Design Document of BookHub

Introduction to Cloud Computing – CS 5610

Summer 2024

Team Members:

Dhiraj Bandi 700760856

Yenigalla Tejasree 700766181

Killa Saranya 700764705

Eshwar Reddy 700764930

Prof. Dr. Qixiang Pang

The University of Central Missouri.

Table of Contents

1.Overview:

Description: BookHub is designed to store and manage the inventory information of a library. The application will feature a comprehensive catalog of books, allowing librarians and users to efficiently browse, borrow, and manage books. Users can search for books based on various attributes such as title, author, genre, and availability. The app will also include features for borrowing and returning books, tracking user borrowing history, and generating reports for library management.

Rationale: The library management industry needs a centralized and efficient solution to manage diverse operations and meet the demands of its users. Our cloud-based platform will harness cutting-edge technology to ensure scalability, data security, and real-time accessibility, effectively addressing these challenges and propelling the industry toward greater success.

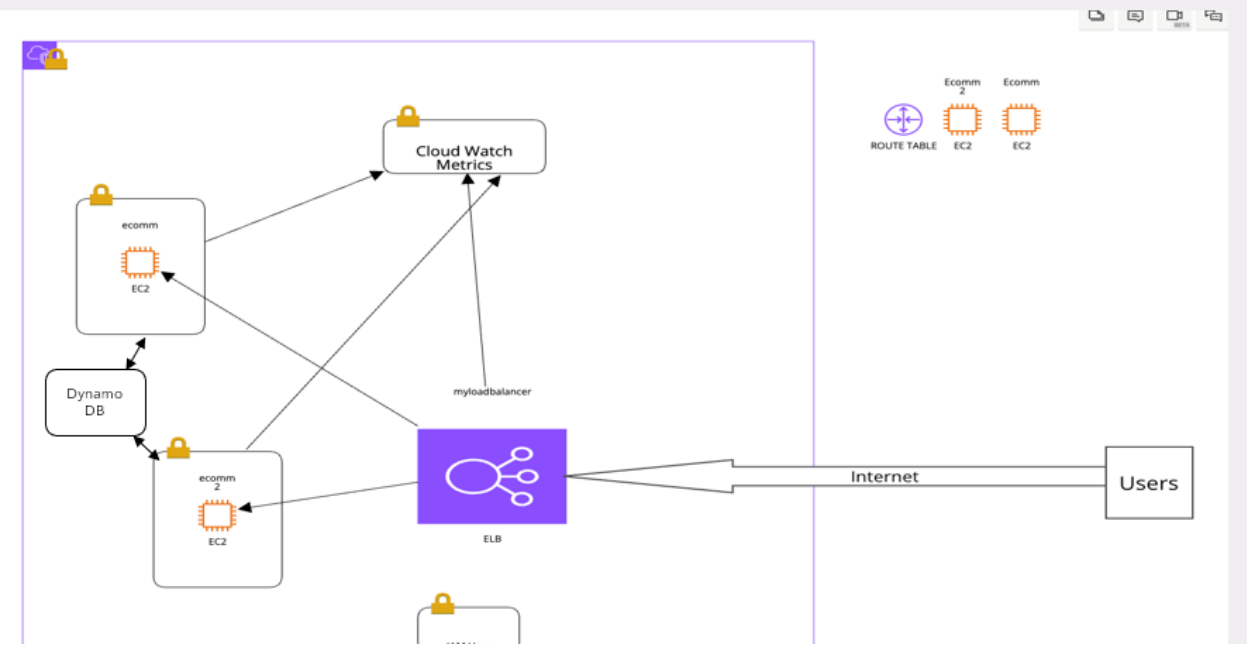
2. Cloud Services/Platforms/Tools/Frameworks:

Our project will utilize the power of Amazon Web Services (AWS) cloud infrastructure to deliver unparalleled performance, scalability, and security. We will use the following services and tools:

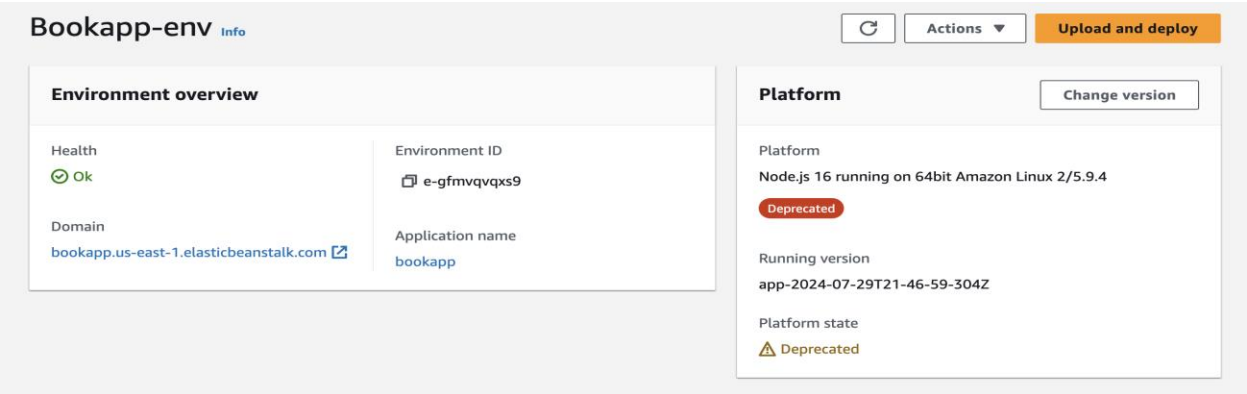
1. AWS EC2: Provides elastic compute resources, ensuring smooth performance even during peak traffic. EC2 instances will host the backend services and database servers.
2. AWS Elastic Beanstalk: An orchestration service for deploying applications, managing EC2, S3, SNS, CloudWatch, autoscaling, and load balancing.
3. AWS RDS with MySQL: Ensures a reliable and fully managed relational database service, guaranteeing data integrity and high availability. MySQL will be used to manage structured data.
4. AWS IAM: Manages secure access to AWS services and resources for users and applications.
5. AWS DynamoDB: A fully managed NoSQL database service for storing unstructured data, such as book metadata and user information.
6. AWS CloudWatch: Collects monitoring and operational data in the form of logs, metrics, and events. It provides real-time monitoring of application performance and resource utilization.
7. NodeJS: For server-side development, handling API requests, and managing application logic.
8. ReactJS: For building the user interface, ensuring a responsive and interactive experience for users.
9. HTML, CSS, Bootstrap, JavaScript: For front-end development, providing the structural and stylistic elements of the web application.

3. Architecture:

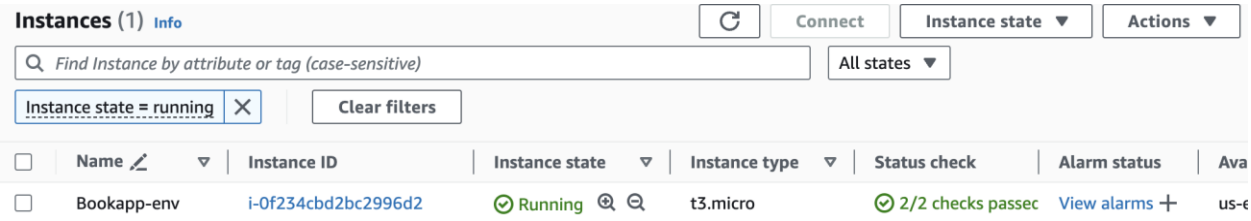
The architecture of BookHub will include the following components:



