**Glidion Company – Business Requirement Document (BRD)**

**Project:** Website Hosting Migration to AWS Services

**Prepared By:** Gaurav Singaria- Business Analyst

**Date:** 26-Feb-2025

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VERSION** | **APPROVED BY** | **REVISION DATE** | **DESCRIPTION OF CHANGE** | **AUTHOR** |
| 1.0 | <Senior Business Analyst> | 22/03/25 | -- | GAURAV |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TABLE OF CONTENTS**

1.Executive Summary 3

2.Business Objectives 3

3.Scope of the Project 3

In-Scope

Out-of-Scope

4.Stakeholder Information 4

5.Functional Requirements 4

6.Non-Functional Requirements 5

7.Use Cases 5

8.Assumptions & Constraints 6

9.Acceptance Criteria 7

10. Risks & Dependencies 7

11. Data Flow & Process Diagrams 8

12. Conclusion 9

13.Glossary 10

14.References 10

15.Appendix 11

**1. Executive Summary**

Glidion Company plans to **migrate its current website hosting** to **Amazon Web Services (AWS)** to improve **scalability, performance, and security**. This BRD outlines the business objectives, functional and non-functional requirements, scope, and acceptance criteria for the website hosting migration project.

**2. Business Objectives**

The primary objectives of this migration are:

* **Enhanced Website Performance:** Faster load times and improved user experience.
* **Increased Scalability:** Handle traffic spikes efficiently using AWS Auto Scaling and Elastic Load Balancing (ELB).
* **Improved Security:** Use AWS security features such as WAF, Shield, and encryption.
* **Cost Efficiency:** Reduce hosting costs with AWS pay-as-you-go pricing.
* **Reliability and Uptime:** Leverage AWS’s high-availability infrastructure.

**3. Scope of the Project**

**In-Scope:**

* **Website Hosting Migration:**
  + Migrating from the current hosting provider to AWS EC2, S3, and RDS.
* **Domain and DNS Configuration:**
  + Pointing the domain to AWS Route 53.
* **Backup and Data Integrity:**
  + Full back up before migration.
  + Verification of data integrity post-migration.
* **Security Enhancements:**
  + Enabling SSL/TLS certificates.
  + Implementing AWS WAF and Shield for DDoS protection.
* **Performance Optimization:**
  + Enabling AWS CloudFront for content delivery.
  + Using Amazon Relational Database Services for optimized database performance.

**Out-of-Scope:**

* **Website Redesign:** The project only focuses on migration, not a redesign.
* **Third-Party App Integrations:** No external integrations or major code modifications.

**4. Stakeholder Information**

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Responsibility** |
| CEO | Executive Sponsor | Final approval and project oversight. |
| IT Team | Technical Engineers | Execute migration and verify stability. |
| Web Developers | Developers | Ensure website functionality post-migration. |
| Business Analyst (Me) | Business Analyst | Document requirements and track progress. |
| AWS Solutions Architect | Technical Consultant | Provide AWS infrastructure expertise. |

**5. Functional Requirements**

**A. Infrastructure Setup:**

* Create AWS EC2 instances to host the website.
* Configure Amazon RDS for the website’s database.
* Set up Amazon S3 for storing static content.
* Enable CloudFront CDN for content delivery and caching.

**B. Domain and DNS Configuration:**

* Migrate domain to AWS Route 53.
* Configure DNS settings to point to the new AWS environment.

**C. Security and Compliance:**

* Enable **SSL/TLS certificates** for secure data transmission.
* Configure **AWS WAF and Shield** for protection against DDoS attacks.
* Set up IAM roles and policies for secure access control.

**D. Backup and Data Integrity:**

* Take a full backup of the website and database before migration.
* Verify data integrity and consistency post-migration.

