

Charles Malon Silva Rocha - 2021376

**Cloud Services**

**Implementing a proof-of-concept cloud architecture**

**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:** | Implementing a proof-of-concept cloud architecture |
| **Lecturer Name:** | Michael Weiss |
| **Student Full Name:** | Charles Malon Silva Rocha |
| **Student Number:** | 2021376 |
| **Assessment Due Date:** | 31st October 2024 |
| **Date of Submission:** | 17th November 2024 |

Below you can access the progress of this assignment.

<https://github.com/CharlesMalonRocha/Cloud-Services-CA1>

**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](#_Toc9032)

[TASK 1a: S3 website hosting 5](#_Toc19400)

[TASK 1b: Research Task 6](#_Toc4767)

[TASK 2a: Application Load Balancer Configuration and discussion 7](#_Toc18441)

[TASK 2b Challenge 8](#_Toc26928)

[Challenge Task 3a 9](#_Toc25591)

[Challenge Task 3b 10](#_Toc12695)

[Proposed solution to the cloud engineering manager 11](#_Toc17611)

[Conclusion 12](#_Toc13610)

[References 13](#_Toc19339)

Introduction

This assignment focuses on designing and implementing a proof-of-concept cloud architecture for the startup company, "Clouds-Are-Us." The objective is to leverage Amazon Web Services (AWS) to set up scalable, secure, and highly available infrastructure. The tasks include hosting a static website using Amazon S3, creating a load-balanced group of Linux servers, and configuring an Auto-Scaling Group (ASG) integrated with an Application Load Balancer (ALB). By completing these tasks, the assignment demonstrates the application of cloud architectural principles, reliability, and performance efficiency while providing insights into real-world use cases and best practices.

