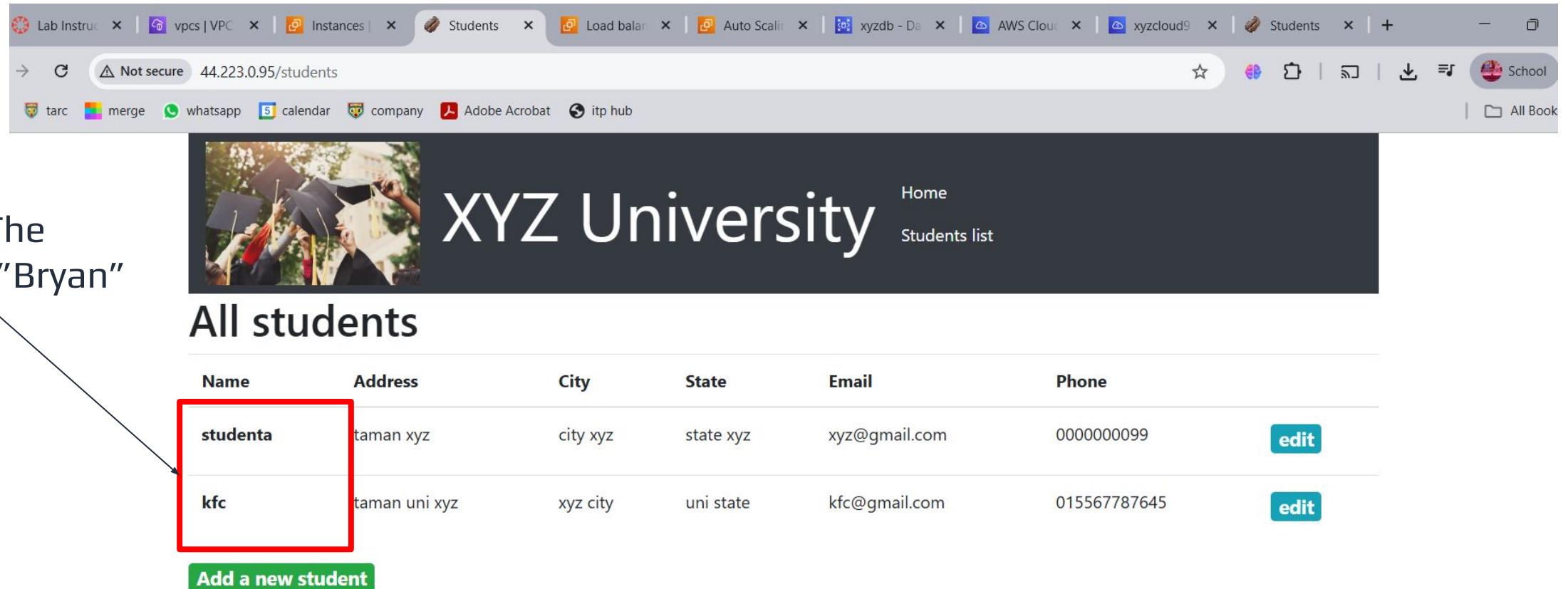
Deleting Existing Student, 'Bryan'



# Performance & Functionality

## Operational tasks: removing record (part 2)

Deleted The Student, "Bryan"



The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "Students" and displays the URL "44.223.0.95/students". The page content is for "XYZ University" and shows a heading "All students". Below the heading is a table with columns: Name, Address, City, State, Email, and Phone. Two rows are visible: one for "studenta" and one for "kfc". The row for "studenta" is highlighted with a red border. At the bottom of the table is a green button labeled "Add a new student".

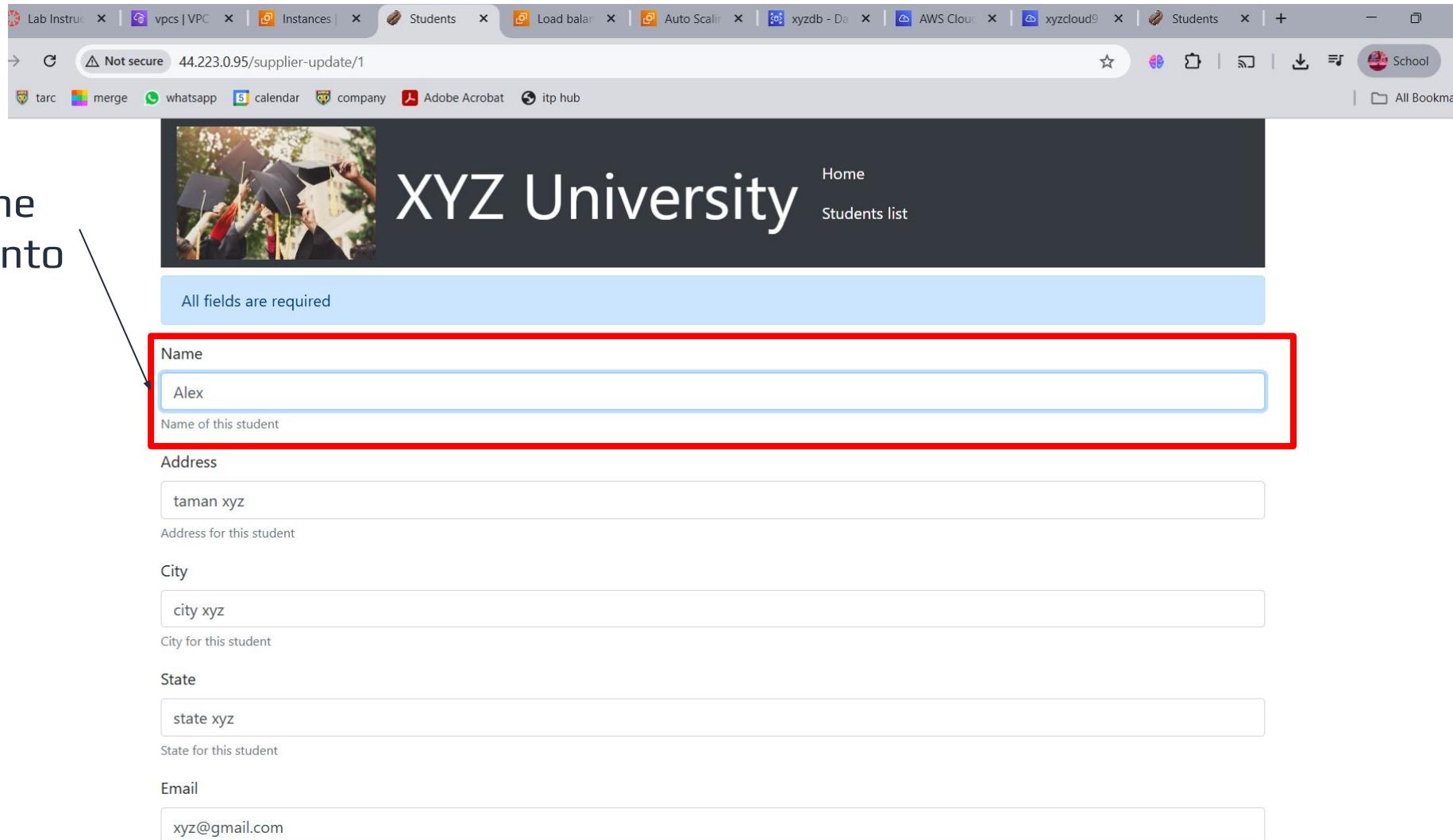
Name	Address	City	State	Email	Phone
studenta	taman xyz	city xyz	state xyz	xyz@gmail.com	0000000099
kfc	taman uni xyz	xyz city	uni state	kfc@gmail.com	015567787645