**E. Performance Optimization:**

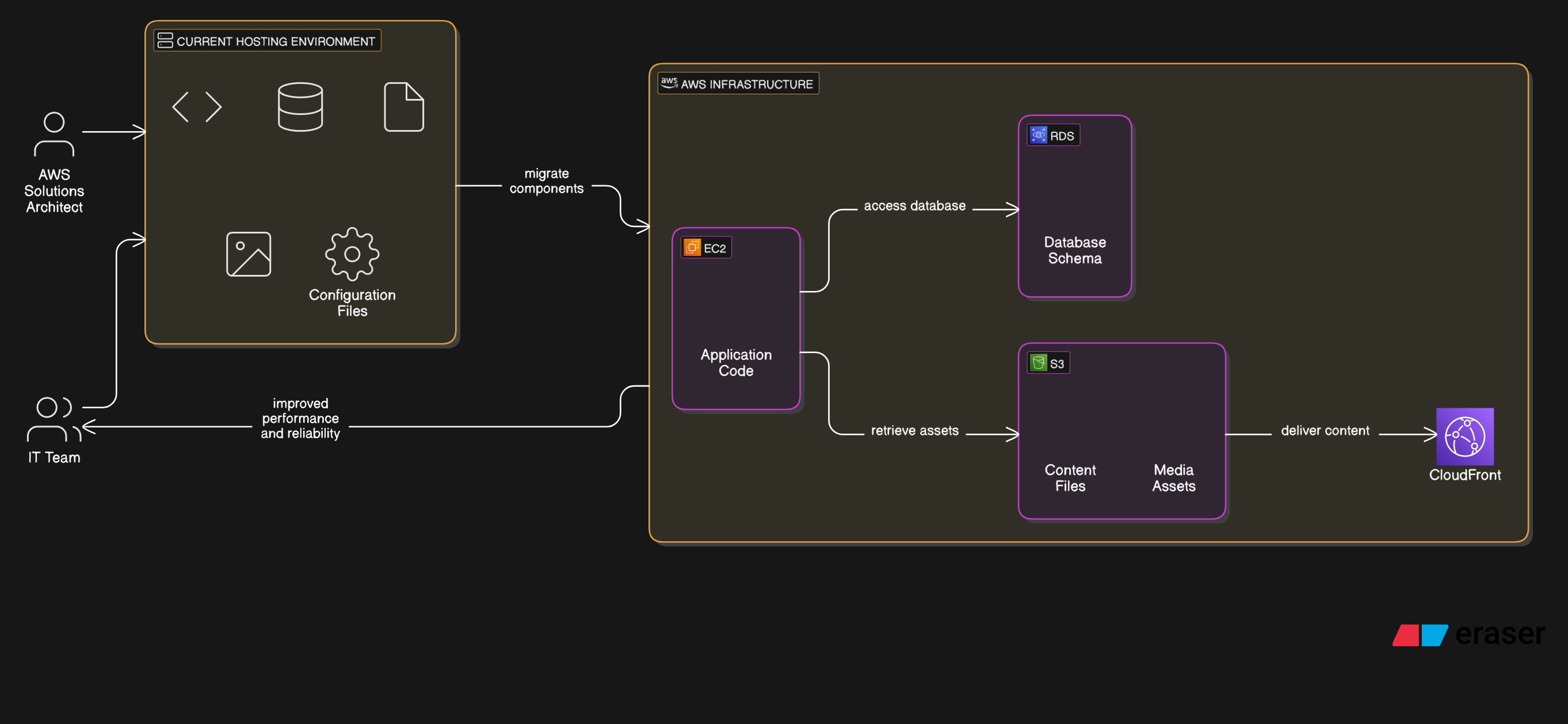
* Use **CloudFront CDN** for faster content delivery.
* Implement **Auto Scaling** and ELB for load balancing.
* Use **Amazon Relational Database Services** for optimized database queries.

**6. Non-Functional Requirements**

* **Performance:** Page load times should be under **2 seconds**.
* **Availability:** Ensure **99.9% uptime** post-migration.
* **Scalability:** The AWS environment should auto-scale during traffic spikes.
* **Security:** Implement **DDoS protection, encryption, and secure access controls**.
* **Cost Efficiency:** Optimize AWS resources to maintain cost-effectiveness.
* **Reliability:** Automated backups and failover mechanisms should be enabled.

**7. Use Cases**

**Use Case 1: Website Migration Execution**

* **Actors:** IT Team, AWS Solutions Architect
* **Description:** Migrate the website and database from the current hosting provider to AWS.
* **Outcome:** Website runs on AWS infrastructure with improved performance and reliability.
* 

**Use Case 2: DNS Configuration**

* **Actors:** IT Team
* **Description:** Configure the DNS settings on Route 53 to point the domain to AWS servers.
* **Outcome:** Website resolves to the new AWS environment without downtime.

**Use Case 3: Performance and Security Testing**

* **Actors:** IT Team, Business Analyst
* **Description:** Test website performance, page load speed, and security features.
* **Outcome:** Website meets performance, security, and uptime criteria.

**8. Assumptions & Constraints**

**Assumptions:**

* All existing website data is properly backed up before migration.
* AWS services are configured with best practices.
* No major code changes are required during the migration.

**Constraints:**

* **Downtime Limitation:** Minimal downtime during migration.
* **Preferred migration process time:** After 12 AM and Process should complete before 5 AM
* **Data Integrity:** Complete data validation post-migration in next 4 hours.

**9. Acceptance Criteria**

* Website is fully operational on AWS with no data loss.
* Domain and DNS settings correctly resolve to AWS.
* Performance benchmarks (load time under **2 seconds**) are met.
* Website maintains **99.9% uptime** post-migration.
* Security measures (SSL, WAF, Shield) are implemented.

