



**FACULTY OF COMPUTING &
INFORMATICS CIT4183 - INTRODUCTION
OF CLOUD COMPUTING
ASSIGNMENT**

Scenario: E-commerce Website

Trimester 1, Session 2023 / 2024





No.	Student ID	Student Name	Role	Signature
1.	1211208647	Ng Kean Tiong	Leader	
2.	1211208093	Tan Zhe Enn	Member	
3.	1211207817	Tan Ze Quan	Member	
4.	1221206199	Wong Mei Jing	Member	

Table of Contents

Table of Contents	1
Introduction	2
Problem Statement	3
Objective	4
Scope	5
Proposed AWS Solution	7
(i) System Architecture Diagram	7
(ii) Detailed Description of AWS Services Used	8
(iii) Implementation Steps	9
Project Timeline	11
Team Roles and Responsibilities	12
Conclusion	13
Assignment Rubrics	14

Introduction

Cloud computing model is based on a software infrastructure which is more flexible and cost-effectively compared with traditional hardware computing model. The fundamental technology that makes Cloud computing possible is the virtualization technology with a hypervisor that is responsible for managing the hardware resource. There are several Cloud service providers such as Microsoft Azure, Amazon Web Services(AWS), Google Cloud Platform(GCP) as well as IBM Cloud.

As we know that Amazon Web Service (AWS) is the one of the well-known and most comprehensive adapted clouds computing. It's self-service infrastructure that provides almost a total of 42,240 products and services. There are several basic services that are provided by AWS such as the computer service, security,identity and compliance service, storage service, database service, management and government service, AWS cost management service as well as the networking and content delivery service. Some of the services such as Amazon VPC, Elastic Beanstalk, Auto Scaling, AWS CloudFormation and the AWS Identity and Access Management(IAM), these services were provided by AWS with no charge, are free to use. It allows users to custom build, develop and manage their website with a low pay-as-you-go price. It allows users to build and run an app or web on a virtual computer, server as well as database.

Problem Statement

The aim of this program is to set up an e-commerce website using the AWS scenario to investigate effective development and management strategies of a business website to help a company to compete in a global market.

We are going to figure out how to host a website by launching an Amazon EC2 instance. As we know that network security is very important to prevent unauthorized access to network resources, we will also take part in figuring out how to build a secure dynamic e-commerce website to protect a company's confidential business data as well as the customer's private data being stolen by a hacker.

For a big company's e-commerce website such as Amazon and Ebay, they'll need to run a lot of backend programming systems. If they save their code directly inside a traditional hardware server, they will need to pay an expensive price per year for purchasing and maintaining fees. Based on this problem, we will also study how to run code without provisioning or managing servers using the Lambda function and storing the customer's data into a serverless database, so developers no need to worry about the capacity planning and improve time to market and agility.

Moreover, we are also going to investigate how to provide a high efficiency Internet connection for users from around the world through the AWS global infrastructure to help to promote the flow of the world's economy.

In this project, we are going to study how a Cloud Computing model can solve the deficiency of the traditional computing model and help a company to launch their own digital e-commerce website in a cost effective way with a Cloud Computing through the AWS scenario.

Objective**1. A scalable ecommerce platform:**

Develop an e-commerce platform that can scale to handle variable traffic and transaction volumes while maintaining high availability and performance.

2. Secure Transactions and Data:

Maintain security of transactions and consumer data while complying to business standards and regulations.

3. User-Friendly Experience:

Provide customers with an intuitive and seamless user experience across web and mobile platforms, including rapid navigation, search, and checkout processes.

4. Inventory Management in Real Time:

To keep track of stock levels, update product availability, and avoid overselling, use real-time inventory management.

5. Data Storage and Management:

Securely store and manage large volumes of agricultural data, ensuring scalability, durability, and easy access for analysis and reporting.

6. All-inclusive Analytics and Reporting:

Assure adherence to agricultural laws and regulations and offer extensive reporting features to assist in meeting regulatory and decision-making needs.

7. Economical Functions:

Reduce operating expenses by using AWS-managed services, which will guarantee an affordable infrastructure that can expand with the company.