TASK 1a: S3 website hosting

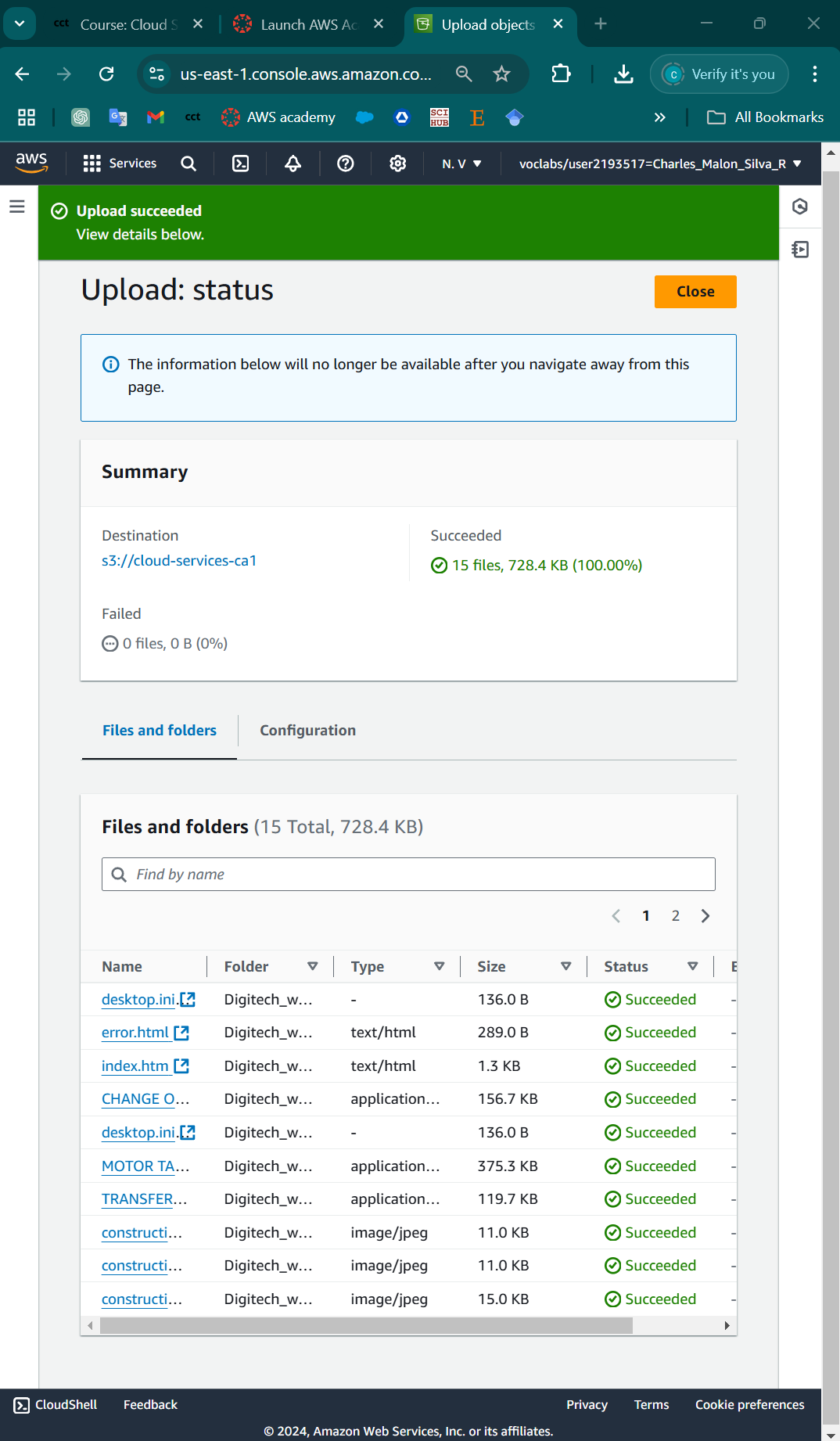


Image 1. Uploading website files into the bucket.

