

Charles Malon Silva Rocha - 2021376

**Cloud Services**

**Designing and Implementing a Proof-of-Concept Cloud Solution Based on a Client’s Needs**

**CA2**

**Dublin**

**2024**

**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission.*

|  |  |
| --- | --- |
| **Module Title:** | Cloud Services |
| **Assessment Title:** | Designing and Implementing a Proof-of-Concept Cloud Solution  Based on a Client’s Needs |
| **Lecturer Name:** | Michael Weiss |
| **Student Full Name:** | Charles Malon Silva Rocha |
| **Student Number:** | 2021376 |
| **Assessment Due Date:** | 22nd November2024 |
| **Date of Submission:** | 08th December 2024 |

Below you can access the progress of this assignment:

**<https://github.com/CharlesMalonRocha/Could-Services-CA2>**

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Contents

[Introduction 4](#_Toc9358)

[TASK 1: Translating Business Requirements into an AWS Cloud Solution 5](#_Toc8356)

[TASK 2: Designing the Cloud Architecture for High Availability 6](#_Toc27587)

[TASK 3a: Identifying and Resolving Existing Anti-Patterns 7](#_Toc4147)

[TASK 3b: Enhancing Cost Efficiency and Resilience Using AWS Best Practices 8](#_Toc1445)

[TASK 4a: Understanding VPC and Its Components 9](#_Toc24698)

[TASK 4b: Hosting the Medi-Advice Website in a Custom VPC 10](#_Toc13358)

[TASK 5a: Accelerating Content Delivery with AWS CloudFront 11](#_Toc2922)

[Challenge Task 1: Demonstrating Content Delivery with Edge Caching 12](#_Toc20961)

[Challenge Task 2: Implementing a Highly Available Auto-Scaling Web Solution 13](#_Toc4203)

[Challenge Task 3: Automating Infrastructure Deployment with CloudFormation 14](#_Toc31662)

[Conclusion 15](#_Toc28465)

[References 16](#_Toc14070)

Introduction

Cloud computing has changed how businesses manage their IT systems, making it easier, cheaper, and more reliable to handle growing demands. This report focuses on creating a cloud solution for Medi-Advice, a medical startup in Dublin. Medi-Advice helps patients and doctors connect through online consultations, prescription transfers, and payment services for users in Ireland and North America. Right now, their on-premises setup struggles with issues like slow performance, limited scalability, and downtime, making it hard to keep up with growing user needs.

To fix these problems, this report suggests moving to Amazon Web Services (AWS). The solution will use services like EC2, S3, CloudFront, and VPC to create a system that is fast, scalable, and cost-effective. This report explains how these AWS tools can solve the company’s issues, improve performance, and support global users better.

By addressing Medi-Advice’s current challenges, this project will show how AWS can help build a stronger, more efficient system for the company’s needs. It also includes steps for implementation, diagrams, and tips on saving costs while improving performance.

TASK 1: Translating Business Requirements into an AWS Cloud Solution

The proposed technical solution was designed to address Medi-Advice’s current infrastructure challenges while meeting their specific needs for scalability, global accessibility, reliability, and cost efficiency. As a growing medical startup, Medi-Advice faces issues such as slow performance during high traffic, regional latency affecting U.S. users, and limited resilience in case of server failures. These problems can lead to poor user experiences and disruptions in critical medical services.

By leveraging AWS services, the solution provides a flexible and modern infrastructure capable of handling these demands. Services like Amazon EC2, Auto Scaling, and Elastic Load Balancer ensure that the system can scale dynamically with user demand, maintaining high availability. AWS CloudFront addresses global latency by caching content closer to users, while Amazon RDS and VPC enhance reliability, security, and disaster recovery capabilities. Importantly, this solution incorporates AWS’s cost-effective tools, allowing Medi-Advice to optimize expenses without compromising performance (Amazon.com, 2024, Padhy and Patra, 2013, AWS, 2019).

This solution was chosen because it not only solves Medi-Advice’s current issues but also prepares the company for future growth, ensuring a stable, secure, and scalable platform for their services.

#### ****Key AWS Services and Their Roles****

#### ****Amazon EC2 (Elastic Compute Cloud):****

* **Purpose:** Host scalable web, app, and database servers.
* **Benefits**: Elasticity for demand changes, ability to automate scaling using Autoscaling Groups.

#### ****Amazon S3 (Simple Storage Service):****

* **Purpose:** Store brochures and other static files.
* **Benefits**: Durable storage, cost-effective for large files, and integration with CloudFront for global delivery.

#### ****AWS CloudFront (Content Delivery Network):****

* **Purpose:** Improve performance for global users by caching content at edge locations.
* **Benefits**: Reduced latency, improved access speeds for brochures.

#### ****AWS VPC (Virtual Private Cloud):****

* **Purpose:** Securely isolate Medi-Advice’s network with public and private subnets.
* **Components:** Internet Gateway, NAT Gateway, Security Groups, and NACLs.
* **Benefits**: Enhanced security and network management.

#### ****Amazon RDS (Relational Database Service):****

* **Purpose:** Host the SQL Server database.
* **Benefits**: Automatic backups, cross-region replication for disaster recovery.

#### ****Elastic Load Balancer (ELB)****:

* **Purpose:** Distribute traffic across servers in multiple Availability Zones.
* **Benefits**: Enhanced fault tolerance, improved performance.

#### ****AWS Auto Scaling****:

* **Purpose:** Automatically scale EC2 instances based on traffic.
* **Benefits**: Handles spontaneous changes in demand.

