International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Cost - Effective Big Data Storage and Processing: Analysing Tradeoffs between S3, RDS, and DYNAMODB in AWS

Girish Ganachari

Email: *girish.gie[at]gmail.com*

Abstract: ***Everything today is reliant on data, so firms need fast, cheap solutions to store and manage massive volumes of it. Amazon S3, RDS, and DynamoDB are three popular AWS services examined in this study. The test evaluates their performance, scalability, and compatibility with big data apps. The advantages and downsides of each service for companies has been emphasised. This helps them make wise data management decisions.***

Keywords: Big Data, AWS, S3, RDS, DynamoDB, Cost - Effective Storage, Cloud Computing

1. Introduction

Big data has transformed company data management as businesses need affordable storage and processing options to keep up with their rising data volume, speed, and variety. AWS cloud services like Amazon S3, Amazon RDS, and Amazon DynamoDB handle a lot of data. Each service has pros and cons in pricing, speed, and growth. To help organisations choose the best service, this study examines these services in detail.

2. Overview of AWS Storage Solutions

a) Amazon S3

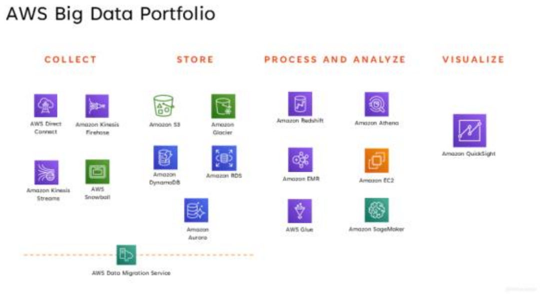

Figure 1.1: AWS Big Data Portfolio

A popular object storage solution, Amazon Simple Storage solution (S3), can scale, protect data, and be fast. A firm can store and retrieve any amount of online data. Data saved, queries made, and data transmitted outside AWS regions determine S3 prices. This makes huge data storage and transmission cheap. S3 supports Standard, Intelligent - Tiering, and Glacier storage. Each class should have varied supply and cost - effectiveness [1].

b) Amazon RDS

Amazon Relational Database Service (RDS) simplifies cloud relational database setup, management, and growth. It handles time - consuming administrative activities including setting up hardware, databases, patches, and backups while offering flexible and economical storage. Firms can utilise Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server with RDS. Companies can choose the database engine that best suits their apps and tech [2].

c) Amazon DynamoDB

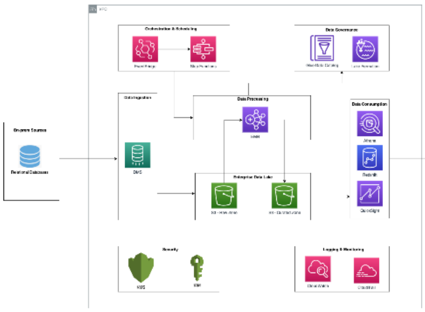

Figure 1.2: AWS Database Blog

Amazon DynamoDB, a fully managed NoSQL database service, provides quick, stable, and scalable performance. Store and retrieve any quantity of material and manage any number of queries. The pay - per - request pricing approach in DynamoDB ensures customers only pay for what they consume. It's ideal for apps that need to be online 24/7 and manage a lot of traffic. DynamoDB Streams and global tables allow real - time data processing and replication between regions [3].

2.1 Cost Analysis

a) Storage Costs

Prices are crucial when choosing cloud storage as Amazon S3 rates by monthly data storage. A firm must pay to send and receive data. Amazon RDS pricing depends on database instance type, storage space, and IOPS. How much data it stores, how fast it can read and write, and any extra tools like backups and global tables affect its price.

Amazon S3: Amazon S3's typical storage costs $0.023, whereas Glacier Deep Archive costs $0.004. S3 data lifecycle management is also customisable. Data can be automatically moved between storage types to save money [4].

DOI: https://dx.doi.org/10.21275/SR24827095532

Amazon RDS: Amazon RDS prices vary greatly by database engine and server and large instances with 100 GB of SSD storage costs $0.25 per hour and $0.115 per GB - month. Extra fees apply for backup storage and file movement. Reading copies and using numerous AZs increases expense [5].

Amazon DynamoDB: Read and write data costs extra with Amazon DynamoDB, and storage costs $0.25 per GB each month. $1, 000 write request units cost $1.25 per month, whereas 1, 000 read request units cost $0.25. DynamoDB users can choose between on - demand and planned capacity options to reduce costs [6].

b) Performance and Scalability Costs

Big data applications that handle real - time data and access it quickly need speed and growth.