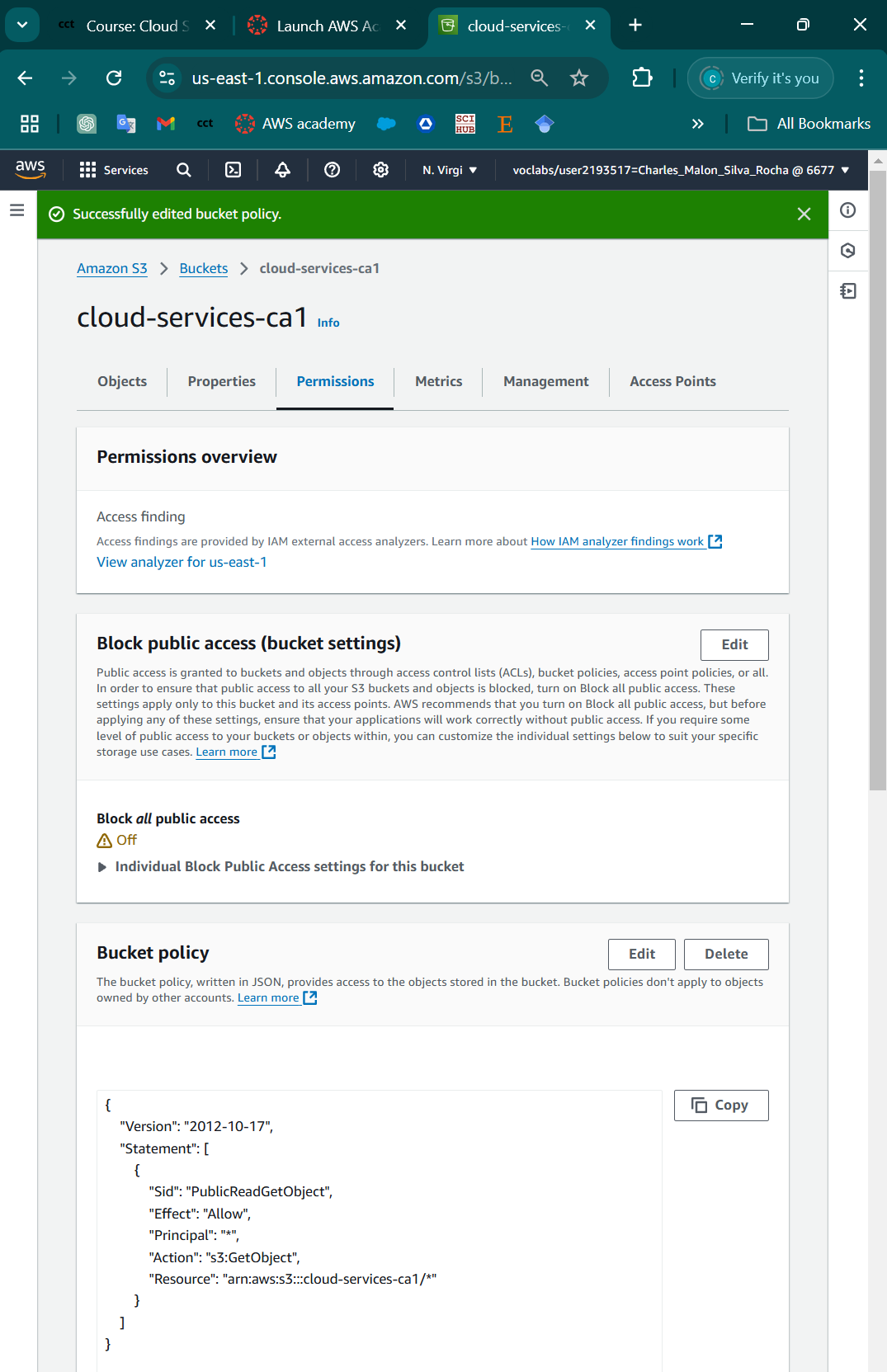


Image 2. Edited bucket policy.