Scope

1. Empowering User Experience

Components:

- AWS Services: Amazon S3 for static website hosting, AWS Amplify for frontend development, and AWS CloudFront for content delivery.

Implementation:

- Create a responsive and inclusive web application using AWS Amplify, ensuring ease of navigation and accessibility features.
- Host the storefront on Amazon S3 for reliability and scalability, with CloudFront optimizing global content delivery for faster load times.
- Implement features like personalized recommendations and customer reviews to enrich the shopping journey.

2. Ethical Product Representation

Components:

- AWS Services: Amazon DynamoDB for NoSQL database, AWS Lambda for serverless computing.

Implementation:

- Utilize DynamoDB to maintain accurate and detailed product information, including sourcing details, sustainability metrics, and ethical certifications.
- Develop AWS Lambda functions to automate validation processes and ensure consistency in product data across the platform.
- Empower customers to make informed choices by highlighting eco-friendly and socially responsible products.

3. Privacy-First Security Measures

Components:

- AWS Services: Amazon Cognito for user authentication, AWS IAM for access management, and AWS KMS for encryption.

Implementation:

- Configure Amazon Cognito to manage user authentication and authorization securely, with options for multi-factor authentication (MFA) and federated identities.
- Implement strict AWS IAM policies to enforce least privilege access control and protect customer data integrity.
- Utilize AWS Key Management Service (KMS) for encryption of sensitive data at rest and in

transit, ensuring compliance with data protection regulations.

4. Customer-Centric Order Management

Components:

- **AWS Services:** Amazon RDS for transactional data storage, AWS Lambda for backend processing, and Amazon API Gateway for API management.

Implementation:

- Utilize Amazon RDS to manage transactional data such as orders, payments, and inventory levels, ensuring real-time updates and accurate order tracking.
- Develop serverless AWS Lambda functions to orchestrate order workflows, including payment processing, inventory synchronization, and shipment tracking.
- Integrate with third-party logistics providers and payment gateways via secure API Gateway endpoints for seamless operations and proactive customer notifications.

5. Insights-Driven Growth Strategies

Components:

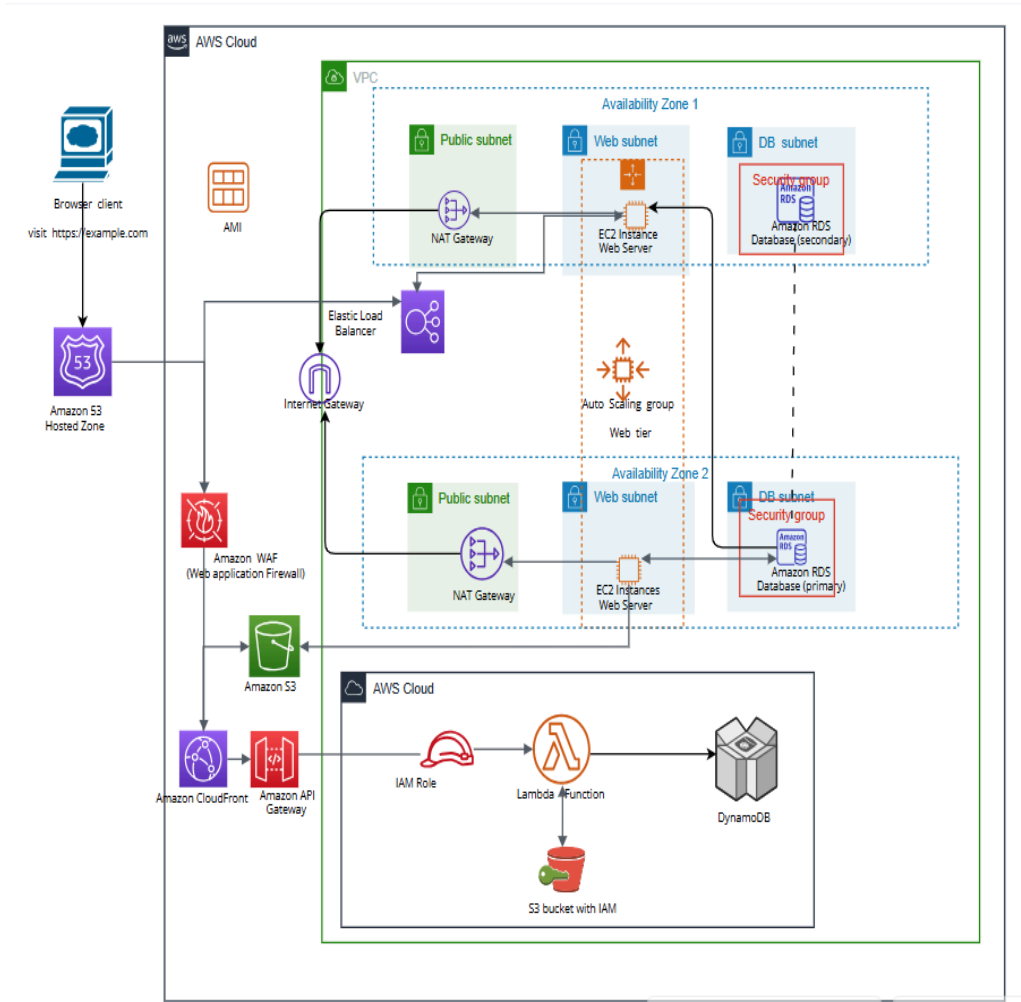
- **AWS Services:** Amazon CloudWatch for monitoring and logging, AWS CloudTrail for audit trails, and Amazon QuickSight for business intelligence.