Amazon S3

Given that it grows, S3 can handle lots of data and queries quickly. Item size and request count affect performance. Performance enhancements like S3 Transfer Acceleration cost more. S3 runs websites and stores plenty of data [7]. Because it grows and runs swiftly.

Amazon RDS

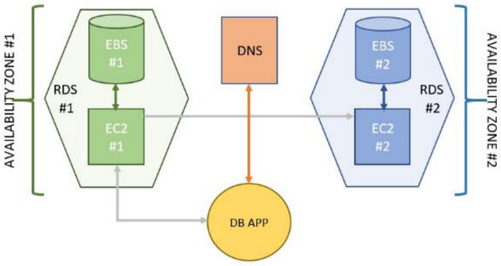

Figure 1.3: Amazon RDS

Different instance kinds balance cost and performance. Moving to larger instances costs more as the firm expand. Read copies and multi - AZ deployments improve performance and uptime but cost more. RDS has numerous uses because it can be moved up, down, and left and right [8].

Amazon DynamoDB

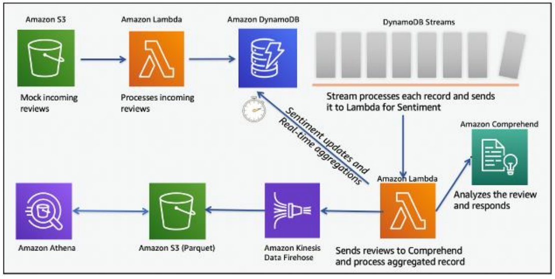
Amazon S3
Amazon Lambda
Amazon DynamoDB
DynamoDB Streams
Mock incoming reviews
Processes incoming reviews
Stream processes each record and sends it to Lambda for Sentiment
Sentiment updates and Real-time aggregations
Amazon Comprehend
Amazon Lambda
Analyzes the review and responds
Amazon Athena
Amazon S3 (Parquet)
Amazon Kinesis Data Firehose
Sends reviews to Comprehend and process aggregated record

Figure 1.4: Integrate Amazon DynamoDB with machine learning

Amazon DynamoDB claims it can adjust data read and write speeds based on application needs. On - demand capacity costs more than provided capacity, but it performs better for unplanned jobs. Apps with many calls benefit from DynamoDB's ability to handle millions of calls per second without delay [9].

c) Total Cost of Ownership (TCO)

Business direct and indirect costs are included. Direct costs include storage and speed.

Amazon S3: S3's low TCO comes from its ease of usage and maintenance. Moving data to cheaper storage types on their own with intelligent tiering and lifecycle controls can reduce costs even more. S3 is affordable for long - term file storage because it requires little management [10].

Amazon RDS: Many administrative duties can be simplified with AWS RDS. However, managing backups, security, and scaling can increase TCO. The lower requirement for DBA tools covers these costs. RDS handles difficult questions and deals, making it useful for many groups [11].

Amazon DynamoDB: Fully managed and self - growing, DynamoDB requires less management. This lowers app TCO for changing workloads. Pay - per - use prices let users pay for resources used [12]. This reduces overprovisioning risk and expenses.

3. Use Cases and Suitability

a) Amazon S3 Use Cases

Data Lakes: “Data Lakes" store structured and unstructured data. This enables advanced analytics and machine learning. S3 is ideal for data lakes since it can scale up or down and handle numerous types of data [13].

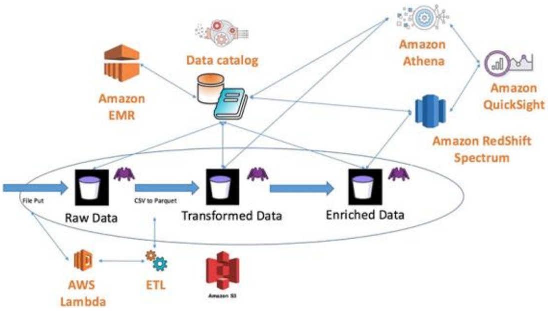

Figure 1.5: Amazon S3 Big Data Lake

Backup and restore: This feature saves backups for a long time and is cheap. For even greater savings, choose automated lifetime management. Important files are always safe in S3 because it's long - lasting and available [14].

Content Distribution: Distributing non - changing photos, movies, and large files worldwide, S3 and Amazon CloudFront coordinate well to transfer data quickly [15].

b) Amazon RDS Use Cases

Transactional Databases: ACID - compliant programs employ these for complex searches. Because it supports many database engines, RDS can manage many transactional jobs [16].