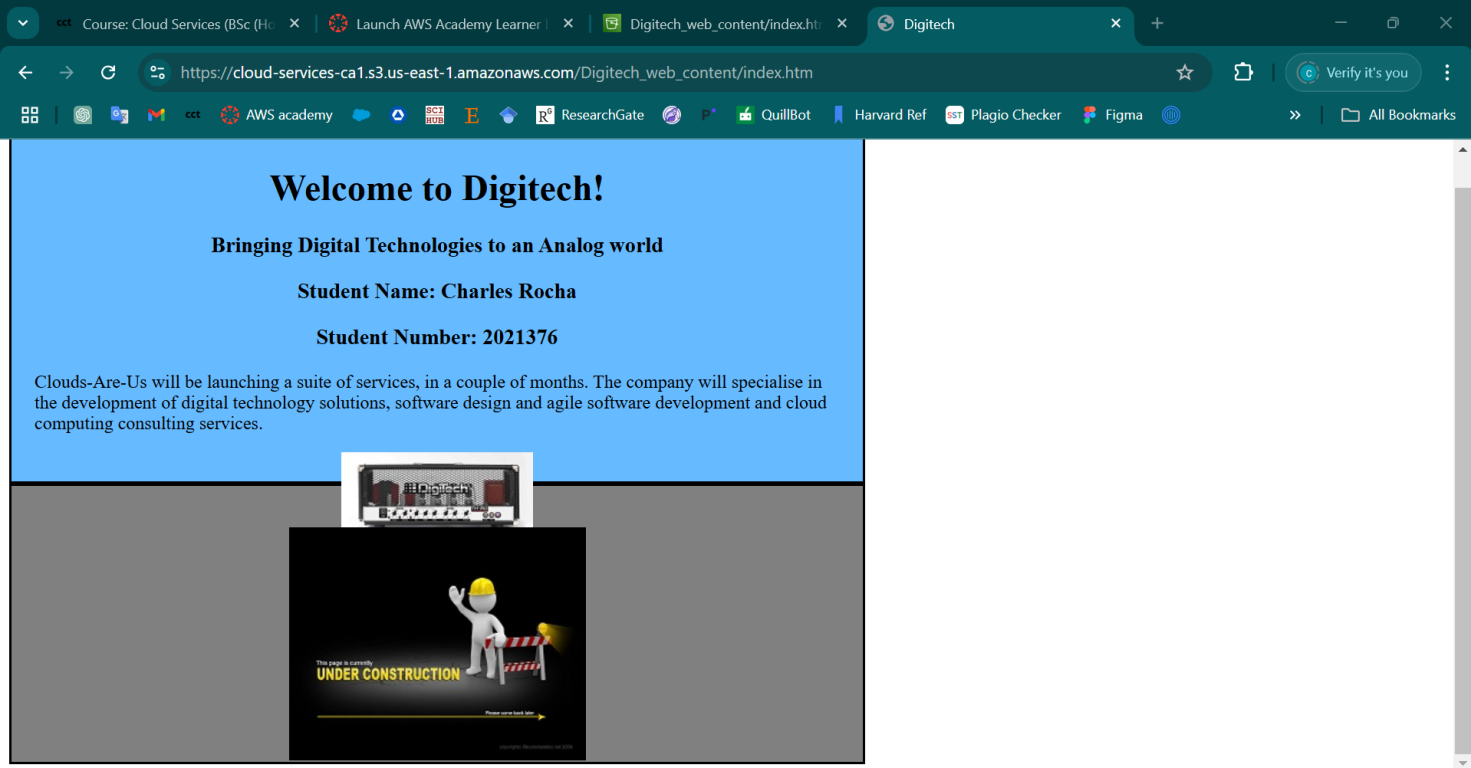


Image 3. Under Construction.

TASK 1b: Research Task

**Amazon S3 vs. EC2 with EBS: A Detailed Analysis**

Amazon Web Services (AWS) offers a range of cloud services, and among the most popular are Amazon S3 (Simple Storage Service) and EC2 (Elastic Compute Cloud) with EBS (Elastic Block Store). Both provide storage solutions, but they serve different purposes and have distinct features, use cases, and architectures. Let's break down the differences, use cases, similarities, and references (docs.aws.amazon.com, n.d. and Amazon Web Services, 2019).

### Amazon S3: Simple Storage Service

**Overview**:  
 Amazon S3 is a scalable object storage service. It stores data as objects within buckets (similar to folders), and each object is identified by a unique key. It's optimized for storing and retrieving large volumes of unstructured data and is known for its "write-once, read-many" design (docs.aws.amazon.com, n.d.).

**Key Characteristics**:

* **Object Storage**: S3 uses an object storage model, which stores data as individual objects.
* **Global Scalability**: S3 is globally accessible, and AWS handles data replication across regions.
* **Durability and Availability**: Designed for high durability (99.999999999%, or 11 9s), meaning data is highly protected.
* **Storage Classes**: Offers various storage classes (e.g., Standard, Intelligent-Tiering, Glacier) to optimize cost based on access frequency.
* **Data Accessibility**: Primarily accessed over HTTP(s) using REST APIs, allowing easy integration with various applications.

**Use Cases**:

* **Backup and Archival**: Ideal for storing backups and archived data due to its high durability and low-cost storage options (e.g., Glacier).
* **Static Content Hosting**: Perfect for hosting images, videos, and static websites, as data can be accessed over HTTP(S).
* **Data Lakes**: S3’s scalable and cost-effective nature makes it a strong foundation for data lakes used in big data analytics.
* **Content Distribution**: Integrated with CloudFront, S3 helps distribute content globally with low latency.

### Amazon EC2 with EBS: Elastic Compute Cloud with Elastic Block Store

**Overview**:  
 Amazon EC2 provides scalable computing power in the cloud. EBS is a persistent block storage system for EC2 instances, allowing data storage in block format (like a traditional hard drive) and supporting applications that require low-latency access and high IOPS (Input/Output Operations Per Second) (Amazon Web Services, 2019).

**Key Characteristics**:

* **Block Storage**: EBS is a block storage service, meaning it works at a lower level, providing raw storage for OS-level formatting and partitioning.
* **Persistent Storage for EC2**: EBS is designed specifically for EC2 instances, providing storage that persists even after instances are stopped or restarted.
* **High-Performance and Low Latency**: EBS volumes can be optimized for high IOPS, making them suitable for databases and transactional applications.
* **Snapshottable and Backups**: Users can create snapshots of EBS volumes, which are stored in S3 and can be used for backups or replicating data across instances.

**Use Cases**:

* **Databases**: EBS is suitable for database storage (e.g., MySQL, PostgreSQL) that requires high IOPS and low-latency access.
* **Transactional Applications**: Ideal for applications needing frequent read/write access to data.
* **Big Data Processing**: EC2 with EBS can be configured for compute-heavy tasks where data needs to be processed on local storage.
* **Applications Needing Persistent Disk Storage**: EBS volumes remain even when EC2 instances are stopped, unlike instance store volumes, which are ephemeral.

### Similarities Between Amazon S3 and EC2 with EBS

* **Storage Solutions**: Both S3 and EBS provide storage services, albeit with different architectures and data models.
* **Scalability**: Both services scale according to demand, although they serve different purposes.
* **Data Durability**: AWS ensures high durability for both services, making them reliable for critical data storage.
* **Backup Capabilities**: EBS volumes can be backed up using snapshots stored in S3, and S3 can archive data to lower-cost storage classes for backup purposes.