Implementation:

- Set up Amazon CloudWatch alarms to monitor application performance metrics, detect anomalies, and ensure high availability.
- Enable AWS CloudTrail to audit user actions, enforce compliance, and maintain accountability in data handling.
- Utilize Amazon QuickSight for interactive dashboards and data visualizations, empowering stakeholders to make data-driven decisions for business growth and customer satisfaction.

Proposed AWS Solution

(i) System Architecture Diagram



(ii) Detailed Description of AWS Services Used

1. Amazon EC2 (Elastic Compute Cloud):

Purpose: Hosting the application servers and web servers.

Justification: Provides scalable compute capacity to handle varying loads from the e-commerce website.

2. Amazon S3 (Simple Storage Service):

Purpose: Storing static content such as images, videos, and downloadable files.

Justification: Highly durable and scalable storage solution, integrates well with Amazon CloudFront for content delivery.

3. Amazon RDS (Relational Database Service):

Purpose: Hosting the relational database (e.g., MySQL, PostgreSQL) for storing transactional data.

Justification: Managed service for databases, handles automated backups, and supports Multi-AZ deployment for high availability and fault tolerance.

4. Amazon EBS (Elastic Block Store):

Purpose: Providing block-level storage volumes for EC2 instances.

Justification: Allows for persistent data storage and can be used for boot volumes or additional storage volumes attached to EC2 instances.

5. Amazon Route 53:

Purpose: DNS web service for routing end users to Internet applications.

Justification: Supports domain registration, DNS routing, and health checking of endpoints (e.g., load balancers, web servers).

6. Elastic Load Balancer (ELB):

Purpose: Distributes incoming application or network traffic across multiple targets (e.g., EC2 instances).

Justification: Ensures high availability and fault tolerance by automatically distributing incoming traffic across multiple targets.

(iii) Implementation Steps

Step 1 : Setup Networking and DNS:

- Register domain with Amazon Route 53.
- Configure DNS records to point to Elastic Load Balancer.

Step 2 : Deploy EC2 Instances:

- Launch EC2 instances for web servers and application servers.
- Configure security groups, IAM roles, and instance profiles.

Step 3 : Configure Auto Scaling:

- Create an Auto Scaling group to automatically adjust the number of EC2 instances based on traffic demand.

Step 4 : Implement Amazon S3:

- Create S3 buckets for storing static content (e.g., images, CSS files).
- Configure permissions and enable versioning where necessary.