**Add a new student**

# Performance & Functionality

## Operational tasks: updating record (part 1)

Update the student name "studenta" into "Alex"



All fields are required

Name  
Alex  
Name of this student

Address  
taman xyz  
Address for this student

City  
city xyz  
City for this student

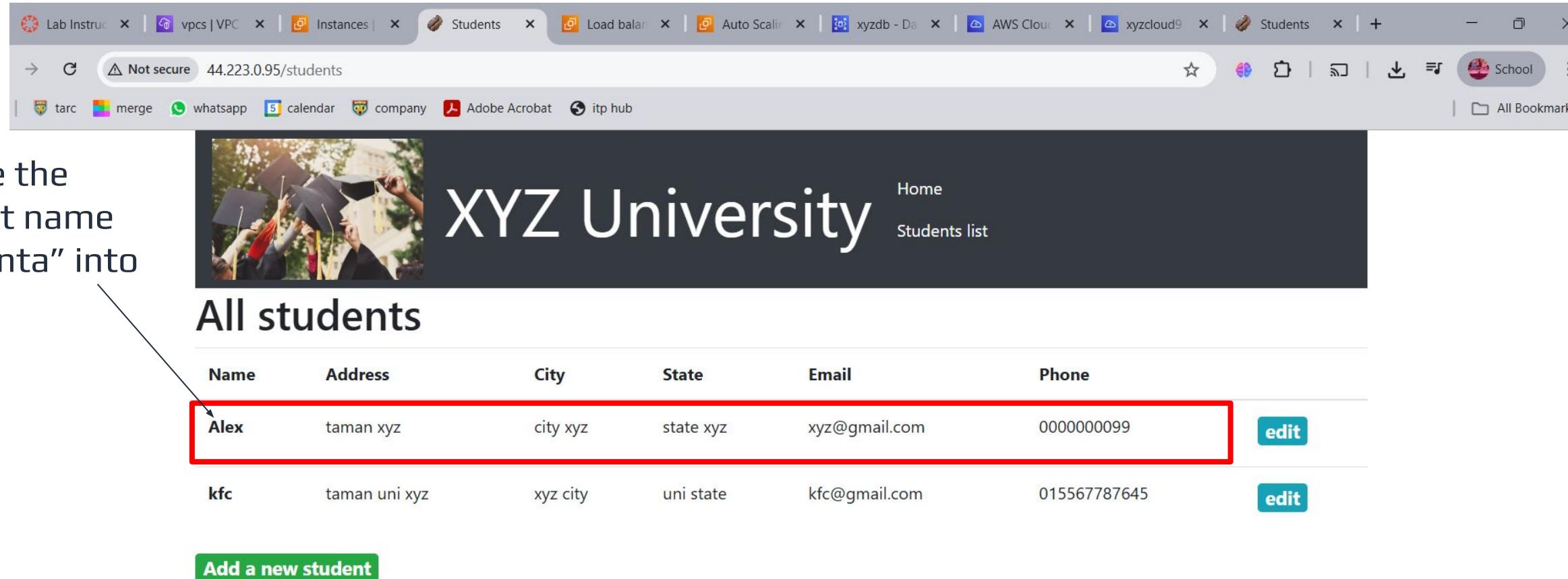
State  
state xyz  
State for this student

Email  
xyz@gmail.com

# Performance & Functionality

## Operational tasks: updating record (part 2)

Update the student name "studenta" into "Alex"



XYZ University

All students

Name	Address	City	State	Email	Phone
Alex	taman xyz	city xyz	state xyz	xyz@gmail.com	0000000099
kfc	taman uni xyz	xyz city	uni state	kfc@gmail.com	015567787645

Add a new student

# Performance & Functionality

## Migrated data was available

The screenshot shows a browser-based terminal interface with multiple tabs and a sidebar. The main area displays a MySQL session:

```
mysql - "ip-10-0-0-25.ec2.x" Immediate
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.6.4
npm notice To update run: npm install -g npm@11.6.4
npm notice
npm error A complete log of this run can be found in: /home/ec2-user/.npm/_logs/2025-11-28T09_48_26_054Z-debug-0.log
voclabs:~/environment $ mysql -h xyzdb.crgme0sob1d3.us-east-1.rds.amazonaws.com -u nodeapp -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 69
Server version: 8.0.43 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 STUDENTS;
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 [STUDENTS]> select * from students;
+----+-----+-----+-----+-----+-----+
| id | name | address | city | state | email | phone |
+----+-----+-----+-----+-----+-----+
| 1 | Alex | taman xyz | city xyz | state xyz | xyz@gmail.com | 0000000099 |
| 3 | kfc | taman uni xyz | xyz city | uni state | kfc@gmail.com | 015567787645 |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.006 sec)

MySQL [STUDENTS]>
```

A yellow callout box highlights the command `mysql -h xyzdb.crgme0sob1d3.us-east-1.rds.amazonaws.com -u nodeapp -p` with the text "connect to database using database endpoint". Another yellow callout box highlights the output of the `select * from students;` query with the text "display all the records in STUDENTS database".

## Performance & Functionality

```
voclabs:~/environment $ mysql -h xyzdb.crgme0so6ld3.us-east-1.rds.amazonaws.com -u nodeapp -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MySQL connection id is 69  
Server version: 8.0.43 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 STUDENTS;  
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 [STUDENTS]> select * from students;  
+----+-----+-----+-----+-----+  
| id | name | address      | city     | state    | email           | phone      |  
+----+-----+-----+-----+-----+  
| 1  | Alex  | taman xyz    | city xyz | state xyz | xyz@gmail.com | 0000000099 |  
| 3  | kfc   | taman uni xyz | xyz city | uni state | kfc@gmail.com | 015567787645 |  
+----+-----+-----+-----+-----+  
2 rows in set (0.006 sec)
```

display all the records in STUDENTS database

# Performance & Functionality

The screenshot shows the AWS RDS (Aurora and RDS) console for the 'xyzdb' database. The database identifier 'xyzdb' is highlighted with a red box and a yellow background. A red arrow points from the text 'RDS database' at the top to the highlighted 'xyzdb' identifier.

**RDS database**

**Aurora and RDS** > **Databases** > **xyzdb**

**xyzdb**

**Summary**

DB identifier	Status	Role	Engine	Recommendations
xyzdb	Available	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
-	db.t4g.micro		us-east-1a	

**Connectivity & security**

Endpoint & port	Networking	Security
<b>Endpoint</b> xyzdb.crgme0so6ld3.us-east-1.rds.amazonaws.com	<b>Availability Zone</b> us-east-1a	<b>VPC security groups</b> dbaccess (sg-0f21d1fdfda4c0c22)
<b>Port</b> 3306	<b>VPC</b> xyzvpc-vpc (vpc-07eeef8704f8a82a98)	<b>Active</b>
		<b>Publicly accessible</b> No

# Solution Overview:

## Scalability

# Scalability

The screenshot shows the AWS Auto Scaling Groups activity history for an Auto Scaling group named 'xyzas'. Two successful launches are listed:

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0214f24348455c461	At 2025-11-28T09:41:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-11-28T09:41:39Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 November 28, 05:41:41 PM +08:00	2025 November 28, 05:41:46 PM +08:00
Successful	Launching a new EC2 instance: i-05cd0f6a46c34c261	At 2025-11-28T09:41:34Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-11-28T09:41:39Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 November 28, 05:41:41 PM +08:00	2025 November 28, 05:42:13 PM +08:00

A yellow callout box highlights the text: "Auto Scaling automatically scale out by increasing the number of instances and staying within range, without reaching maximum limit". A red arrow points from this callout to the second launch entry in the activity history table.

