National College of Ireland

Designing a Web-Based Cloud Storage Management System to Optimize Data Retrieval, Storage Efficiency, Security, and Accessibility

Research Project
MSc Cloud Computing

RAJU GUMPULA
Student ID: x23220708

School of Computing
National College of Ireland

Supervisor: Abubakr Siddig

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	RAJU GUMPULA
Student ID:	X23220708
Programme:	MSc in Cloud Computing **Year:** 2024
Module:	Research Project
Supervisor:	Abubakr Siddig
Submission Due Date:	12/12/2024
Project Title:	Designing a Web-Based Cloud Storage Management System to Optimize Data Retrieval, Storage Efficiency, Security, and Accessibility
Word Count:	405 **Page Count** 4

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Raju Gumpula

Date: 12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Raju Gumpula
X23220708

1 Introduction

This configuration manual helps readers to analyse the libraries and software used for the project. The library and software used in the project helps in the development procedure.

2 Visual Studio Code

This project is developed on VS Code which is a multipurpose software for developing projects. It has intuitive user interface which makes it the best choice for developing this project.

3 Amazon Web Services (AWS)

This project is deployed on AWS which is a cloud storage platform. AWS is provided by Amazon to its users to deploy their projects on cloud and also provides cloud related services.

4 Bruno API Client

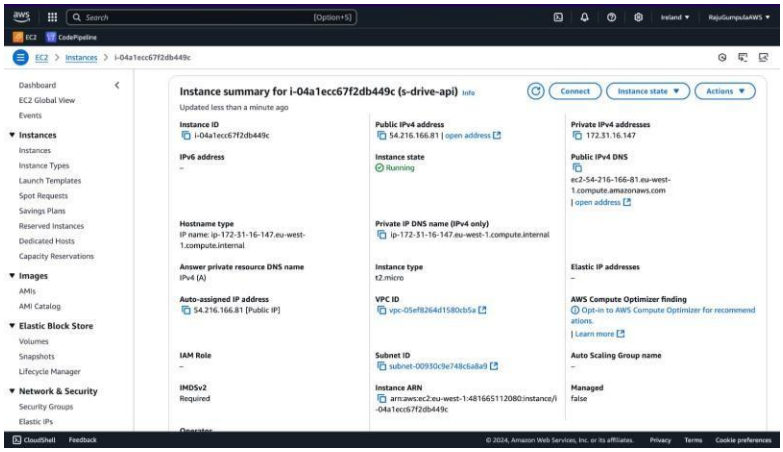
Bruno API Client is, easy to use, and complete tool for testing and debugging APIs supporting REST, GraphQL and WebSocket endpoints (Usebruno, 2024). It makes building APIs easy through features like collections, environment variables and real time response previews.

5 Summary Version of Software

Software Name	**Version**	**Download Link**
Visual Studio Code	1.95.1	https://code.visualstudio.com/download
Node.js	22.7.0	Node.js
Bruno API Client	1.36.1	https://www.usebruno.com/

6 Implementation

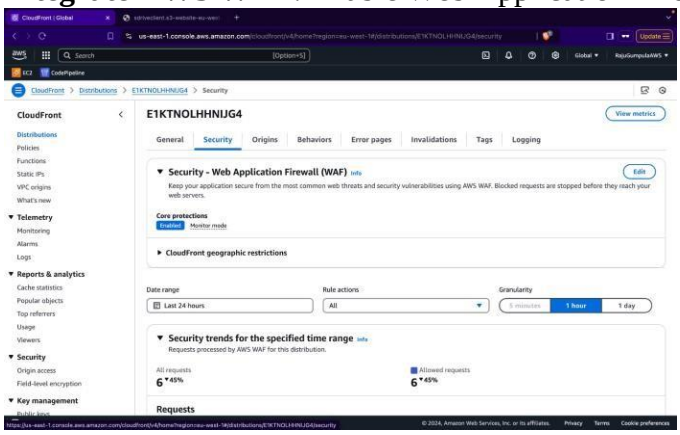
Set Up AWS EC2 Instance: Launch and configure an EC2 instance to host the backend.



