# Service Rationale for Migrated Application Components

## Introduction

This document explains the architectural decisions and AWS service selections made for each component of the application. The choices are based on requirements such as automation, scalability, integration, and AWS best practices.

The sections below describe each application component, the specific AWS service chosen for migration, and the justification based on both project requirements and AWS best practices.

## Component 1: Application Server

Selected AWS Service: Amazon EC2 (Elastic Compute Cloud)

### Requirements Justification:

- \*\*Automation\*\*: EC2 instances support auto-scaling, allowing dynamic adjustment of instances based on demand.  
- \*\*Scalability\*\*: EC2 auto-scaling and Elastic Load Balancing (ELB) allow the application to scale horizontally based on workload, ensuring performance during peak times.  
- \*\*Integration\*\*: EC2 integrates seamlessly with other AWS services like CloudWatch for monitoring, IAM for access control, and RDS for database connectivity.  
- \*\*Best Practices\*\*: EC2 aligns with AWS’s well-architected framework for compute services, providing flexibility and control over configurations.

## Component 2: Database

Selected AWS Service: Amazon RDS (Relational Database Service) for MySQL/PostgreSQL

### Requirements Justification:

- \*\*Automation\*\*: RDS offers automated backups, patching, and snapshots to reduce management overhead.  
- \*\*Scalability\*\*: RDS supports read replicas and multi-AZ deployments, enabling high availability and scalability.  
- \*\*Integration\*\*: Easily integrates with EC2 instances and AWS Lambda functions, supporting real-time data access.  
- \*\*Best Practices\*\*: RDS adheres to AWS best practices by supporting multi-AZ deployments and encryption, enhancing security and availability.

## Component 3: Application Storage

Selected AWS Service: Amazon S3 (Simple Storage Service)

### Requirements Justification:

- \*\*Automation\*\*: Lifecycle policies automate storage management by moving objects between storage classes.  
- \*\*Scalability\*\*: S3 provides virtually unlimited storage space that can handle large volumes of data.  
- \*\*Integration\*\*: Integrates with Lambda, CloudFront, and other services for real-time processing and content delivery.  
- \*\*Best Practices\*\*: S3 offers strong data durability, access logging, and versioning, ensuring secure and reliable storage.

## Component 4: Authentication and Authorization

Selected AWS Service: AWS IAM (Identity and Access Management) and Amazon Cognito

### Requirements Justification:

- \*\*Automation\*\*: Cognito’s user management features allow automation of user access.  
- \*\*Scalability\*\*: Cognito scales to support millions of users, ensuring performance under heavy loads.  
- \*\*Integration\*\*: Works seamlessly with API Gateway and Lambda, offering secure access to backend services.  
- \*\*Best Practices\*\*: Cognito and IAM support secure authentication and permissions management following AWS’s principle of least privilege.

## Component 5: Monitoring and Logging

Selected AWS Service: Amazon CloudWatch and AWS CloudTrail

### Requirements Justification:

- \*\*Automation\*\*: CloudWatch alarms and logs automate monitoring, while CloudTrail provides API call logs for compliance.  
- \*\*Scalability\*\*: CloudWatch handles large-scale monitoring for distributed applications.  
- \*\*Integration\*\*: Integrates with EC2, RDS, Lambda, and S3 to provide a comprehensive view of application health.  
- \*\*Best Practices\*\*: CloudWatch and CloudTrail promote operational excellence by supporting real-time monitoring and auditing.

## Conclusion

The selected AWS services align with the project’s requirements and AWS best practices, ensuring a solution that is scalable, automated, and well-integrated. Each decision supports the application's long-term performance, security, and maintainability.