# Solution Overview:

## High Availability

# High Availability

The screenshot shows the AWS EC2 Instances page with five running instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
xyzfinal	i-05cd0f6a46c34c261	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-3-84-96-89.comput...	3.84.96
aws-cloud9-xy...	i-0aa6134d8ab1e27af	Running	t3.micro	3/3 checks passed	<a href="#">View alarms</a> +	us-east-1a	ec2-54-146-204-177.co...	54.146
xyz1	i-0c2c6f0e697f649d2	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1a	ec2-98-88-37-34.comp...	98.88.3
xyz2	i-0689401f029d86ff0	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1a	ec2-44-223-0-95.comp...	44.223
xyzfinal	i-0214f24348455c461	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1a	ec2-3-94-132-163.com...	3.94.12

A red box highlights the 'Availability Zone' column, and a yellow callout box points to it with the text: "The instances have different Availability Zones".

# High Availability

The screenshot shows the AWS VPC console for a VPC named `vpc-07eef8704f8a83ab8`. The main configuration pane displays various settings like Main network ACL, IPv6 CIDR, and Route 53 Resolver DNS Firewall rule groups. A yellow callout box highlights that both availability zones have public and private subnets. The Resource map section shows four subnets: `xyzvpc-subnet-public1-us-east-1a`, `xyzvpc-subnet-private1-us-east-1a`, `xyzvpc-subnet-public2-us-east-1b`, and `xyzvpc-subnet-private2-us-east-1b`. Red boxes highlight the subnets in each availability zone, and red arrows point from the text in the callout box to these highlighted subnets.

Both availability zones have public and private subnets

Subnet ID	Availability Zone	Type
<code>xyzvpc-subnet-public1-us-east-1a</code>	us-east-1a	Public
<code>xyzvpc-subnet-private1-us-east-1a</code>	us-east-1a	Private
<code>xyzvpc-subnet-public2-us-east-1b</code>	us-east-1b	Public
<code>xyzvpc-subnet-private2-us-east-1b</code>	us-east-1b	Private

# High Availability

The screenshot shows the AWS VPC Resource Map for a specific VPC. The left sidebar lists various network components like Subnets, Route tables, Internet gateways, etc. The main area has tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The Resource map tab is selected, displaying a diagram of network resources:

- Subnets (4):** Subnets within this VPC are categorized by AZ:
  - us-east-1a:** xyzvpc-subnet-public1-us-east-1a (10.0.0.0/27), xyzvpc-subnet-private1-us-east-1a (10.0.64/27)
  - us-east-1b:** xyzvpc-subnet-public2-us-east-1b (10.0.0.32/27), xyzvpc-subnet-private2-us-east-1b (10.0.0.96/27)
- Route tables (4):** Route network traffic to resources
  - xyzvpc-rtb-private1-us-east-1a: 1 subnet association, 2 routes including local
  - xyzvpc-rtb-public:** 2 subnet associations, 2 routes including local
  - rtb-05cf7316f2f5ee23c: No subnet associations, 1 route including local
  - xyzvpc-rtb-private2-us-east-1b: 1 subnet association, 2 routes including local
- Network Connections (2):** Connections to other networks
  - xyzvpc-igw:** Internet routes to 2 public subnets, 0 private subnets route to the Internet
  - xyzvpc-vpce-s3: Gateway endpoint to S3

A yellow callout box highlights the connection between the public subnets and their associated route table:

Public subnets are associated with their own route table

# High Availability

The screenshot shows the AWS VPC Resource Map interface. On the left, there's a navigation sidebar with sections for VPC dashboard, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started). The main area has tabs for Resource map, CIDRs, Flow logs, Tags, and Integrations. The Resource map tab is selected, displaying a network diagram. It shows four subnets: xyzvpc-subnet-public1-us-east-1a (A), xyzvpc-subnet-private1-us-east-1a (A), xyzvpc-subnet-public2-us-east-1b (B), and xyzvpc-subnet-private2-us-east-1b (B). These are connected to four route tables: xyzvpc-rtb-private1-us-east-1a, xyzvpc-rtb-public, rtb-05cf731f2f5ee23c, and xyzvpc-rtb-private2-us-east-1b. The xyzvpc-rtb-private1-us-east-1a and xyzvpc-rtb-private2-us-east-1b route tables are highlighted with blue outlines. A network connection named xyzvpc-igw connects to xyzvpc-rtb-public. Another connection, xyzvpc-vpce-s3, connects to xyzvpc-rtb-private2-us-east-1b and is highlighted with an orange outline. A yellow callout box with black text states: "Private subnets are associated with their own route table". The browser address bar shows the URL: us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-07eef8704f8a83ab8.

# High Availability

## Managed services: Secrets Manager

The screenshot shows a web browser window with multiple tabs open, including 'Secrets', 'Instances', 'Students', 'Load bal...', 'Auto Sca...', 'xyzdb - L...', 'AWS Clo...', 'xyzcloud', and 'Students'. The main content area is the AWS Secrets Manager 'Secrets' page. The URL in the address bar is 'us-east-1.console.aws.amazon.com/secretsmanager/listsecrets?region=us-east-1'. The page title is 'Secrets' under 'AWS Secrets Manager'. The top navigation bar includes the AWS logo, a search bar, and account information ('Account ID: 6807-1518-8145' and 'vocabs/user4582294=xavierenky-wm22@student.ta...'). A large orange button on the right says 'Store a new secret'. The main table displays one secret:

Secret name	Description	Last retrieved (UTC)
Mydbsecret	Database secret for web app	November 28, 2025

# High Availability

## Managed services: Elastic Load Balancers

The screenshot shows the AWS EC2 Load Balancers console. On the left, a sidebar navigation menu includes sections for Images, Elastic Block Store, Network & Security, Load Balancing (selected), and Auto Scaling. Under Load Balancing, the 'Load Balancers' section is selected. The main content area displays a table titled 'Load balancers (1/1)'. The table has columns for Name, State, Type, Scheme, IP address type, VPC ID, and Availability Zones. A single row is selected for 'xyzlb', which is listed as Active, application type, Internet-facing scheme, IPv4 IP address type, and associated with VPC ID 'vpc-07eef8704f8a83ab8' across two availability zones. Below the table, a detailed view for the 'xyzlb' load balancer is shown, with the 'Details' tab selected. The 'Details' section contains fields for Load balancer type (Application), Status (Active), Scheme (Internet-facing), Hosted zone (Z355XDOTRQ7X7K), VPC (vpc-07eef8704f8a83ab8), Availability Zones (subnet-0e64513eed79d2b7c and subnet-0382115a03eb85296), Load balancer IP address type (IPv4), and Date created (November 28, 2025, 17:30 (UTC+08:00)).