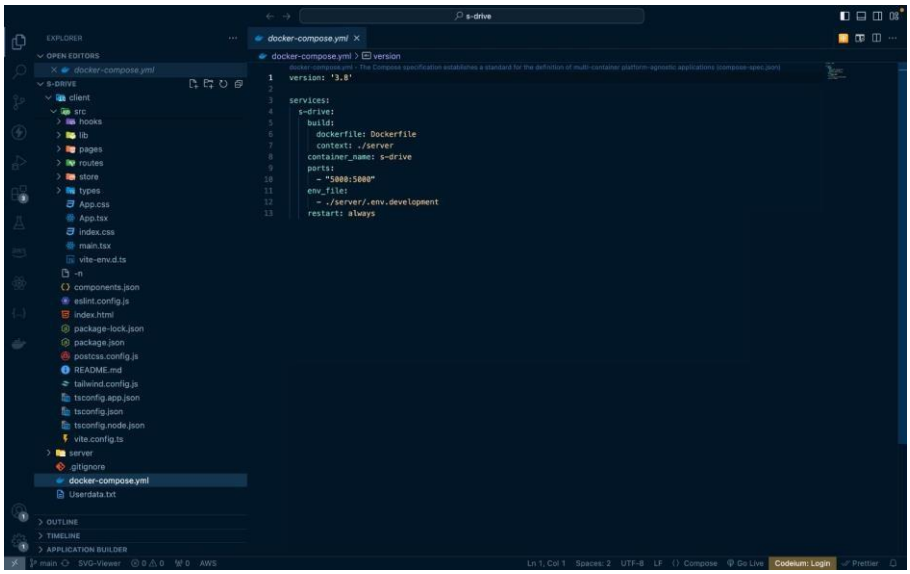
Create S3 Buckets: Set up two S3 buckets for file storage and static website hosting.

Implement CloudFront: Configure CloudFront distributions for caching and content delivery.

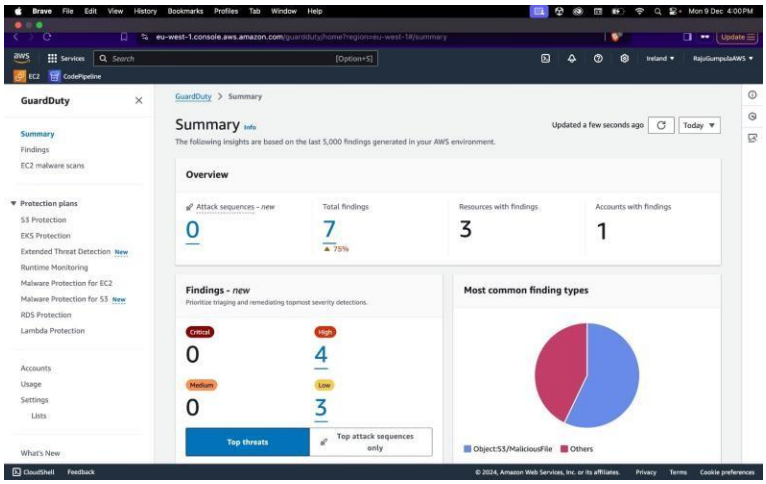
Integrate AWS WAF: Enable Web Application Firewall (WAF) for security against threats.



