

Cloud Based Online Book Store.

By Santosh Kumar Paital.

Abstract

"Successfully designed and implemented a comprehensive Cloud base Online Book Store application, including CRUD Operations, and deployed it on Amazon Web Services. The back-end development was done using Spring Boot, while React was used for the front-end development. The application was deployed using AWS RDS for the database, leveraging two EC2 instances located in two different regions and an Elastic Load Balancer (ELB) to efficiently manage traffic flow. Additionally, the front-end was stored on an S3 Bucket."

Introduction

The purpose of this report is to detail the development and deployment of a Online Book Store application on Amazon Web Services (AWS).

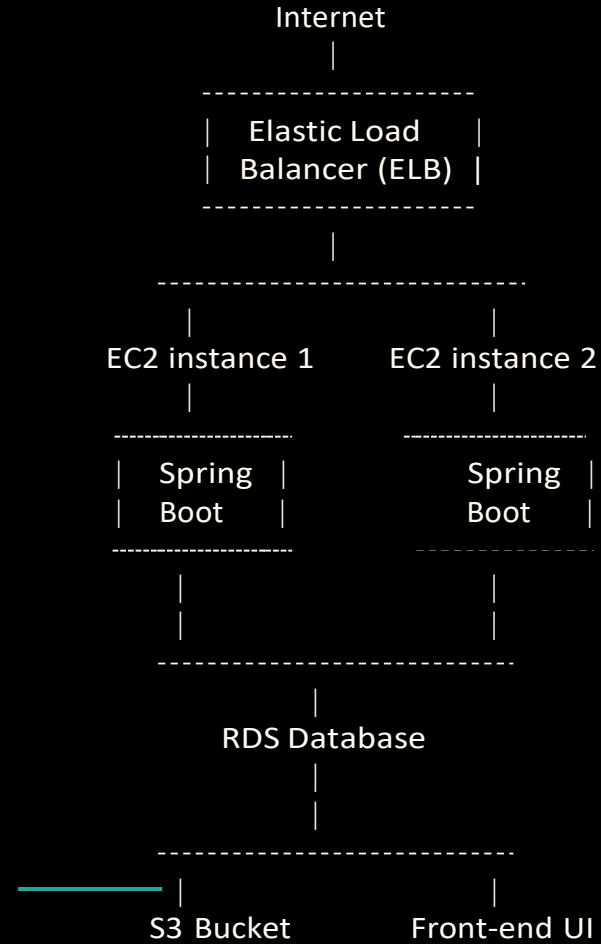
The application was designed to include CRUD Operations and leverages Spring Boot for the back-end development and React for the front-end development. This report outlines the components used in deploying the application on AWS, including AWS RDS, EC2 instances, ELB, and S3 Bucket.



Programming Tools and Technologies

- Java - for building the back-end using the Spring Boot framework.
- React - for building the front-end user interface.
- HTML and CSS - for designing the user interface and styling it.
- AWS RDS - for creating and deploying the database.
- AWS EC2 - for creating virtual servers to run the application and to store the back-end code.
- Elastic Load Balancer (ELB) - for managing traffic between the two EC2 instances.
- AWS Elastic Beanstalk or Docker - for deploying the application on the EC2 instances.
- AWS S3 - for storing the front-end UI files.
- Git - for version control and collaboration among developers.
- Integrated Development Environment (IDE) - such as Eclipse, IntelliJ, or Visual Studio Code, for writing and testing the code.

Deployment architecture



Deployment architecture:

- The diagram above illustrates the various components involved in deploying the Online Book Store application on AWS. The architecture includes an Elastic Load Balancer (ELB) that distributes traffic between two EC2 instances located in two different regions, both of which leverage the Spring Boot framework for back-end development. The database is hosted on AWS RDS, while the front-end UI is stored on an S3 bucket.

Application Development:

- The application was developed using Spring Boot for the back-end and React for the front-end development. The Spring Boot framework provides a robust and flexible environment for developing back-end applications, while React is a popular front-end library for building user interfaces.

Deployment Architecture

The application was deployed on AWS using the following architecture:

- AWS RDS for the database: AWS RDS is a managed database service that simplifies the setup, operation, and scaling of relational databases on AWS. It was used to host the database for the application.
- Two EC2 instances in different regions: The application was deployed across two EC2 instances located in two different regions. This architecture provides redundancy and improves availability.
- Elastic Load Balancer (ELB): ELB was used to manage traffic flow between the two EC2 instances. This improves the application's reliability and availability, as well as provides automatic scaling.
- S3 Bucket: The front-end of the application was stored on an S3 Bucket. S3 is a scalable object storage service that can store and retrieve any amount of data.

Conclusion

In conclusion, the development and deployment of a full-stack application on AWS was successful, leveraging Spring Boot and React for the back-end and front-end development respectively. The application was deployed using AWS RDS for the database, two EC2 instances in different regions, ELB for traffic management, and an S3 Bucket for storing the front-end. The architecture provides redundancy, improves availability, and enhances the application's reliability.

Thank You.