# Solution Overview:

## Load Balancing

# Load Balancing

## All instances were evenly loaded at peak user load

The ALB-EC2 load balancer covers 2 availability zones

**Load balancer: xyzlb**

Details	Status	VPC	Load balancer IP address type
Load balancer type Application	Active	vpc-07eef8704f8a83ab8	IPv4
Scheme Internet-facing			
	Hosted zone Z35SXDOTRQ7X7K		Date created November 28, 2025, 17:30 (UTC+08:00)
		Availability Zones	
		<a href="#">subnet-0e64513eed79d2b7c</a> us-east-1a (use1-az4)	
		<a href="#">subnet-0382115a03eb85296</a> us-east-1b (use1-az6)	

**Load balancer ARN**  
arn:aws:elasticloadbalancing:us-east-1:680715188145:loadbalancer/app/xyzlb/da621934b2fe9b13

**DNS name Info**  
xyzlb-18195448.us-east-1.elb.amazonaws.com (A Record)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Load Balancing

## All instances were evenly loaded at peak user load

The screenshot shows the AWS EC2 Instances page with the following details:

**Instances (3/5) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
xyzfinal	i-05cd0f6a46c34c261	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-3-84-96-89.compute...	3.84.9€
aws-cloud9-xy...	i-0aa6134d8ab1e27af	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	ec2-54-146-204-177.co...	54.146
xyz1	i-0c2c60e697f649d2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-88-37-34.comp...	98.88.3
xyz2	i-0689401f029d86ff0	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-44-223-0-95.comp...	44.223
xyzfinal	i-0214f24348455c461	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-94-132-163.com...	3.94.13

**3 instances selected**

**Metrics**

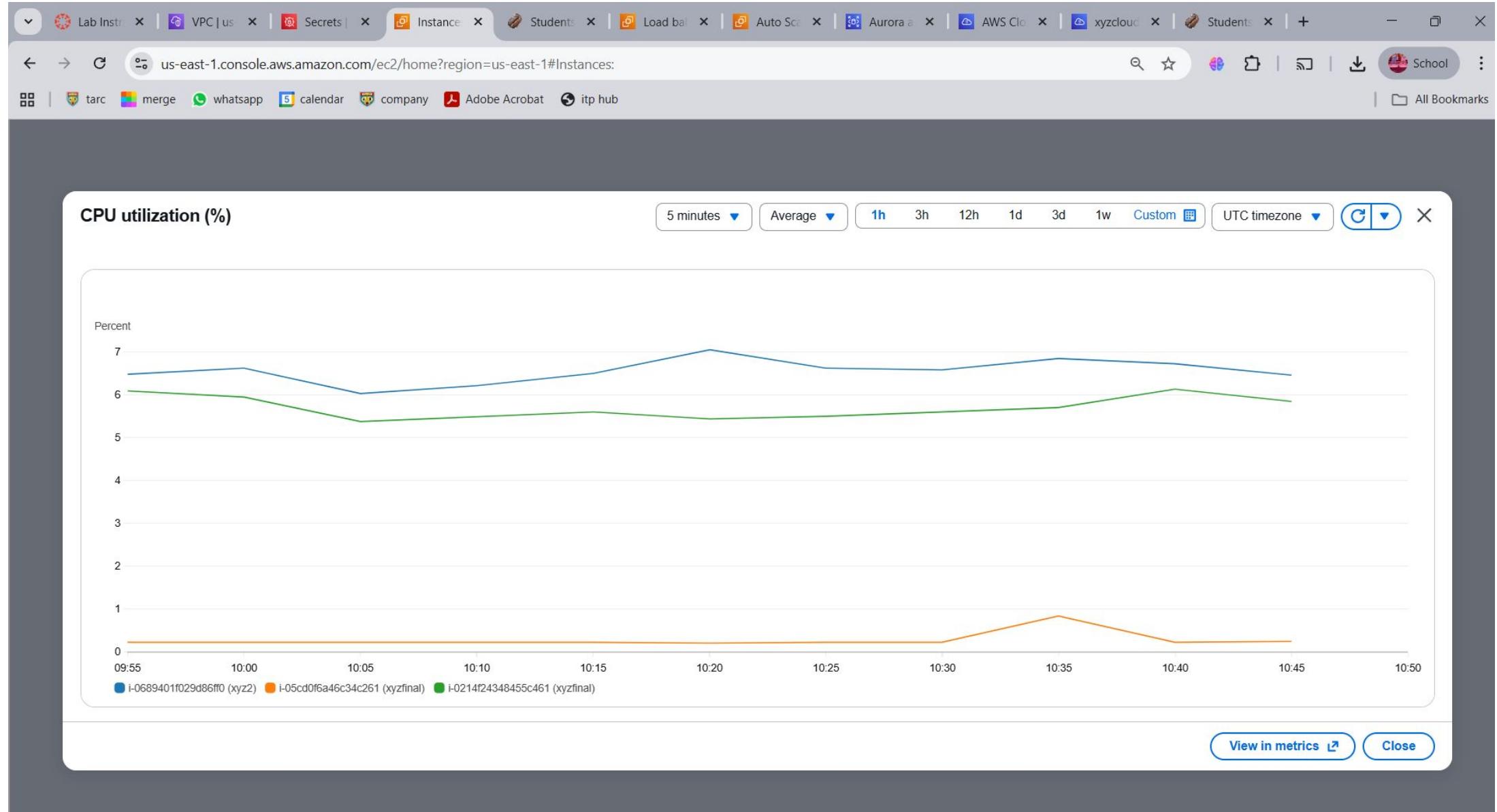
- Alarm recommendations
- Investigate with AI - new
- 1h (selected)
- 3h
- 12h
- 1d
- 3d
- 1w

**Metrics Overview**

- CPU utilization (%)
- Network in (bytes)
- Network out (bytes)
- Network packets in (count)
- Network packets out (count)
- Metadata no token (count)
- CPU credit usage (count)
- CPU credit balance (count)

**Yellow Callout Box Text:** The EC2 instances, MyASG created by auto scaling and the main website

# Load Balancing



# Solution Overview:

## Security

# Security

## Allow http and cloud9 to access in port 3306

The screenshot shows the AWS Management Console with the URL [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroups](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroups). The left sidebar is collapsed, and the main content area displays the 'Security Groups' list. A red box highlights the 'Inbound rules' section for the 'xyzec2' security group.

**Security Groups (1/5) Info**

Name	Security group ID	Security group name	VPC ID	Description
<input checked="" type="checkbox"/> -	sg-0e6da3f85574cb17a	xyzec2	vpc-07eef8704f8a83ab8	allow http and cloud9 for 3306
<input type="checkbox"/> -	sg-0ad35b37860d1df35	default	vpc-014250eb44a48afc7	default VPC security group
<input type="checkbox"/> -	sg-036545d12a7cec3f8	aws-cloud9-xyzcloud9-a2e8ead9e09c4...	vpc-07eef8704f8a83ab8	Security group for AWS Cloud9 environ...
<input type="checkbox"/> -	sg-0ca61592181ea15b	default	vpc-07eef8704f8a83ab8	default VPC security group
<input type="checkbox"/> -	sg-0f21d1fdfda4c0c22	dbaccess	vpc-07eef8704f8a83ab8	allow ec2 and cloud9 to access database

**sg-0e6da3f85574cb17a - xyzec2**

**Inbound rules (2)**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/> -	sgr-0b3346b7ddb469752	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/> -	sgr-0ed2d5d9f2413c586	-	MYSQL/Aurora	TCP	3306	sg-036545d12a7cec3

# Security

## Only allow EC2 and Cloud9 to access database

The screenshot shows the AWS EC2 Security Groups page. The left sidebar is collapsed, and the main area displays a table of security groups. One row, 'sg-0f21d1fdfda4c0c22 - dbaccess', is selected and highlighted with a blue border. This row corresponds to the security group defined in the configuration file.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0e6da3f85574cb17a	xyzec2	vpc-07eef8704f8a83ab8	allow http and cloud9 for 3306
-	sg-0ad35b37860d1df35	default	vpc-014250eb44a48afc7	default VPC security group
aws-cloud9-xyzcloud9...	sg-036545d12a7cec3f8	aws-cloud9-xyzcloud9-a2e8ead9e09c4...	vpc-07eef8704f8a83ab8	Security group for AWS Cloud9 environ...
-	sg-0c0a61592181ea15b	default	vpc-07eef8704f8a83ab8	default VPC security group
<input checked="" type="checkbox"/>	sg-0f21d1fdfda4c0c22	dbaccess	vpc-07eef8704f8a83ab8	allow ec2 and cloud9 to access database

Below the table, a modal window titled 'sg-0f21d1fdfda4c0c22 - dbaccess' is open, specifically showing the 'Inbound rules' tab. This tab lists two rules, both of which are highlighted with a red border. The first rule allows MySQL/Aurora traffic on port 3306 from the security group 'sg-036545d12a7cec3f8'. The second rule allows MySQL/Aurora traffic on port 3306 from the security group 'sg-0e6da3f85574cb17a'.