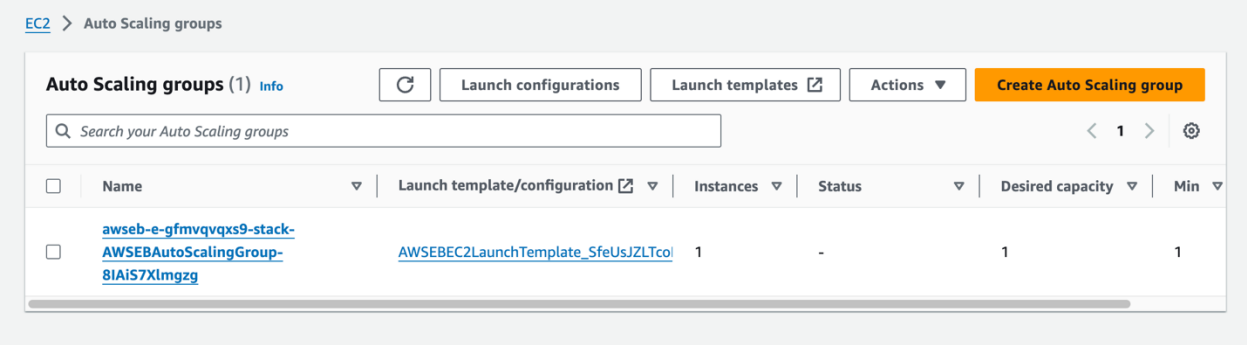
1. Elastic Beanstalk: Manages the deployment and scaling of web applications and services. It abstracts the complexity of managing the infrastructure.



2. EC2 Instances: Virtual servers in the cloud that will host the application and database servers. They can be scaled up or down based on traffic.

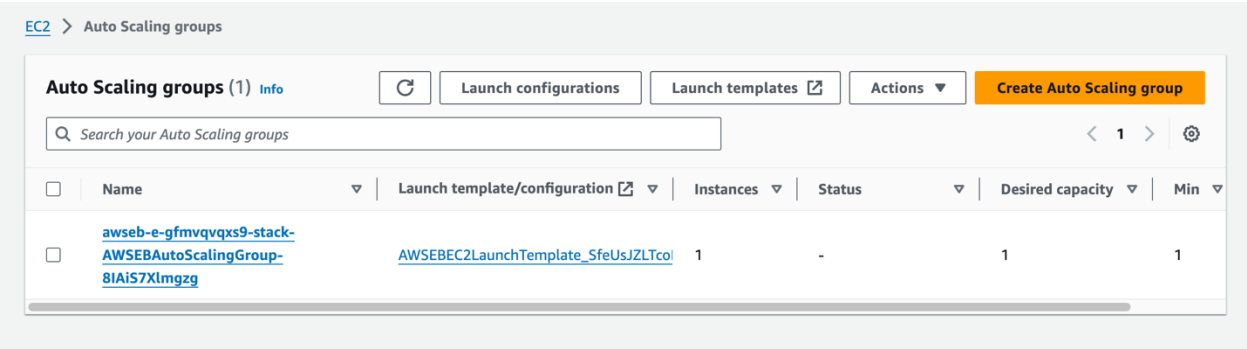


3. Auto Scaling Groups: Automatically adjusts the number of EC2 instances to handle load variations, ensuring optimal performance and cost-efficiency.

