

White Paper: Database Hosting Solutions

Brycen Williams

Date: 10/11/2024

1. Executive Summary

This white paper evaluates various database hosting solutions for our application. We will compare three major database hosting options: (Compare hosting solutions here) This document will also discuss development considerations, including programming languages and ORM frameworks that best suit each database solution based on criteria such as (Criteria here) The analysis will lead to recommendations for the optimal database hosting solution.

- **Purpose:** To identify the best database hosting solution for application deployment.
- **Problem:** Choosing the right hosting solution is critical for ensuring data reliability and application performance.
- **Recommendation:** A detailed analysis will support recommendations for the best database hosting option.

2. Problem Statement

Selecting an appropriate database hosting solution can significantly impact data management, application performance, and overall user experience. Different hosting options have unique features, pricing models, and support mechanisms, making it essential to evaluate them thoroughly to ensure alignment with application requirements and business goals.

3. Criteria for Evaluation

4. Overview of Solutions

Solution 1: AWS RDS

- **Description:** Amazon web services relational database service.

Solution 2: Google Cloud SQL

- **Description:** A Google cloud database service.

Solution 3: Microsoft Azure SQL Database

- **Description:** Supports MySQL databases.

5. Comparative Analysis

Criteria	AWS RDS	Google Cloud SQL	Azure SQL
Cost per GB	.115	.17	.115
Cost per hour of hosting	.018 - .1	.0133 - .096	.0047
Ease of use	Pros: Easy to setup. Cons: Steeper learning curve due to high number of features	Pros: Very beginner friendly in general Cons: Not made for larger / more complex databases	Pros: Easy to setup basic databases. Cons: N/A. Simple to use
Features (diagnostics, usage dashboard, etc)	VERY feature heavy. If you can think of a feature you would want, it exists within the AWS ecosystem.	Lighter on features. Allows for easy-to-use pre-configured settings.	Similar to AWS. One additional bonus is that it will work well with other Microsoft software such as Visual Studio.

## 6. Recommendation – AWS RDS.

### Why AWS?

- We are learning to use AWS as part of this course.
- While AWS is slightly more expensive on average than Google Cloud SQL, it costs less per GB of storage. It also generally has more features.
- AWS also has a free tier. This includes 750 service hours a month, for 12 months. We would also be able to use 20GB of standard storage.
  - o This would allow us to test and develop the database, without worrying about database hosting costs during development.
- Depending on the backend hosting service we decide to go with, we could also use this same free tier for the web server.

## 7. Database Type

This white paper has purposely left out the database type that we will be using. The two main options for AWS RDS are MySQL, and Postgres. Both would work fine, but I would recommend going with Postgres if we decide to use a backend development framework such as ASP.NET, due to its ability to store data as objects. This feature would be useful for integrating with OOP languages such as C#.

---

## 8. Implementation Plan

1. **Cost and Budget:** .018 per hour of hosting over 750 / mo. .115 per GB over 20 / mo.
  2. **Timeline:**
  3. **Resources Required:**
    - AWS Canvas course for training.
    - SQL language.
    - A code editor / text editor for writing SQL statements.
    - A local SQL interpreter.
    - An Oracle account.
- 

## 9. Conclusion

A main goal of ours is to keep our costs low during development. By using AWS, we will be able to design and test a relational (or object relational) database without needing to worry about initial costs. We will need to decide between two database types, relational and object relational databases. We should use PostgreSQL alongside an object relational database if we are going to be using object-oriented programming practices in our backend.

---

## References

- [AWS Free Tier Information](#)
- [Azure SQL](#)
- [AWS RDS](#)
- [About PostgreSQL](#)
- [MySQL Documentation'](#)
- [About Cloud SQL](#)