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0828e93e3a53950e5	-	MySQL/Aurora	TCP	3306	sg-036545d12a7cec3f8
-	sgr-0f0e94544ab38ae41	-	MySQL/Aurora	TCP	3306	sg-0e6da3f85574cb17a

# Security

## The DB credentials were stored securely in Secrets Manager

The screenshot shows the AWS Secrets Manager console in a web browser. The URL in the address bar is `us-east-1.console.aws.amazon.com/secretsmanager/listsecrets?region=us-east-1`. The page title is "Secrets". A search bar at the top left contains the placeholder text "Filter secrets by name, description, tag key, tag value, owning service or primary Region". On the right side, there is a "Store a new secret" button. The main table displays one secret:

Secret name	Description	Last retrieved (UTC)
Mydbsecret	Database secret for web app	November 28, 2025

The "Secret name" column for the "Mydbsecret" row is highlighted with a red box.

# Security

The load balancer was protected, with security groups having access only on port 3306

The screenshot shows the AWS EC2 Load Balancers console. On the left, a sidebar navigation includes: Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main content area displays a table of Load balancers, with one entry named "xyzlb". The table columns are: Name, State, Type, Scheme, IP address type, VPC ID, Availability Zones, and Security groups. The "xyzlb" row has a checked checkbox in the first column. The "Actions" button is visible above the table. Below the table, the "Load balancer: xyzlb" details page is shown, with the "Security" tab selected. The "Security groups" section lists one item: "sg-0e6da3f85574cb17a" with the description "xyzec2 allow http and cloud9 for 3306". A red box highlights this row. At the bottom of the page, there are tabs for Details, Listeners and rules, Network mapping, Resource map, Monitoring, Integrations, Attributes, Capacity, and Tags. The AWS logo and CloudShell/Feedback links are at the very bottom.

Name	State	Type	Scheme	IP address type	VPC ID	Availability Zones	Security groups
xyzlb	Active	application	Internet-facing	IPv4	vpc-07eef8704f8a83ab8	2 Availability Zones	sg-0e6da3f85574cb17a

**Load balancer: xyzlb**

Details    Listeners and rules    Network mapping    Resource map    **Security**    Monitoring    Integrations    Attributes    Capacity    Tags

**Security groups (1)**

A security group is a set of firewall rules that control the traffic to your load balancer.

Security Group ID	Name	Description
sg-0e6da3f85574cb17a	xyzec2	allow http and cloud9 for 3306

# Solution Overview:

## Monitoring and Threat Detection



# Monitoring and Threat Detection

## Mydbsecret information

The screenshot shows the AWS Secrets Manager console with the URL `us-east-1.console.aws.amazon.com/secretsmanager/secret?name=Mydbsecret&region=us-east-1&tab=versions`. The browser tabs include Lab Ins, VPC, Mydbsecret, Instances, Security, Studen, Load b, Auto S, Aurora, AWS C, xyzclo, Studen, and others. The AWS navigation bar shows the account ID: 6807-1518-8145 and the region: United States (N. Virginia). The main page displays the 'Mydbsecret' secret details, including its encryption key (aws/secretsmanager), secret name (Mydbsecret), and secret ARN (arn:aws:secretsmanager:us-east-1:680715188145:secret:Mydbsecret-E4O1uO). The 'Versions' tab is selected, showing one version (9db16e94-4545-4d82-905c-7a4ce906ff54) labeled as 'AWS CURRENT'. The secret description is 'Database secret for web app' and the secret type is listed as '-'. At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information for 2025, Amazon Web Services, Inc. or its affiliates, and links for Privacy, Terms, and Cookie preferences.

**Secret details**

Encryption key: aws/secretsmanager

Secret name: Mydbsecret

Secret ARN: arn:aws:secretsmanager:us-east-1:680715188145:secret:Mydbsecret-E4O1uO

Secret description: Database secret for web app

Secret type: -

**Versions**

**Secret versions (1)**

Version ID	Staging labels	Last accessed	Created on (UTC)
9db16e94-4545-4d82-905c-7a4ce906ff54	AWS CURRENT	November 28, 2025	November 28, 2025 at 09:07:26



# Cost Estimation

**aws** pricing calculator

Feedback Language: English ▾ Contact Sales ▾ Create an AWS Account

AWS Pricing Calculator > My Estimate > Edit Amazon EC2

Edit Amazon EC2 [Info](#)

Description  
EC2 Estimate

Choose a location type [Info](#) Choose a Region

Region US East (N. Virginia)

EC2 specifications [Info](#)

Tenancy  
Choose the tenancy type to run your Amazon EC2 instances on.  
Shared Instances

Operating system  
Choose the operating system to run your Amazon EC2 instances on.  
Ubuntu Pro

Workloads  
Choose the graph that best represents your monthly workload

Constant usage  Daily spike traffic  Weekly spike traffic  Monthly spike traffic

Total Upfront cost: 0.00 USD Total Monthly cost: 14.16 USD Show Details ▾

Cancel **Update**

aws pricing calculator

AWS Pricing Calculator > My Estimate > Edit Amazon Virtual Private Cloud (VPC)

## Edit Amazon Virtual Private Cloud (VPC) Info

Description  
Enter a description for your estimate

Choose a location type Info  
Region ▾

Choose a Region  
US East (N. Virginia)

Select VPC service(s) that you want to estimate

VPN Connection     Network Access Analyzer     Reachability Analyzer  
 Traffic Mirroring     Public IPv4 Address     IPAM  
 Data Transfer     VPC Route Server     Network Address Translation (NAT) Gateway  
 Transit Gateway     AWS PrivateLink     Gateway Load Balancer  
 Cloud WAN

VPC Configuration Summary

Total Upfront cost: 0.00 USD    Show Details ▾    Cancel

Total Monthly cost: 0.00 USD

Privacy | Site terms | Cookie preferences | © 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Elastic Load Balancing

aws pricing calculator

Feedback Language: English ▾ Contact Sales ▾ Create an AWS

Successfully updated Amazon RDS for MySQLEstimate.

AWS Pricing Calculator > My Estimate > Edit Elastic Load Balancing

## Edit Elastic Load Balancing Info

Description  
ELB

Choose a location type Info Choose a Region

Region ▾ US East (N. Virginia)

### Elastic Load Balancing Info

Select Elastic Load Balancing Function Options that you want to estimate

Application Load Balancer  Network Load Balancer  Gateway Load Balancer

Classic Load Balancer

### Application Load Balancer feature

Total Upfront cost: 0.00 USD Total Monthly cost: 16.47 USD Show Details ▾ Cancel

# Secrets Manager

[AWS Pricing Calculator](#) > [My Estimate](#) > [Edit AWS Secrets Manager](#)

## Edit AWS Secrets Manager [Info](#)

Description

Secret

Choose a location type [Info](#)

Choose a Region

Region

US East (N. Virginia)

## Secrets configuration [Info](#)

Number of secrets

1

Average duration of each secret

Validity timeframe of each secret. A secret valid for an entire month would be 730 hours or 30 days.