**Differences Between Amazon S3 and EC2 with EBS**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Amazon S3** | **Amazon EC2 with EBS** |
| Storage Model | Object Storage | Block Storage |
| Data Structure | Bucket and Object | Volumes attached to EC2 instances |
| Use Case Focus | Data lake, backup, and archive | High-performance storage for compute tasks |
| Data Access | HTTP-based (REST API) | Disk-level access through attached EC2 instances |
| Performance | Suitable for high-throughput | Optimized for low-latency, high IOPS |
| Durability | 99.999999999% (11 9s) durability | 99.9% - 99.999% availability, dependent on volume type |
| Data Persistence | Data is independent of compute | Persistent with EC2, linked to specific instances |
| Storage Classes | Multiple tiers (e.g., Glacier) | Types vary by performance and IOPS (e.g., gp3, io2) |

TASK 2a: Application Load Balancer Configuration and discussion

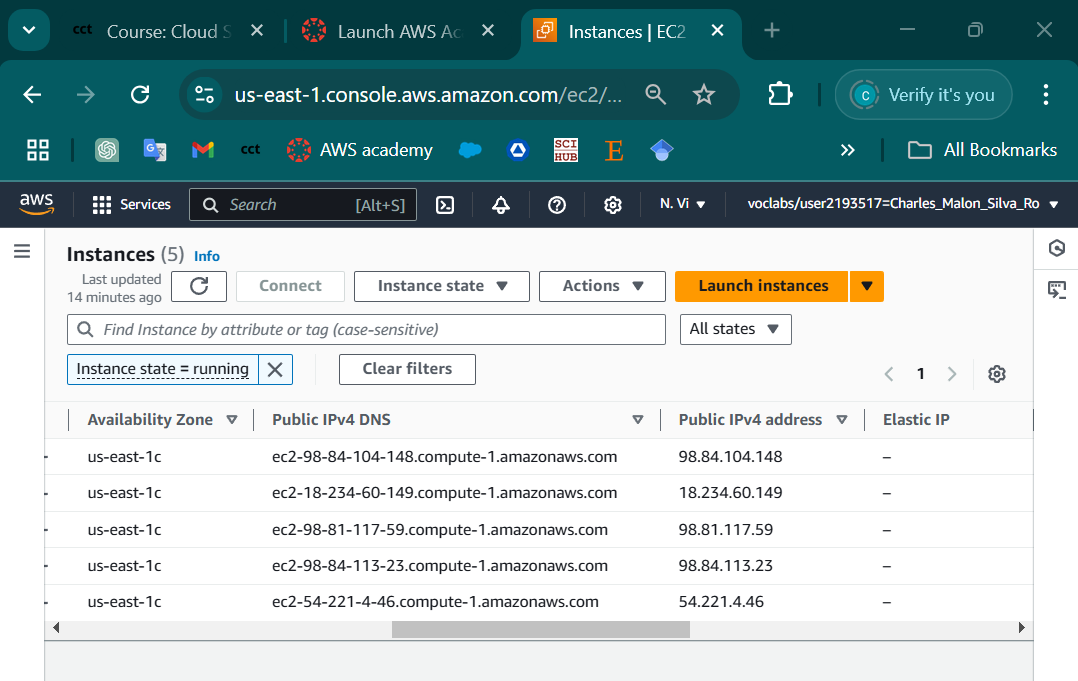


Image 4. Instances IP’s.