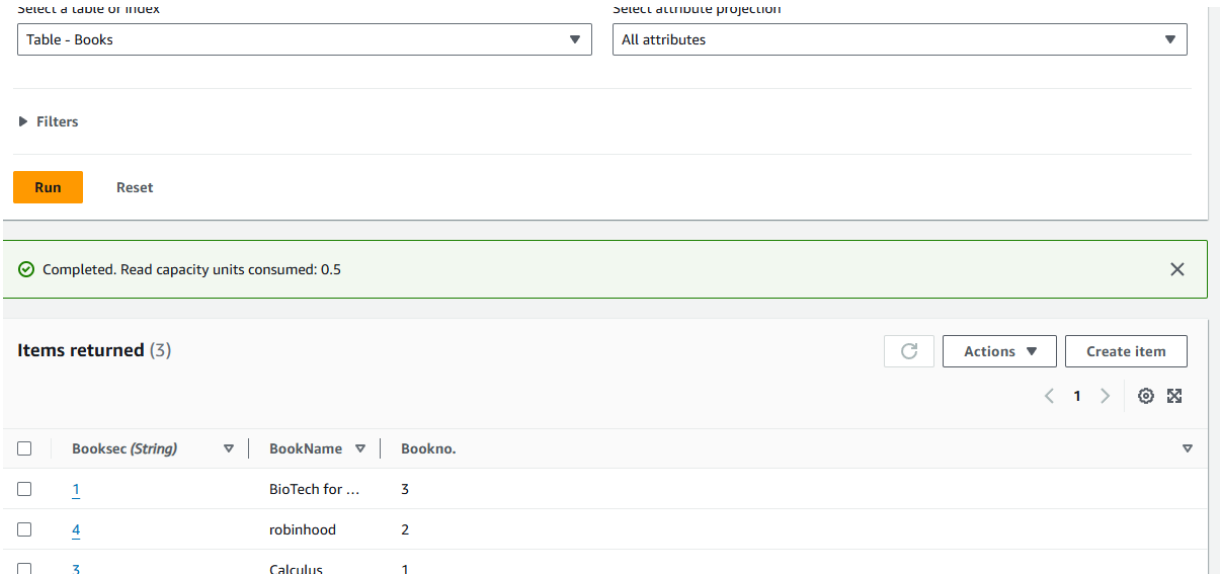


4. Elastic Load Balancer: Distributes incoming application traffic across multiple EC2 instances to ensure high availability and reliability.

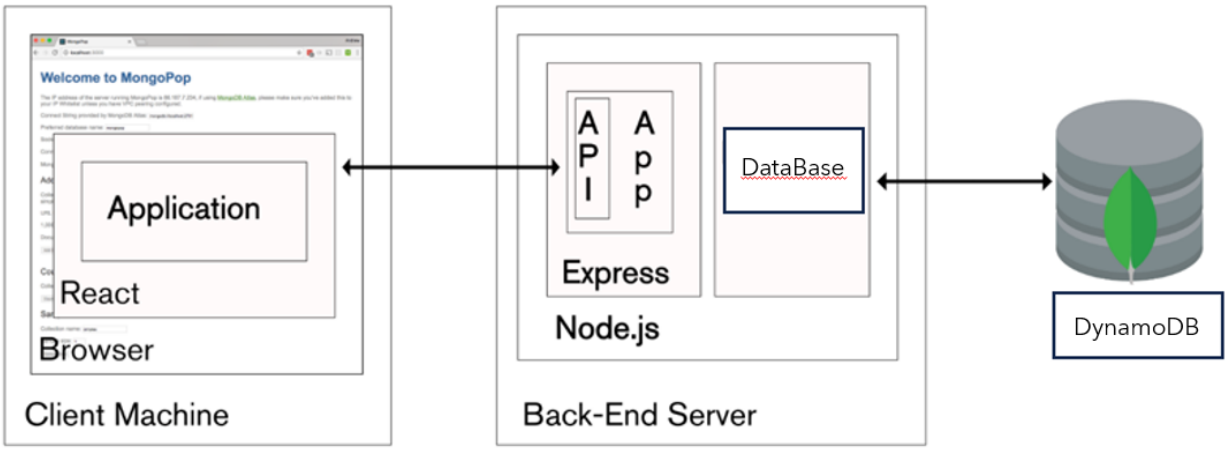
5. CloudWatch Metrics: Provides real-time monitoring of application performance and resource utilization. Custom metrics and alarms will be set up to detect and respond to performance issues.



6. DynamoDB: Stores unstructured data, providing fast and flexible access to metadata and other dynamic content.



7. IAM: Manages secure access to AWS resources, implementing role-based access control (RBAC) to ensure that only authorized users can perform specific actions.



4. Major Features/Functions

Application Pages:

Register: User registration page for creating new user accounts.

Login: User login page for authentication and access to personalized features.

Home: Dashboard displaying the book catalog with search functionality.

Book Details: Detailed view of a selected book, including its description, author, genre, and availability status.

Borrow: Page for borrowing a book, updating the availability status and recording the borrow transaction.

Return: Page for returning a borrowed book, updating the availability status and recording the return transaction.

User Profile: Page displaying user information and borrowing history.

Admin Dashboard: Administrative interface for managing book inventory, user accounts, and generating reports.

Key Functionalities:

Search and Filter: Users can search for books by title, author, genre, and availability status. Advanced filtering options will help narrow down search results.

User Authentication: Secure user authentication using JWT (JSON Web Tokens) for session management.