Value

Unit

30

days

**Total Upfront cost:** 0.00 USD

Show Details ▾

**Total Monthly cost:** 0.40 USD

Cancel

[Privacy](#) | [Site terms](#) | [Cookie preferences](#) | © 2025 Amazon Web Services, Inc. or its affiliates. All rights reserved.



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws pricing calculator

Feedback Language: English ▾ Contact Sales [?] Create an AWS Account

Successfully updated Amazon RDS for MySQLEstimate. X

AWS Pricing Calculator > My Estimate > Edit Amazon RDS for MySQL

Edit Amazon RDS for MySQL Info

Description

Enter a description for your estimate

Choose a location type Info

Region ▾

Choose a Region

US East (N. Virginia) ▾

MySQL instance specifications Info

Nodes

Enter the number of DB instances that you need.

1

db.t4g.micro X

Selected Instance:

Total Upfront cost: 0.00 USD Show Details ▾ Cancel Update

Total Monthly cost: 68.11 USD

[Privacy](#) | [Site terms](#) | [Cookie preferences](#) | © 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Final Estimation Cost

aws pricing calculator

Feedback Language: English ▾ Contact Sales Create an AWS Acc

My Estimate [Edit](#)

Estimate summary [Info](#)

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	99.14 USD	<b>1,189.68 USD</b> Includes upfront cost

Getting Started with AWS

[Get started for free](#) [Contact Sales](#)

**My Estimate**

Duplicate Delete Move to Create group Add support Add service

Find resources

<input type="checkbox"/>	Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary	
<input type="checkbox"/>	Amazon EC2	<a href="#">Edit</a>	-	0.00 USD	14.16 USD	EC2 Estimate	US East (N. Virginia)	Tenancy (Shared Instances), Operating system (Ubuntu Pro), Workload (Monthly, ...)
<input type="checkbox"/>	Amazon Virtual Priv...	<a href="#">Edit</a>	-	0.00 USD	0.00 USD	-	US East (N. Virginia)	Working days per month (22)
<input type="checkbox"/>	Elastic Load Balancing	<a href="#">Edit</a>	-	0.00 USD	16.47 USD	ELB	US East (N. Virginia)	Number of Application Load Balancers (1)
<input type="checkbox"/>	AWS Secrets Manager	<a href="#">Edit</a>	-	0.00 USD	0.40 USD	Secret	US East (N. Virginia)	Number of secrets (1), Average duration of each secret (30 days), Number of API c...
<input type="checkbox"/>	Amazon RDS for MySQL	<a href="#">Edit</a>	-	0.00 USD	68.11 USD	-	US East (N. Virginia)	Storage amount (20 GB), Storage for each RDS instance (General Purpose SSD (gp...)



# Lesson Learned

# Lessons learned (Challenges)

---

- **Challenge:** Configuring Auto Scaling to handle peak traffic and prevent server overload.

Solution: We initially struggled with determining the right scaling thresholds. After multiple tests, we implemented Target Tracking Scaling Policies, which allowed the system to automatically adjust EC2 instances based on real-time CPU utilization. This helped maintain performance during peak periods without over-provisioning resources.

- **Challenge:** Securely connecting the web application to the Amazon RDS database.

Solution: We encountered issues with storing sensitive database credentials securely. By using AWS Secrets Manager, we were able to securely store and retrieve these credentials dynamically. Configuring the appropriate IAM roles ensured that only authorized EC2 instances could access these secrets.

# Lessons learned (Challenges)

---

- Challenge: Managing costs while ensuring high availability and performance.

Solution: We implemented Elastic Load Balancer (ELB) and EC2 Auto Scaling to distribute traffic efficiently and scale resources as needed. We also utilized Reserved Instances for predictable workloads to reduce costs while maintaining availability.

- Challenge: Creating a fault-tolerant system across multiple availability zones (AZs).

Solution: To overcome this, we configured the architecture to deploy EC2 instances in multiple availability zones. This ensured that if one AZ went down, traffic would automatically be routed to the healthy instances in another AZ, improving fault tolerance.

- Challenge: Optimizing database performance under heavy load.

Solution: We optimized the RDS instance by enabling read replicas, allowing the database to handle more concurrent queries without degrading performance during high traffic periods.

# Lessons learned (Helpful Resources)

---

- Auto Scaling - AWS Documentation

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

- AWS Secrets Manager: Amazon RDS integration for master user password management

<https://youtu.be/tAXYA9QAR2o?si=ZrXduFoFIALzkjOD>

- Load Balancing Algorithm Explained - AWS - Amazon.com

[https://aws.amazon.com/what-is/load-balancing/#:~:text=Elastic%20Load%20Balancing%20\(ELB\)%20is.AWS%20and%20on%2Dpremises%20resources.](https://aws.amazon.com/what-is/load-balancing/#:~:text=Elastic%20Load%20Balancing%20(ELB)%20is.AWS%20and%20on%2Dpremises%20resources.)

- AWS RDS Best Practices

<https://jayendrapatil.com/aws-certification-rds-best-practices/>

- AWS Well-Architected Framework

[https://d1.awsstatic.com/whitepapers/architecture/AWS\\_Well-Architected\\_Framework.pdf](https://d1.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf)

# Lessons learned (Helpful Resources)

---

- AWS Tutorial to create Application Load Balancer and Auto Scaling Group

[https://youtu.be/fZuxp\\_pOzgl?si=6XqU9OITNtReMi0-](https://youtu.be/fZuxp_pOzgl?si=6XqU9OITNtReMi0-)

- Harnessing AWS CloudWatch for Effective Monitoring: A Comprehensive Guide

<https://medium.com/@oguzhanhiziroglu/harnessing-aws-cloudwatch-for-effective-monitoring-a-comprehensive-guide-6c787abf7425>

- AWS VPC (Virtual Private Cloud)

<https://medium.com/@deepeshjaiswal6734/aws-vpc-virtual-private-cloud-965b7c82110d>

- AWS cost calculator used by experts to create cost estimation

<https://medium.com/@2017tejasgupta/create-your-cost-estimation-with-the-aws-cost-calculator-570ff7d3298b>

- The 50 Most Popular Linux & Terminal Commands - Full Course for Beginners

[https://www.youtube.com/watch?v=ZtqBQ68cfJc&ab\\_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=ZtqBQ68cfJc&ab_channel=freeCodeCamp.org)



# Lessons learned (New Skills)

---

- Auto Scaling & Elastic Load Balancing

We learned how to configure Auto Scaling policies to handle traffic fluctuations automatically and ensure high availability using Elastic Load Balancer.

- AWS Secrets Manager & Security Best Practices

We gained hands-on experience using AWS Secrets Manager to securely store sensitive information like database credentials. We also improved our knowledge of IAM roles to control access between services.

- Cost Management & Optimization

Through this project, we understood how to balance cost and performance using Reserved Instances, on-demand instances, and AWS Cost Explorer for detailed cost analysis and optimization.

# Lessons learned (New Skills)

---

- **Fault Tolerance**

We learned how to design a fault-tolerant architecture across multiple availability zones, ensuring that the system remains operational even during failures or outages in one zone.

- **Database Performance Tuning**

We became proficient in optimizing Amazon RDS performance by configuring read replicas and adjusting parameters for handling peak traffic efficiently.