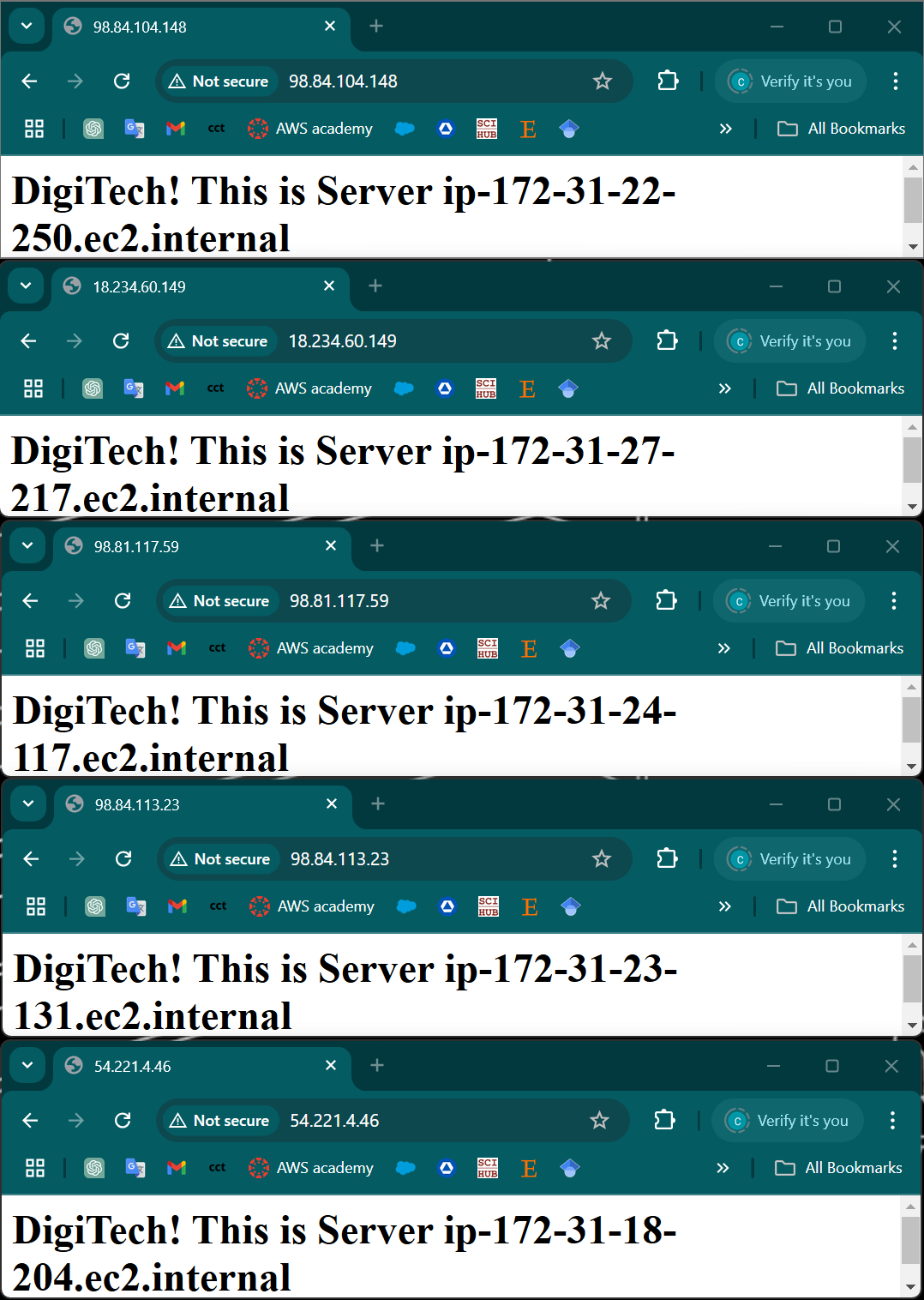


Image 5. Verifying the servers.