**10. Risks & Dependencies**

**Potential Risks:**

* **Data Loss:** Incomplete backups or failed data transfer.
* **Downtime:** Website downtime during DNS propagation.
* **Configuration Issues:** Incorrect AWS setup leading to security vulnerabilities.
* **Cost Overruns:** Unoptimized AWS usage leading to higher costs.

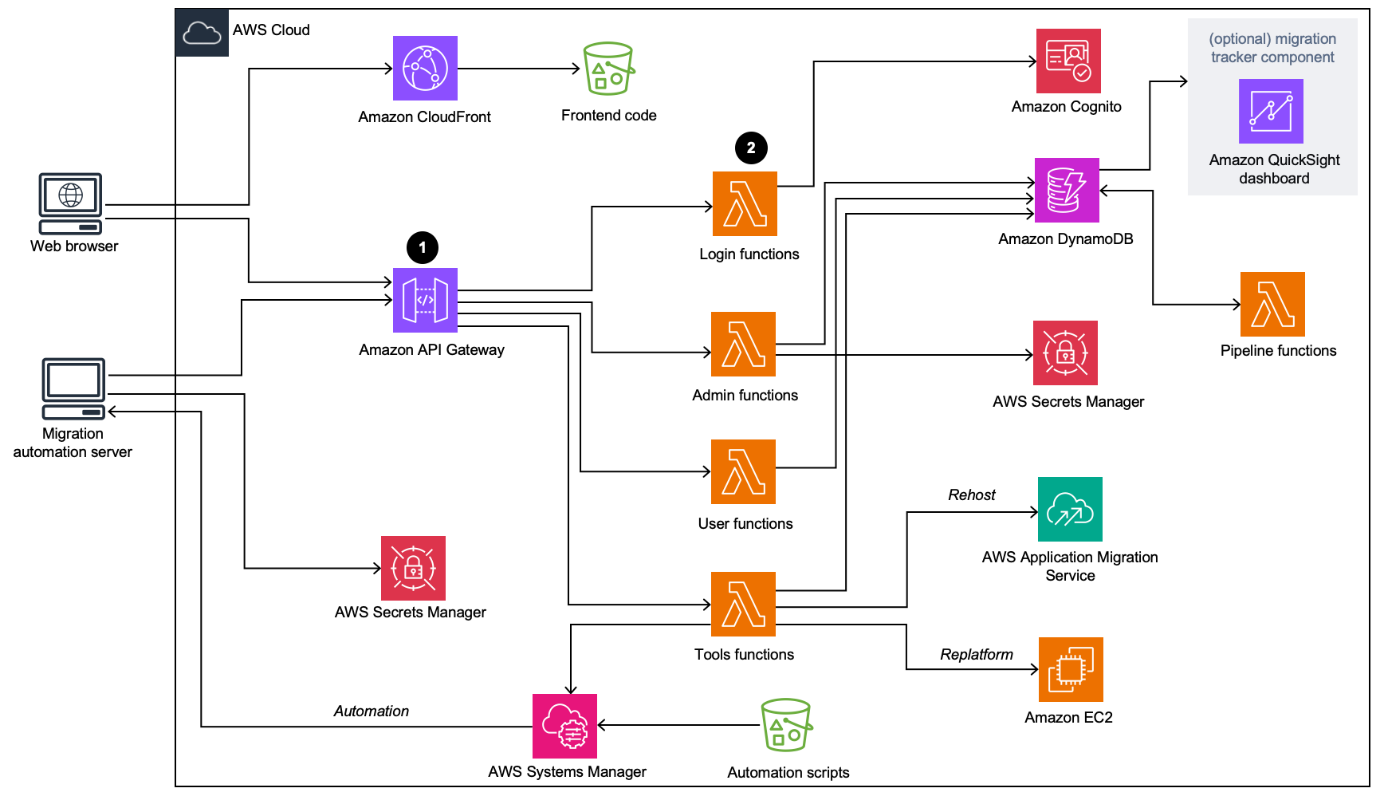
**Dependencies:**

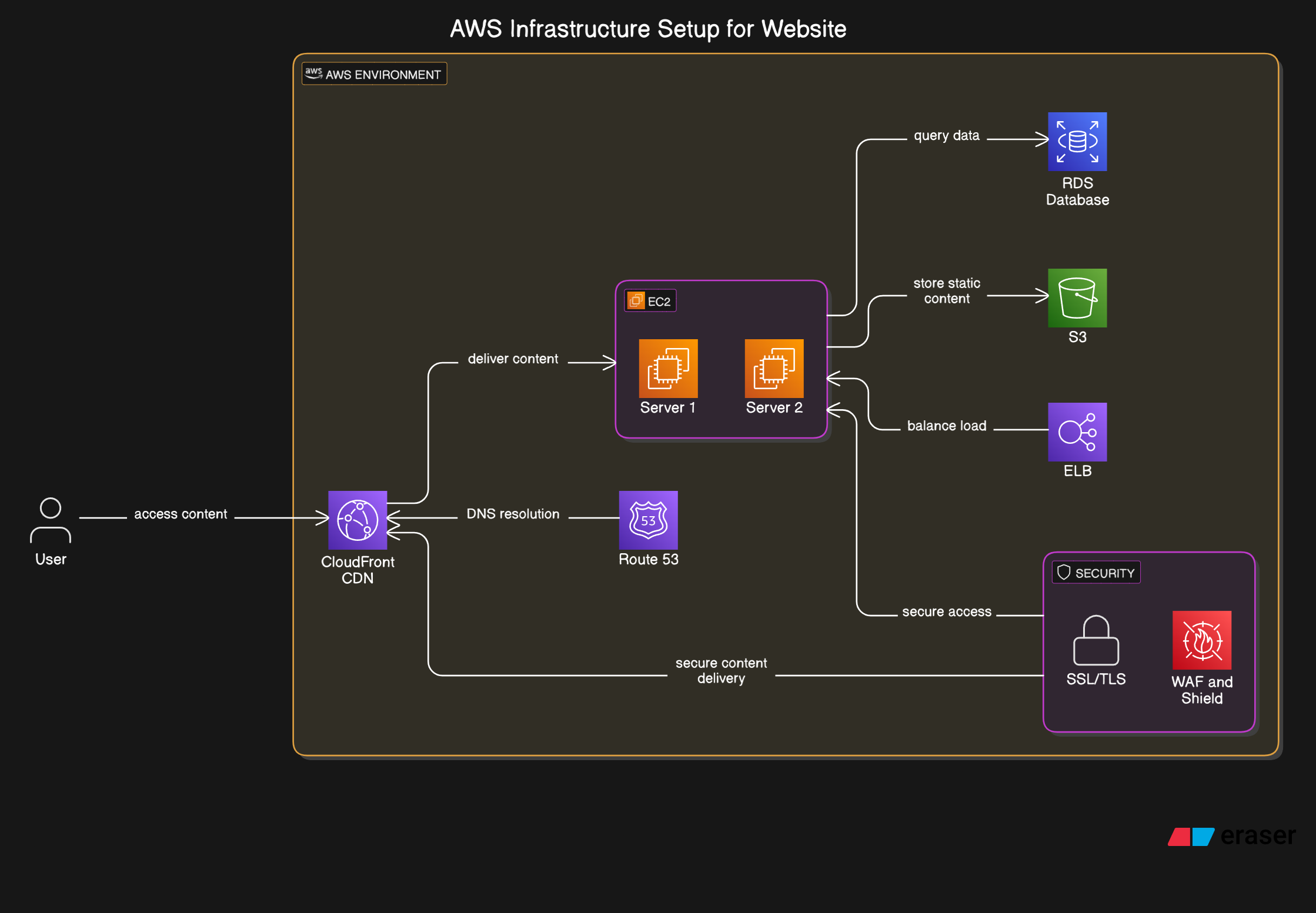
* Access to current hosting provider’s admin panel.
* AWS account with proper service permissions.
* Collaboration with IT team and AWS consultants.

**11. Data Flow & Process Diagrams**

Include process flow diagrams for website migration steps, such as backup, migration, DNS configuration, and performance testing



AWS Setup Reference -



**Conclusion**

This BRD defines the **functional, non-functional, and business requirements** for migrating Glidion’s website hosting to AWS services. It outlines the project scope, use cases, acceptance criteria, and potential risks, providing a clear roadmap for successful implementation.

**Glossary**

|  |  |
| --- | --- |
| **Term** | **Explanation** |
| Propagation | multiplication (as of a kind of living thing) in number of individuals |
| Dependencies | dependency is a task that relies on the completion of a different task |
| CDN | A content delivery network (CDN) is a group of geographically distributed servers that speed up the delivery of web content by bringing it closer to where users are. |
| ELB | Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones (AZs) |
| WAF | A web application firewall (WAF) is a firewall that monitors, filters and blocks Hypertext Transfer Protocol (HTTP) traffic as it travels to and from a website or web application. A WAF can be network based, host based or cloud based. |

**References**

1.use of A.I tools for report building

2.analysing web migration process from aws faq page

3.two process diagrams generated from eraser.io (an AI tool for generating process diagrams)

**Appendix**