**המחלקה להנדסת תוכנה**

**פרויקט גמר – ה'תשפ"ד**

**גיבוי לבית מלון**

**reservation backups for hotels**

**מאת**

**Saja Abu Aisheh**

**ת.ז. 308444223**

**מנחה אקדמית: ד"ר Shimrit Tzur**

מערכות ניהול הפרויקט:

|  |  |  |  |
| --- | --- | --- | --- |
| # |  | מערכת | מיקום |
| 1 |  | מאגר קוד | https://github.com/sajabilal/reservation-backup-for-hotels |
| 2 |  | יומן | https://docs.google.com/spreadsheets/d/1DIBM57YSvE4yOHi6ZkDjtKxJR9NmcMMiaQtz846IhaQ/edit?gid=0#gid=0 |

מידע נוסף (מחקו את המיותר)

|  |  |
| --- | --- |
| סוג הפרויקט | יוזמה שלי |
| פרויקט ממשיך | זה פרויקט חדש |

**Introduction**

In today's digital landscape, ensuring uninterrupted access to critical business systems is paramount, especially for industries like hotel reservation companies that handle high volumes of time-sensitive customer data. This project focuses on creating a robust solution to mitigate the impact of system downtime by providing customers and authorized employees with seamless access to reservation data, even during outages.

The proposed solution is designed to automatically detect system downtime using AWS Route 53 health checks and implement a failover routing policy. Upon identifying a system outage, traffic will be redirected to an alternative website hosted on AWS S3 and served via CloudFront. This alternative system allows users to query reservation data from a secondary database, ensuring business continuity and customer satisfaction.

A key component of this solution is the regular backup of reservation data from the primary database to the secondary database. The frequency of these backups will be customizable, based on the company's requirements, to ensure that data remains up-to-date. By leveraging AWS services, this approach not only provides a failover mechanism but also maintains a secure, scalable, and highly available environment.

**Problem Description**

## **Problem Requirements and characterizations**

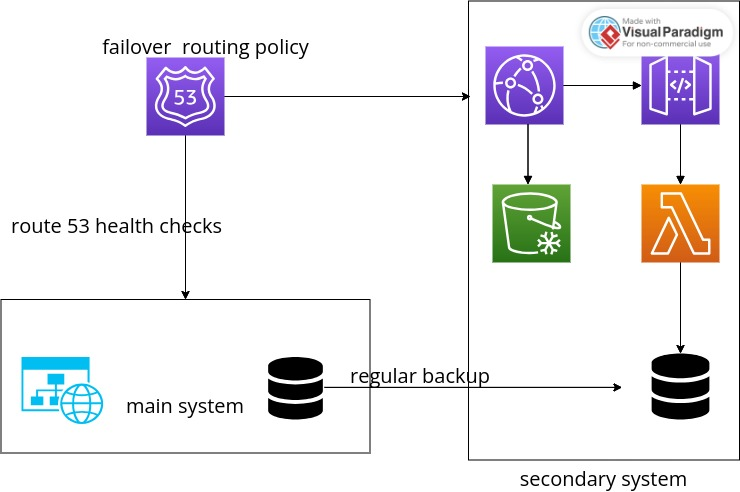
The proposed solution provides users with a reliable platform to access reservation data during system downtime. From the user's perspective, the product ensures seamless redirection to an alternative interface, where both customers and authorized employees can securely query and retrieve reservation information. Users will experience minimal disruption as the system automatically detects downtime and routes traffic to a backup website. The product prioritizes usability by presenting a straightforward error notification page, followed by a clear pathway to access reservation data. Additionally, the system allows companies to define access permissions, ensuring that only authorized users can query the backup database. This focus on availability, security, and ease of access ensures that users have a dependable and intuitive experience when engaging with the solution during critical events.

**The problem in terms of software engineering**

#### Expected Challenges

1. Seamless Failover Implementation:  
   Configuring AWS Route 53 health checks and the failover routing policy to ensure instantaneous redirection to the backup system during downtime is a technical challenge. Any delays or misconfigurations could lead to service unavailability or inconsistent user experiences.
2. Data Synchronization:  
   Keeping the secondary database consistently updated with reservation data from the primary system requires precise scheduling and robust backup mechanisms. Balancing update frequency with cost and performance is a key challenge.
3. Access Control Without Customer Credentials:  
   Designing a secure access mechanism for customers who lack login credentials demands innovative approaches, such as leveraging unique identifiers and multi-field validation, while ensuring data privacy and security.
4. System Security and Data Integrity:  
   Preventing unauthorized access to the secondary database and maintaining data integrity are critical. Implementing encryption, secure transmission protocols, and access control policies must be carefully planned to mitigate risks.
5. Scalability and Cost Optimization:  
   The solution must handle varying traffic loads during downtime without incurring excessive costs, particularly for storage and database queries in AWS. Balancing scalability with budgetary constraints is a unique software engineering challenge.
6. User-Friendly Design:  
   Ensuring that the failover system provides an intuitive and seamless experience for customers and employees requires attention to user interface design and clear workflows.

**Solution Description**

The system implementing the proposed solution consists of two main components: the primary system and the secondary system, working together to ensure continuous access to reservation data during system downtime.

1. Primary System:  
   The primary system operates as the main service, handling user requests and serving data during normal operation (system uptime). It is the primary point of interaction for users under regular conditions.
2. Secondary System:  
   The secondary system functions as the backup solution, designed to take over during downtime. It ensures that users can still query reservation data when the primary system is unavailable.

AWS Route 53 is a critical component of this solution. It continuously performs health checks on the primary system to monitor its availability. In the event of a failure or an unhealthy status, Route 53 automatically redirects traffic to the secondary system using its failover routing policy.

The secondary system leverages several AWS services:

* Amazon S3: Hosts the static content, including the HTML and CSS files for the alternative user interface.
* CloudFront: Acts as a content delivery network (CDN) to efficiently serve the static website with low latency.
* API Gateway and Lambda: Serve as the backend of the secondary system, processing user requests and interacting with the database.
* Secondary Database: A read-only database that stores synchronized reservation data, enabling user queries during downtime.

Together, these components create a seamless and automated failover system that ensures uninterrupted access to critical reservation data, enhancing user experience and system reliability.

***Similar Existing Solutions And*** **C*omparison***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Predictive Maintenance with AI** | Real-Time Recovery Solutions | Backup Sites | **High Availability Software** | MY Solution | Feature/Aspect |
| Predicts and prevents failures in systems or hardware by utilizing AI and analytics. | Rapidly restores IT infrastructure to reduce operational disruption. | Provides alternative physical or cloud locations to continue operations in case of system failure. | Maintains system availability by minimizing downtime with redundancy and failover. | Ensures continuous access to hotel reservation data during system downtime. | **Purpose** |
| Integrates IoT devices and sensors to collect data, using ML to predict and act on potential issues. | Combines failover, cloud restoration, and automated recovery mechanisms. | Physical/cloud duplicate sites categorized as hot, warm, or cold depending on readiness. | Uses load balancers, redundant servers, and clustering to distribute traffic and ensure availability. | Relies on AWS services like Route 53, S3, CloudFront, API Gateway, and a secondary database. | **Architecture** |
| High initial setup cost, but long-term savings by preventing failures. | Moderate to high, depending on the level of automation and infrastructure setup. | High for hot sites; moderate for warm and cold sites, based on readiness and maintenance needs. | High costs due to redundancy in hardware, software, and operational requirements. | Cost-effective for hotel reservation companies due to selective resource usage and customization options. | **Cost** |

\*\*it is a solution to add to other system down solutions to make the system resilient and not a stand alone solution.