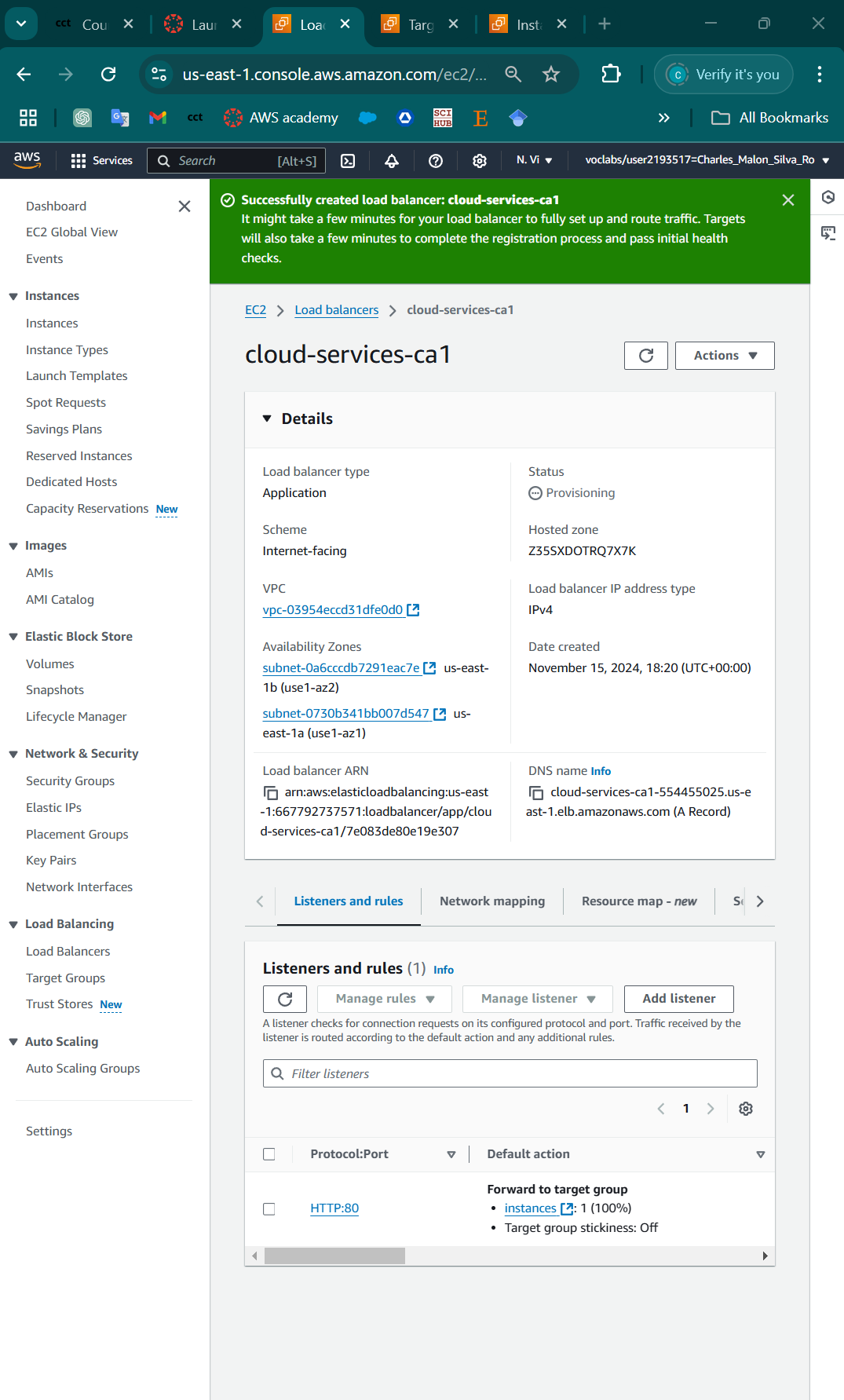


Image 6. Load Balancer created

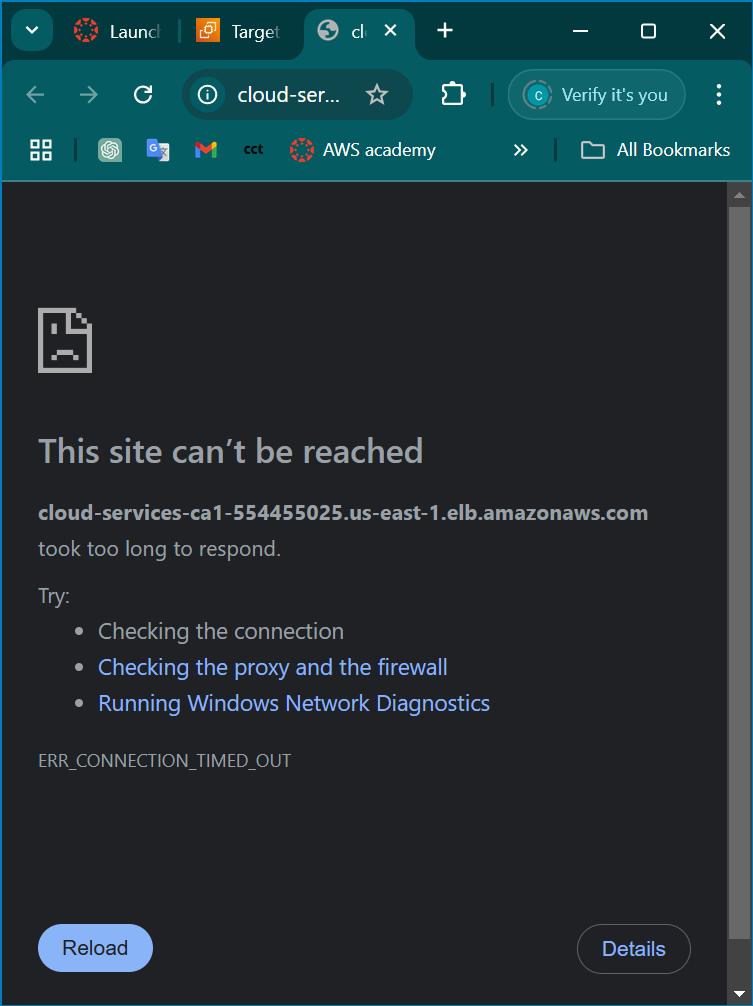


Image 7. Testing Load Balancer

TASK 2b Challenge

Challenge Task 3a

Challenge Task 3b: Proposed solution to the cloud engineering manager

### Explanation of the DigiTech ALB and Its Benefits

The **DigiTech Application Load Balancer (ALB)** is a key component in the cloud architecture, providing smart routing and distribution of incoming traffic across multiple EC2 instances. The ALB operates at application layer of the OSI model, enabling it to make advanced routing decisions based on HTTP headers, request paths, or hostnames. This setup ensures that DigiTech's website remains highly accesible, fault-tolerant, and scalable (AWS, 2019).

#### ****How the DigiTech ALB Works****

* The ALB routes traffic to healthy EC2 instances in its target group, using health checks to monitor the status of each instance.
* It dynamically balances traffic, ensuring no single instance becomes overloaded, which optimises performance and availability.
* The ALB can be integrated with an **Auto-Scaling Group (ASG)** to automatically adjust the number of instances based on demand, providing a seamless user experience during traffic surges.

### ****Benefits of the DigiTech ALB****

1. **High Availability**: The ALB ensures uninterupted service by distributing traffic across multiple instances, even if some instances fall.
2. **Scalability**: When combined with ASG, the ALB supports scaling the infrastructure up or down based on traffic, ensuring cost-efficiency.
3. **Fault Tolerance**: Instances that fail health checks are automatically excluded from traffic, maintaining a reliable environment.
4. **Enhanced Performance**: Intelligent routing ensures traffic is directed to the least loaded and healthiest instances.