Step 5 : Set Up Amazon RDS:

- Select appropriate database engine (e.g., MySQL, PostgreSQL) and instance type.
- Configure Multi-AZ deployment for high availability.
- Set up database backups and maintenance windows.

Step 6 : Integrate CloudFront:

- Create a CloudFront distribution for CDN services.
- Configure origin settings to point to the S3 buckets and EC2 instances.

Step 7 : Security and Monitoring:

- Implement security best practices such as IAM roles, SSL certificates (via ACM), and network ACLs.
- Set up CloudWatch alarms for monitoring CPU utilization, latency metrics, and database performance.

Step 8 : Testing and Deployment:

- Conduct load testing and performance tuning to ensure scalability.
- Deploy application updates using AWS CodeDeploy or similar services.

Step 9 : Backup and Disaster Recovery:

- Regularly backup S3 data and RDS snapshots.
- Implement disaster recovery plans and perform periodic drills.

Project Timeline

Task	Week 12	Week 13	Week 14	Week 15
Introduction	✓			
Problem Statement		✓		
Objective		✓		
Scope			✓	
Proposed AWS solution			✓	
Team Roles and Responsibilities			✓	
Conclusion				✓
Assignment Rubrics				✓

Team Roles and Responsibilities

Role	Name	Responsible
Project Coordination	Ng Kean Tiong	Coordinates and manages project planning and communication, assigns tasks, makes clear roles and deadlines, and reduces risks to ensure project success.
Technical Architect	Tan Zhe Enn	Designs scalable and reliable technical architecture using AWS services. Guides implementation and optimization efforts.
E-commerce Analyst	Wong Mei Jing	Gathers and evaluates the needs of the business. outlines the parameters for the project's success. guarantees congruence with corporate goals.
Documentation and Reporting	Tan Ze Quan	Assembles presentations and keeps track of project material to inform stakeholders of developments and accomplishments.

Conclusion

This proposal outlines a comprehensive approach to building an e-commerce platform on AWS, leveraging its scalable and reliable services to ensure high performance, availability, and security. By implementing the proposed solution, businesses can effectively manage varying traffic loads, enhance user experience, and reduce operational costs associated with infrastructure maintenance.

Assignment Rubrics

The Assignment rubrics is as follows:

No	Assessment Details	Weightage %	Points [0-5]	Max points
1	Organization and Professional Presentation	3		15
2	Table of Contents	2		10
3	Body of Proposal	12		60
4	Formatting	3		15
	Total			100
Marks (30%)				

Criteria	Exemplary (5 points)	Proficient (4 points)	Satisfactory (3 points)	Needs Improvement (2 points)	Unsatisfactory (0-1 points)
Organization and Professional Presentation	Proposal is exceptionally well-organized and professionally presented, with clear structure and logical flow.	Proposal is well-organized and professionally presented, with minor structural issues.	Proposal is somewhat organized and professionally presented, with noticeable structural issues.	Proposal lacks organization and professional presentation, with significant structural issues.	Proposal is poorly organized and unprofessionally presented, with major structural issues.
Table of Contents	Complete and accurate Table of Contents, with clear and logical listing of sections.	Mostly complete and accurate Table of Contents, with minor inaccuracies or missing sections.	Table of Contents is present but has noticeable inaccuracies or missing sections.	Table of Contents is present but significantly inaccurate or incomplete.	Table of Contents is missing or completely incorrect.
Body of Proposal	Body of proposal is complete, addresses all required components, and is presented clearly and coherently.	Body of proposal is mostly complete and addresses most required components, with minor clarity issues.	Body of proposal is somewhat complete, addresses some required components, with noticeable clarity issues.	Body of proposal is incomplete, addresses few required components, with significant clarity issues.	Body of proposal is largely incomplete, addresses very few required components, with major clarity issues.
Formatting	Consistently follows all formatting guidelines, including font, spacing, justification, headers, and footers.	Mostly follows formatting guidelines, with minor deviations.	Follows some formatting guidelines, with noticeable deviations.	Poor adherence to formatting guidelines, with significant deviations.	Does not follow formatting guidelines.