- **Setting Up VPCs and Networking**

We learned how to design and configure a Virtual Private Cloud (VPC) with public and private subnets, enabling secure communication between the web servers (EC2) and the database (RDS). This also included setting up security groups and NAT gateways to control traffic flow and provide internet access to instances in private subnets without exposing them to the public internet.

# Lessons learned (Next Steps)

---

- **Implement Authentication and Authorization**

We plan to integrate user authentication and role-based access control to restrict access to specific parts of the application, improving security and user management.

- **Multi-Region Deployment**

To enhance disaster recovery and reduce latency for global users, we plan to deploy the application across multiple regions, ensuring that the system remains highly available even if an entire region fails.

- **Advanced Monitoring & Alerting**

We plan to implement more sophisticated monitoring using Amazon CloudWatch to track performance metrics and set up custom alarms for faster response to any issues that may arise.

# Lessons learned (Next Steps)

---

- **Data Encryption**

To further enhance security, we will implement encryption at rest and encryption in transit for all sensitive data, using AWS Key Management Service (KMS) for managing encryption keys.

- **Database Auto Scaling**

In the future, we plan to explore RDS auto-scaling to dynamically adjust the database resources based on demand, ensuring cost efficiency and performance during peak times.

- **Add Caching Layer**

To improve response times and reduce load on the database, we could integrate a caching solution like Amazon ElastiCache to store frequently accessed data.



# Guided and Challenge Lab Scores (Individual)

# Guided and Challenge Lab Scores (Cho Wei Bin)

Grades for chowb-wp21@student.tarc.edu.my

 Print Grades (javascript:window.print())

Course **Arrange By**

AWS Academy Cloud Arch

Module

Apply

Name	Due	Submitted	Status	Score
<a href="#">Module 2 Knowledge Check Assignments</a>				- / 100
<a href="#">Guided Lab: Exploring AWS Identity and Access Management (IAM) Lab Assignments</a>	Nov 15 at 4:59pm	56 / 56		
<a href="#">Module 3 Knowledge Check Assignments</a>				- / 100
<a href="#">Challenge (Café) lab: Creating a Static Website for the Café Lab Assignments</a>	Nov 15 at 5:28pm	29 / 29		
<a href="#">Module 4 Knowledge Check Assignments</a>				- / 100
<a href="#">Guided lab: Introducing Amazon Elastic File System (Amazon EFS) Lab Assignments</a>	Nov 15 at 5:49pm	15 / 15		
<a href="#">Challenge (Café) lab: Creating a Dynamic Website for the Café Lab Assignments</a>	Nov 15 at 11:33pm	27 / 30		
<a href="#">Module 5 Knowledge Check Assignments</a>				- / 100

Name	Due	Submitted	Status	Score
<a href="#">Guided lab: Creating an Amazon RDS Database Lab Assignments</a>				20 / 20
<a href="#">Challenge (Café) lab: Migrating a Database to Amazon RDS Lab Assignments</a>				22 / 25
<a href="#">Module 6 Knowledge Check Assignments</a>				- / 100
<a href="#">Guided lab: Creating a Virtual Private Cloud Lab Assignments</a>				56 / 56
<a href="#">Challenge (Café) lab: Creating a VPC Networking Environment for the Café Lab Assignments</a>				56 / 56
<a href="#">Module 7 Knowledge Check Assignments</a>				- / 100
<a href="#">Guided lab: Creating a VPC Peering Connection Lab Assignments</a>				56 / 56
<a href="#">Module 8 Knowledge Check Assignments</a>				- / 100
<a href="#">Guided lab: Securing Applications by using Amazon Cognito Lab Assignments</a>				37.33 / 56
<a href="#">Guided lab: Encrypting Data at Rest by Using AWS Encryption Options Lab Assignments</a>				56 / 56
<a href="#">Module 9 Knowledge Check Assignments</a>				- / 100



# Guided and Challenge Lab Scores (Cho Wei Bin)

Name	Due	Submitted	Status	Score
<a href="#">Guided lab: Creating a Highly Available Environment</a> Lab Assignments	Dec 10 at 3:49pm		48.53 / 56	
<a href="#">Challenge (Café) lab: Creating a Scalable and Highly Available Environment for the Café</a> Lab Assignments	Dec 14 at 9:02pm		56 / 56	
<a href="#">Module 10 Knowledge Check</a> Assignments			- / 100	
<a href="#">Guided lab: Automating Infrastructure with AWS CloudFormation</a> Lab Assignments			- / 56	
<a href="#">Challenge (Café) lab: Automating Infrastructure Deployment</a> Lab Assignments			- / 56	
<a href="#">Module 11 Knowledge Check</a> Assignments			- / 100	
<a href="#">Module 12 Knowledge Check</a> Assignments			- / 100	
<a href="#">Guided lab: Building Decoupled Applications by Using Amazon SQS</a> Lab Assignments			- / 56	
<a href="#">Module 13 Knowledge Check</a> Assignments			- / 100	
<a href="#">Guided lab: Implementing a Serverless Architecture on AWS</a> Lab Assignments			- / 56	
<a href="#">(Optional) Guided lab: Breaking a Monolithic Node.js Application into Microservices</a>			- / 56	

Name	Due	Submitted	Status	Score
<b>Lab Assignments</b>				
<a href="#">Challenge (Café) lab: Implementing a Serverless Architecture for the Café</a> Lab Assignments				- / 56
<b>Module 14 Knowledge Check</b> Assignments				
<a href="#">Module 15 Knowledge Check</a> Assignments				- / 100
<a href="#">Guided Lab: Configuring Hybrid Storage and Migrating Data with AWS Storage Gateway S3 File Gateway</a> Lab Assignments				- / 40
<a href="#">Module 16 Knowledge Check</a> Assignments				- / 100
<a href="#">Capstone Project</a> Lab Assignments				- / 35
<a href="#">Academy Cloud Architecting Course Assessment</a> Assignments				- / 100
<a href="#">Capstone Project</a> Lab Assignments				- / 56
<b>Lab Assignments</b>			94.33%	534.87 / 567.00
<b>Assignments</b>			N/A	0.00 / 0.00
<b>Total</b>			94.33%	534.87 / 567.00

# Guided and Challenge Lab Scores (Heng Zheng Teck)

Grades for hengzt-wm22@student.tarc.edu.my

Course **AWS Academy Cloud Archi** ▾

Arrange By **Module** ▾

**Apply**

Print Grad

Name	Due	Submitted	Status	Score
Module 2 Knowledge Check Assignments	Dec 9 at 8:46am	100 / 100	✉	
Guided Lab: Exploring AWS Identity and Access Management (IAM) Lab Assignments	Nov 17 at 7:30am	56 / 56	✉	
Module 3 Knowledge Check Assignments	Dec 11 at 6:26am	100 / 100	✉	
Challenge (Cafe) lab: Creating a Static Website for the Café Lab Assignments	Dec 14 at 3:45pm	29 / 29	✉	
Module 4 Knowledge Check Assignments	Dec 11 at 6:44am	100 / 100	✉	
Guided lab: Introducing Amazon Elastic File System (Amazon EFS) Lab Assignments	Nov 27 at 3:38am	15 / 15	✉	
Challenge (Cafe) lab: Creating a Dynamic Website for the Café Lab Assignments	Nov 27 at 4:33am	15 / 30	✉	
Module 5 Knowledge Check Assignments	Dec 11 at 12:19pm	100 / 100	✉	