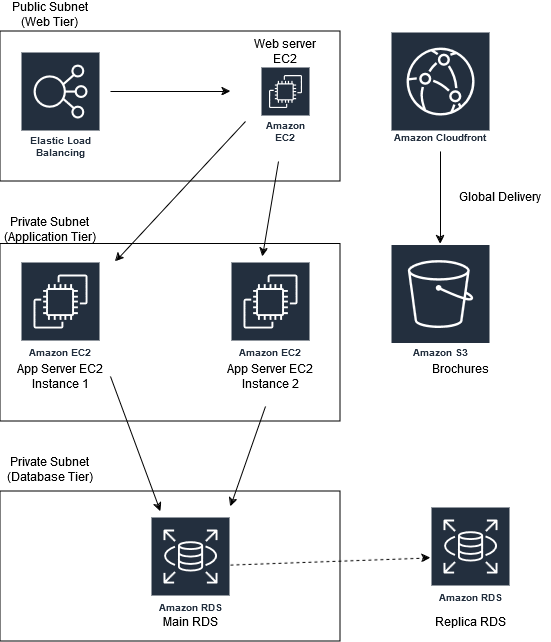
#### ****AWS IAM (Identity and Access Management)****:

* **Purpose:** Manage user permissions.
* **Benefits**: Ensures secure access to AWS resources.

#### ****How Services Address Challenges****

* **Scalability**: EC2 with Auto Scaling dynamically adjusts resources.
* **Global Accessibility**: CloudFront reduces latency.
* **Disaster Recovery**: RDS with cross-region replication ensures uptime.
* **Cost Optimization**: Use reserved/free-tier resources for low-demand periods (Amazon.com, 2024, AWS, 2019).

TASK 2: Designing the Cloud Architecture for High Availability

****

(Draw.io, 2024)

TASK 3a: Identifying and Resolving Existing Anti-Patterns

Research

#### ****Anti-Pattern 1: Manual Scaling****

* **Problem**: Current setup requires manual intervention to scale servers during high demand.
* **Solution**: Use AWS Auto Scaling for EC2 instances to dynamically adjust capacity.

#### ****Anti-Pattern 2: Single Region Deployment****

* **Problem**: Hosting only in Dublin increases latency for U.S. users and risks single points of failure.
* **Solution**: Deploy cross-region with failover using RDS and multi-region Load Balancers.

TASK 3b: Enhancing Cost Efficiency and Resilience Using AWS Best Practices

Research

· **Operational Excellence**:

* Automate deployments using CloudFormation or AWS CLI.

· **Security**:

* Implement IAM roles, encrypt data in S3 and RDS.

· **Reliability**:

* Use multi-AZ deployment for RDS and ELB.

· **Performance Efficiency**:

* Enable Elastic Load Balancing for traffic distribution.

· **Cost Optimization**:

* Use S3 lifecycle policies to archive old files.

· **Sustainability**:

* Scale down unused resources during off-peak hours.

TASK 4a: Understanding VPC and Its Components

Research

#### ****VPC Components****:

1. **Public and Private Subnets**:
   1. Public subnet: Hosts resources like Load Balancers accessible from the internet.
   2. Private subnet: Secures resources like databases and app servers.
2. **CIDR Blocks**:
   1. Define IP address ranges for subnets.
3. **Internet Gateway**:
   1. Connects the VPC to the internet.
4. **NAT Gateway**:
   1. Enables instances in private subnets to access the internet securely.
5. **Security Groups**:
   1. Firewall rules at the instance level.
6. **NACLs**:
   1. Firewall rules at the subnet level.
7. **VPC Peering**:
   1. Connects multiple VPCs securely.

TASK 4b: Hosting the Medi-Advice Website in a Custom VPC

Practical

TASK 5a: Accelerating Content Delivery with AWS CloudFront

Research

Challenge Task 1: Demonstrating Content Delivery with Edge Caching

Practical

Challenge Task 2: Implementing a Highly Available Auto-Scaling Web Solution

Practical

Challenge Task 3: Automating Infrastructure Deployment with CloudFormation

Practical

Conclusion

‌

‌

References

docs.aws.amazon.com. (n.d.). *What is Amazon S3? - Amazon Simple Storage Service*. [online] Available at: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#S3Features.>

(docs.aws.amazon.com, n.d.)

‌

Amazon Web Services (2019). What Is Amazon EC2? - Amazon Elastic Compute Cloud. [online] Amazon.com. Available at: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html.>

(Amazon Web Services, n.d.)

AWS (2019). *What Is an Application Load Balancer? - Elastic Load Balancing*. [online] Amazon.com. Available at: <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html.>

(AWS, 2019)

‌

AWS (2019). *What Is Amazon EC2 Auto Scaling? - Amazon EC2 Auto Scaling*. [online] Amazon.com. Available at: <https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html.>

(AWS, 2019)

Draw.io (2024). Flowchart Maker & Online Diagram Software. [online] app.diagrams.net. Available at: https://app.diagrams.net.

‌

Rimal, B.P., Jukan, A., Katsaros, D. and Goeleven, Y. (2010). Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. Journal of Grid Computing, 9(1), pp.3–26. doi:https://doi.org/10.1007/s10723-010-9171-y. (Rimal et al., 2010)

Padhy, R.P. and Patra, M.R. (2013). Architecture & Design of Affordable and Highly Available Enterprise Cloud Service. International Journal of Cloud Computing and Services Science (IJ-CLOSER), 2(2). doi:https://doi.org/10.11591/closer.v2i2.1527. (Padhy and Patra, 2013)

‌

‌‌

‌‌

‌‌‌‌

‌