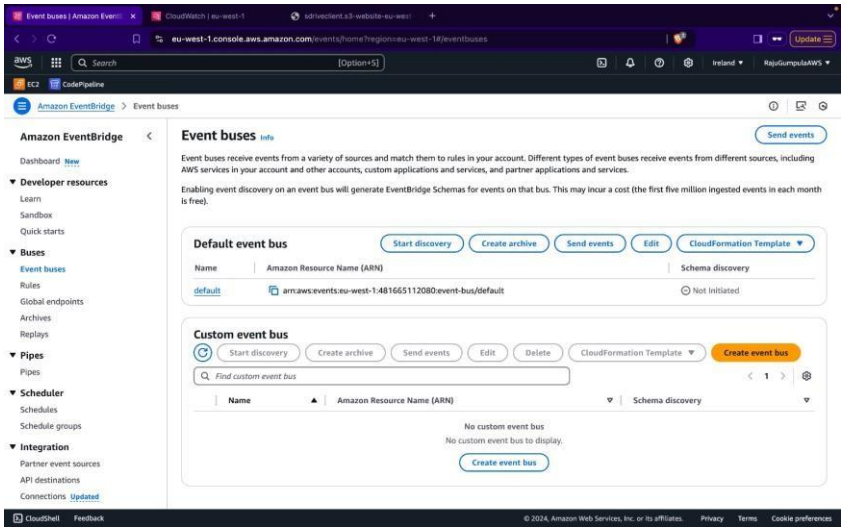
Develop Backend Services: Write APIs using Node.js to handle file uploads, metadata, and scanning.



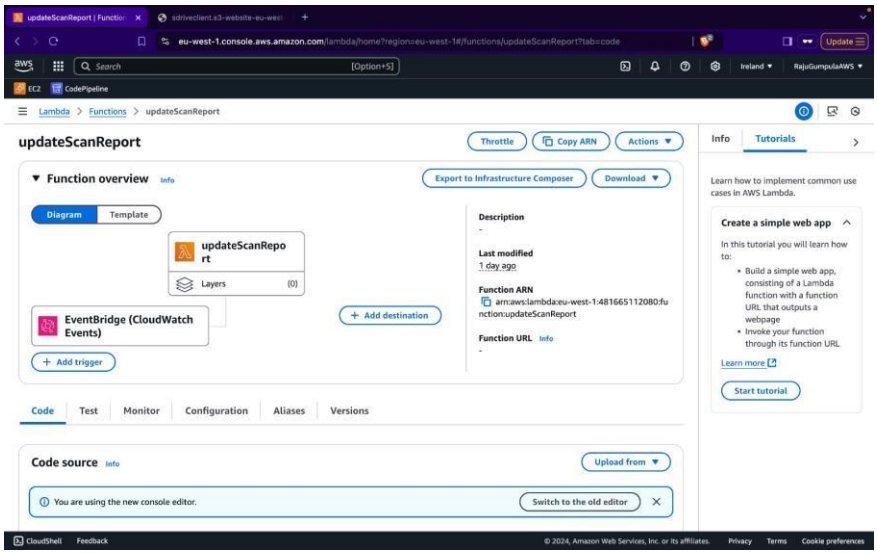
Deploy MongoDB: Configure MongoDB on EC2 to store file metadata and scanning results.
Implement GuardDuty: Set up GuardDuty to scan files for potential threats.



Configure EventBus: Link GuardDuty events to EventBus for handling malicious file detection.



Write Lambda Functions: Develop functions to process GuardDuty events and update metadata.



Enable CloudWatch Monitoring: Set up CloudWatch to monitor logs and system metrics.
Test End-to-End Workflow: Validate file upload, metadata storage, GuardDuty scanning, and event handling.

7 Evaluation

- **Performance Optimization**: The integration of CloudFront reduced latency and improved the application's speed by caching frequently accessed files.
- **Security Enhancements**: AWS WAF effectively prevented DoS attacks, while GuardDuty detected and flagged anomalies in uploaded files, ensuring a secure file-sharing environment.
- **Scalability and Storage Efficiency**: S3 buckets provided cost-effective and scalable storage, accommodating high volumes of user uploads seamlessly.
- **Monitoring and Analytics**: CloudWatch logs and metrics enabled real-time monitoring, ensuring quick detection and resolution of system issues.

References

Usebruno. 2024. *Re-Inventing the API Client.* [Online]. Available at: <https://www.usebruno.com/ >.