Cloud-native E-commerce Application Deployment

David Antonio Cepeda

010409764

D342 Task 2

Western Governors University

Table of Contents

A. Description of Problem

Our client, CartWise E-commerce, is a rapidly growing online business located in New York City. They have been successfully selling products through their web application, which has unfortunately been built as a monolithic application. As the business continues to grow and scale, the client has recognized the limitations of their current application architecture. They are seeking a more flexible, scalable, and maintainable solution.

The current monolithic application lacks the agility needed to adapt to fluctuating business demands and the increasing web traffic on their website. The tightly coupled nature of the application makes it challenging to change, scale, and maintain individual components without affecting the entire codebase. This inflexibility greatly impacts the business's ability to innovate, respond to market changes, and ensure a pleasant user experience.

The main drivers for this project include the need to improve scalability, maintainability, and flexibility. By decoupling the application and deploying it using cloud-based technologies, the client hopes to achieve better resource allocation, increase system reliability, and shorten the time it takes to release new features and updates.

B. Summary of Proposed Solution

The existing monolithic application uses React, Redux, and Firebase, with Stripe for payment processing and Cloudinary for image hosting. The application has product listings, product details pages, shopping cart functionality, and checkout functionality. The

cloud-computing solution involves re-architecting the existing monolithic application to a cloud-native, microservices-based architecture and deploying it on Amazon Web Services (AWS). The solution also uses Docker and Kubernetes for easier deployment, scaling, and management of the application components.

The project's success will be based on the following metrics: reduced deployment time, increased application availability, faster response times to user requests, and improved resource utilization. We will track these metrics by monitoring system logs, analyzing resource usage, and measuring user satisfaction.

B1 Security Needs

The proposed cloud-computing solution addresses the security needs of the organization by taking advantage of AWS's built-in security features, such as Identity and Access Management (IAM), AWS Shield, and Amazon Inspector. These services help ensure proper access control, protect against Distributed Denial of Service (DDoS) attacks, and monitor security vulnerabilities. The use of containerization through Docker and Kubernetes also will provide an isolated environment for each microservice, further reducing the attack surface and improving security.

C. Stakeholder Needs

The cloud-computing solution proposed for CartWise E-commerce involves going from a monolithic application to a cloud-native, microservices-based architecture. This change will impact several stakeholder groups. This includes the development team, IT operations, marketing team, customers, and management.

Development Team: The development team needs an architecture that enables quick and efficient updates while minimizing the impact on the entire system. The proposed microservices-based architecture allows them to develop, test, and deploy individual components independently, streamlining the development process and reducing overall complexity. This leads to improved efficiency, cost savings in development, and faster implementation of new features.

IT Operations: The IT operations team needs a scalable, highly available, and easily manageable infrastructure to ensure smooth application performance. The proposed solution's containerization approach along with Kubernetes provides automated scaling, rolling updates, and self-healing capabilities that simplify the management of the application. It can also improve resource utilization and help reduce infrastructure costs.

Marketing Team: The marketing team strives to launch new features, promotions, and updates rapidly to respond to changes in market and customer needs. The proposed solution's decoupled architecture allows the team to quickly and effectively adapt to changes in the market.

Customers: The customers expect an easy, reliable, and secure shopping experience. The proposed solution's enhanced scalability, availability, and security features ensure that the application can handle increasing traffic and maintain high-performance standards. This ultimately results in general customer satisfaction.

Management: Management wants to make sure that the project is valuable to the business, while also staying within budget and meeting deadlines. The proposed solution can help with this by reducing the time it takes to bring products to market, making better use of resources, and optimizing infrastructure costs; these are all benefits that align with management's goals.

D. Comparison of Proposed Solution

Alternative solutions worth considering are a Platform-as-a-Service (PaaS) approach and the implementation of an on-premises data center.

Platform-as-a-Service (PaaS): A PaaS solution, such as Google App Engine or Microsoft Azure App Service, would provide a managed platform for CartWise E-commerce to build, deploy, and scale their applications. While PaaS offers simplified application deployment and management, it does not have the flexibility or granular control available with the proposed AWS solution. Additionally, a PaaS platform could impose specific restrictions on application architecture, libraries, and configurations. This limits customization and optimization options for the client's application.

On-premises Data Center: To implement an on-premises data center you would have to build and manage private infrastructure only to host a single e-commerce application. This involves setting up and configuring physical servers, installing the required software, and manually managing the scaling of resources. While the benefits of an on-premises data center include full total control over the hardware and software stack, it comes with incredibly high upfront costs, increased maintenance costs, and limited scalability compared to cloud-based solutions.

E. Identification of Threats and Vulnerabilities

Inherently, the proposed cloud-computing solution comes with security concerns since sensitive customer information and transaction data will be stored and processed in the cloud instead of on-premises infrastructure. To mitigate these risks, it is extremely important to recognize and evaluate these threats and vulnerabilities as well as to implement appropriate countermeasures.

Network Vulnerabilities: Some network vulnerabilities worth considering are man-in-the-middle attacks, DDoS attacks, or unauthorized access due to misconfigured security groups. AWS does however provide built-in security features to mitigate these risks, but the likelihood and the impact of network vulnerabilities should still be assessed carefully. This involves constantly monitoring network traffic, implementing access control policies, and ensuring that security groups and virtual private clouds are correctly set up.

Human Vulnerabilities: Human error or malicious insider threats can also significantly impact the security of the application. Even the accidental deletion of data or the misconfiguration of security settings can lead to data breaches. To prevent this the company must enforce strict access control policies, provide regular security training for employees, and implement monitoring tools to detect suspicious activities. In the case of a security breach, a comprehensive incident response plan can ensure that any potential problems are handled accordingly.

F. References

https://github.com/ed-roh/react-ecommerce