Book Management: Admins can add, update, and delete book records, including bulk upload options.

Borrowing System: Tracks the borrowing and returning of books, updating the status and due dates.

Notification System: Sends email or SMS notifications for due dates, overdue books, and administrative alerts.

Reporting: Generates reports on book inventory, user activity, borrowing trends, and overdue books.

5. Database Description and Business Rules:

Table 1: Admins

id: Unique identifier for each admin.

name: Name of the admin.

email: Email address of the admin.

password: Encrypted password for authentication.

Created on: Date and time when the admin account was created.

updated_on: Date and time when the admin account was last updated.

Table 2: Users

id: Unique identifier for each user.

name: Name of the user.

email: Email address of the user.

phone: Phone number of the user.

password: Encrypted password for authentication.

created_on: Date and time when the user account was created.

updated_on: Date and time when the user account was last updated.

Table 3: Books

id: Unique identifier for each book.

title: Title of the book.

author: Author of the book.

genre: Genre of the book.

available: Availability status (true if available, false if borrowed).

Created on: Date and time when the book record was created.

Updated on: Date and time when the book record was last updated.

Table 4: Borrow Records

id: Unique identifier for each borrow record.

book_id: ID of the borrowed book.

user_id: ID of the user who borrowed the book.

borrow_date: Date and time when the book was borrowed.

due_date: Date and time when the book is due.

return_date: Date and time when the book was returned.

status: Status of the borrow record (borrowed, returned).

Business Rules:

Unique Constraints: Ensure that email addresses for users and admins are unique.

Foreign Key Relationships: BorrowRecords should reference valid user_id and book_id values.

Data Integrity: Ensure the availability status of a book is accurately reflected in the Books collection and updated in real-time during borrowing and returning transactions.

Role-Based Access Control: Only admins can perform certain actions such as adding or removing books, generating reports, and managing user accounts.

6. Security:

Authentication: User authentication will be managed through secure login mechanisms with password hashing using bcrypt and JWT for session management.

Authorization: Role-based access control (RBAC) will be implemented to ensure that only authorized users can perform certain actions (e.g., only admins can add or remove books).

Data Security: Data in transit will be encrypted using SSL/TLS, and data at rest will be encrypted using AWS Key Management Service (KMS).

Access Control: AWS IAM will be configured to manage permissions and access controls for AWS resources, ensuring that only authorized services and users can access critical components.

7. Scalability and Performance:

Auto Scaling: AWS Auto Scaling will be configured to handle varying traffic loads by automatically adjusting the number of EC2 instances based on demand.

Load Balancing: Elastic Load Balancing will distribute incoming application traffic across multiple EC2 instances to ensure high availability and fault tolerance.

Database Optimization: AWS RDS with MySQL will be optimized for performance with appropriate indexing, query optimization techniques, and read replicas to handle high read loads.

Caching: Implement caching strategies using AWS ElastiCache to reduce database load and improve response times for frequently accessed data.

8. Monitoring and Logging:

AWS CloudWatch: For real-time monitoring of application performance and resource utilization. Custom metrics and alarms will be set up to detect and respond to performance issues.

Centralized Logging: Application logs will be centralized and managed using AWS CloudWatch Logs for easy troubleshooting and audit trails.

Alerting: Configured alerts will notify administrators of critical issues such as server downtimes, high latency, and security breaches.

9. Development Tools and Practices:

Version Control: Git will be used for version control to manage code changes and collaboration among developers.

CI/CD Pipeline: Continuous Integration and Continuous Deployment (CI/CD) pipelines will be set up using AWS CodePipeline and AWS CodeBuild to automate testing and deployment.

Code Quality: Linting tools and code review processes will be implemented to ensure code quality and maintainability.

Automated Testing: Unit tests, integration tests, and end-to-end tests will be implemented using frameworks like Jest and Cypress to ensure code quality and application stability.

10. Conclusion:

BookHub aims to provide an efficient, scalable, and secure solution for library management. By leveraging AWS cloud infrastructure and modern web development technologies, the application will offer a seamless experience for both librarians and users, enabling easy management and access to library resources. This comprehensive system