### ****Architectural Design Principles (AWS Well-Architected Framework)****

The **AWS Well-Architected Framework** provides guidelines for building secure, high-performing, resilient, and efficient cloud architectures. Two of its key pillars are applied in this solution:

#### ****1. Reliability****

* **Definition**: Ensuring a system can recover from disruptions and automatically scale resources to meet demands.
* **Application**: By combining the ALB with the ASG:
  + The ALB continuously monitors the healthy of instances, routing traffic only to healthy ones.
  + The ASG replaces failed instances automatically, ensuring consistent availability.

#### ****2. Performance Efficiency****

* **Definition**: Using computing resources effectively to fulfill system requirements while optimizing performance and cost.
* **Application**:
  + The ALB distributes traffic efficiently across instances, reducing response times and preventing overload.
  + The ASG adjusts the number of running instances dynamically based on trafic patterns, ensuring optimal performance during high demand and cost-saving during low demand (AWS, 2019).

Conclusion

The integration of the ALB and ASG in DigiTech’s cloud architecture not only enhances the website's reliability and performance but also aligns with cloud computing best practices. By leveraging these components, DigiTech can ensure a highly available, fault-tolerant, and scalable environment to meet business needs effectively. Unfortunately for a reason that I was capable to identify the reason why the application didn't work as you can see in the image 5, and I wasn’t to complete all of the challenges. All the information to do this assignment I retrieve from AWS user’s guides.

‌

‌

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.

‌

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.

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). *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.

‌

‌‌

‌‌‌‌

‌

‌

‌

‌

‌