Guided lab: Creating an Amazon RDS Database Lab Assignments	Dec 10 at 3:49pm	20 / 20	✉
Challenge (Cafe) lab: Migrating a Database to Amazon RDS Lab Assignments	Dec 10 at 4:36pm	22 / 25	✉
Module 6 Knowledge Check Assignments	Dec 14 at 4:55pm	100 / 100	✉
Guided lab: Creating a Virtual Private Cloud Lab Assignments	Dec 10 at 5:13pm	56 / 56	✉
Challenge (Cafe) lab: Creating a VPC Networking Environment for the Café Lab Assignments	Dec 10 at 6:09pm	56 / 56	✉
Module 7 Knowledge Check Assignments		- / 100	
Guided lab: Creating a VPC Peering Connection Lab Assignments	Dec 11 at 7:13am	46.67 / 56	✉
Module 8 Knowledge Check Assignments		- / 100	
Guided lab: Securing Applications by using Amazon Cognito Lab Assignments	Dec 11 at 8:31am	46.67 / 56	✉
Guided lab: Encrypting Data at Rest by Using AWS Encryption Options Lab Assignments	Dec 11 at 8:57am	56 / 56	✉
Module 9 Knowledge Check Assignments		- / 100	
Guided lab: Creating a Highly Available Environment Lab Assignments	Dec 18 at 5:10pm	52.27 / 56	✉
Challenge (Café) lab: Creating a Scalable and Highly Available Environment for the Café Lab Assignments	Dec 19 at 1:34am	19.87 / 56	✉



# Guided and Challenge Lab Scores (Heng Zheng Teck)

Guided lab: Automating Infrastructure with AWS CloudFormation Lab Assignments	Dec 18 at 4:21pm	56 / 56	•	✉
Challenge (Café) lab: Automating Infrastructure Deployment Lab Assignments		- / 56		
Module 11 Knowledge Check Assignments		- / 100		
Module 12 Knowledge Check Assignments		- / 100		
Guided lab: Building Decoupled Applications by Using Amazon SQS Lab Assignments		- / 56		
Module 13 Knowledge Check Assignments		- / 100		
Guided lab: Implementing a Serverless Architecture on AWS Lab Assignments		- / 56		
(Optional) Guided lab: Breaking a Monolithic Node.js Application into Microservices Lab Assignments		- / 56		
Challenge (Café) lab: Implementing a Serverless Architecture for the Café Lab Assignments		- / 56		
Module 14 Knowledge Check Assignments		- / 100		
Module 15 Knowledge Check Assignments		- / 100		
Guided Lab: Configuring Hybrid Storage and Migrating Data with AWS Storage Gateway S3 File Gateway Lab Assignments		- / 40		
Module 16 Knowledge Check Assignments		- / 100		
Capstone Project Lab Assignments				- / 35
Academy Cloud Architecting Course Assessment Assignments				- / 100
Capstone Project Lab Assignments				- / 56
Lab Assignments				87.72%    546.47 / 623.00
Assignments				100%    500.00 / 500.00
Total				<b>93.19%</b> <b>1,046.47 / 1,123.00</b>



# Guided and Challenge Lab Scores (Xavier Ngow Kar Yuen)

Grades for xaviernky-wm22@student.tarc.edu.my

Print Gr

Course      Arrange By

AWS Academy Cloud Archi

Module

Apply

Name	Due	Submitted	Status	Score
Module 2 Knowledge Check Assignments				- / 100
Guided Lab: Exploring AWS Identity and Access Management (IAM) Lab Assignments	Dec 20 at 7:41pm	56 / 56	✉	
Module 3 Knowledge Check Assignments	Dec 20 at 8:18pm	30 / 100	✉	
Challenge (Cafe) lab: Creating a Static Website for the Café Lab Assignments	Dec 20 at 8:16pm	0 / 29	✉	
Module 4 Knowledge Check Assignments	Dec 20 at 8:28pm	20 / 100	✉	
Guided lab: Introducing Amazon Elastic File System (Amazon EFS) Lab Assignments	Dec 20 at 8:52pm	10 / 15	✉	

Challenge (Cafe) lab: Creating a Dynamic Website for the Café Lab Assignments	Dec 20 at 9:35pm	6 / 30	✉
Module 5 Knowledge Check Assignments	Dec 20 at 8:26pm	20 / 100	✉
Guided lab: Creating an Amazon RDS Database Lab Assignments	Dec 21 at 6:46am	15 / 20	✉
Challenge (Cafe) lab: Migrating a Database to Amazon RDS Lab Assignments	Dec 21 at 7:31am	25 / 25	✉
Module 6 Knowledge Check Assignments	Dec 20 at 8:55pm	10 / 100	✉
Guided lab: Creating a Virtual Private Cloud Lab Assignments	Dec 21 at 7:59am	56 / 56	✉
Challenge (Cafe) lab: Creating a VPC Networking Environment for the Café Lab Assignments	Dec 21 at 8:30am	51 / 56	✉
Module 7 Knowledge Check Assignments	Dec 20 at 8:57pm	30 / 100	✉
Guided lab: Creating a VPC Peering Connection Lab Assignments	Dec 21 at 8:49am	0 / 56	✉

# Guided and Challenge Lab Scores (Xavier Ngow Kar Yuen)

Module 8 Knowledge Check Assignments	Dec 20 at 8:58pm	40 / 100		<input checked="" type="checkbox"/>
Guided lab: Securing Applications by using Amazon Cognito Lab Assignments	Dec 21 at 9:23am	9.33 / 56		<input checked="" type="checkbox"/>
Guided lab: Encrypting Data at Rest by Using AWS Encryption Options Lab Assignments	Dec 21 at 9:46am	0 / 56		<input checked="" type="checkbox"/>
Module 9 Knowledge Check Assignments	Dec 20 at 9pm	20 / 100		<input checked="" type="checkbox"/>
Guided lab: Creating a Highly Available Environment Lab Assignments	Dec 21 at 11:54am	41.07 / 56		<input checked="" type="checkbox"/>
Challenge (Café) lab: Creating a Scalable and Highly Available Environment for the Café Lab Assignments	Dec 21 at 12:29pm	10.84 / 56		<input checked="" type="checkbox"/>
Module 10 Knowledge Check Assignments	Dec 20 at 9:02pm	20 / 100		<input checked="" type="checkbox"/>
Guided lab: Automating Infrastructure with AWS CloudFormation Lab Assignments		- / 56		
Challenge (Café) lab: Automating Infrastructure Deployment		- / 56		

Module 11 Knowledge Check Assignments	- / 100
Module 12 Knowledge Check Assignments	- / 100
Guided lab: Building Decoupled Applications by Using Amazon SQS Lab Assignments	- / 56
Module 13 Knowledge Check Assignments	- / 100
Guided lab: Implementing a Serverless Architecture on AWS Lab Assignments	- / 56
(Optional) Guided lab: Breaking a Monolithic Node.js Application into Microservices Lab Assignments	- / 56
Challenge (Café) lab: Implementing a Serverless Architecture for the Café Lab Assignments	- / 56
Module 14 Knowledge Check Assignments	- / 100
Module 15 Knowledge Check Assignments	- / 100

Guided Lab: Configuring Hybrid  
Storage and Migrating Data with AWS  
Storage Gateway S3 File Gateway  
Lab Assignments

- / 40

Module 16 Knowledge Check  
Assignments

- / 100

Capstone Project  
Lab Assignments

- / 35

Academy Cloud Architecting Course  
Assessment  
Assignments

- / 100

Capstone Project  
Lab Assignments

- / 56

Lab Assignments 49.42% 280.24 / 567.00

Assignments 23.75% 190.00 / 800.00

Total 34.4% 470.24 / 1,367.00



# Thank you