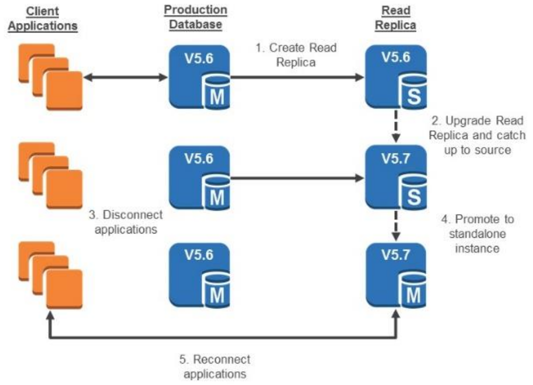
Client Applications
Production Database
Read Replica
V5.6
M
1. Create Read Replica
V5.6
S
2. Upgrade Read Replica and catch up to source
V5.6
M
V5.7
S
3. Disconnect applications
4. Promote to standalone instance
V5.6
M
V5.7
M
5. Reconnect applications

Figure 1.6: Amazon Relational Database

Enterprise Applications: ERP, CRM, and other corporate programs that must be always on and working. Automatic management makes business application database maintenance easier using RDS [17].

Data Warehousing: Amazon Aurora and PostgreSQL aid with complicated searches and data processing for analytical activities. RDS is fast and easy to add users to, making it suitable for warehouse apps [16].

c) Amazon DynamoDB Use Cases

Real - Time Applications: Real - time apps for games, IoT, and phones demand low latency and high performance. DynamoDB scales quickly and is ideal for real - time apps.

NoSQL Workloads: NoSQL workloads store unstructured or partially structured data with flexible model rules. DynamoDB's schema - free design makes modifications and feature additions trivial [20].

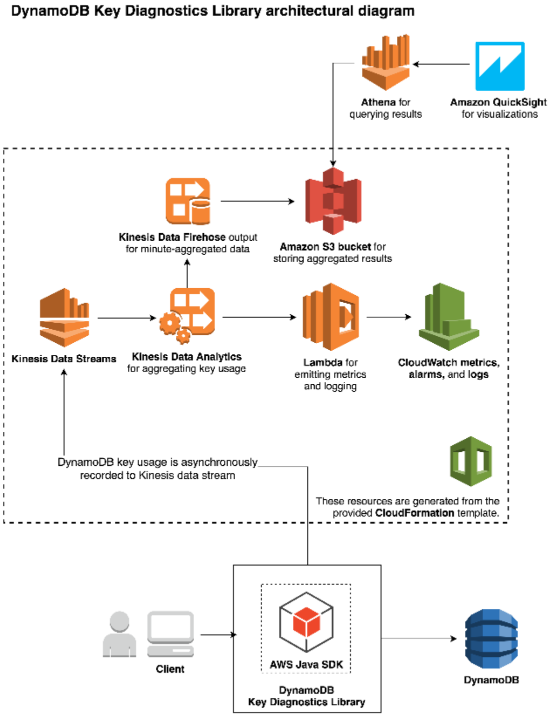
DynamoDB Key Diagnostics Library architectural diagram
Athena for querying results
Amazon QuickSight for visualizations
Kinesis Data Firehose output for minute-aggregated data
Amazon S3 bucket for storing aggregated results
Kinesis Data Streams
Kinesis Data Analytics for aggregating key usage
Lambda for emitting metrics and logging
CloudWatch metrics, alarms, and logs
DynamoDB key usage is asynchronously recorded to Kinesis data stream
These resources are generated from the provided CloudFormation template.
Client
AWS Java SDK
DynamoDB Key Diagnostics Library
DynamoDB

Figure 1.7: Key Diagnostic Library in DynamoDB

Event - Driven Architectures: In this design, DynamoDB Streams process data in real time so apps may react fast to data changes. It integrates with other AWS services, making it ideal for event - driven systems [21].

4. Performance Comparison

a) Amazon S3

Amazon S3 can handle several queries per second. What works relies on its size and request. Uploading in sections and using S3 Transfer Acceleration speeds up uploads. As it grows, S3 can manage plenty of data. This makes it useful for running websites and evaluating huge data [22].

b) Amazon RDS

There are several good Amazon RDS instances and storage alternatives. RDS supports read replicas and Multi - AZ deployments for performance and uptime. RDS can scale up and down and left and right, making it suitable for transactional databases and data warehouse apps [23].

c) Amazon DynamoDB

Amazon DynamoDB is for speedy apps. Due to its millisecond response, several people can utilise it simultaneously. The on - demand capability option is ideal for apps that don't know their workload. Because it can scale, DynamoDB is ideal for speedy, low - latency real - time apps [24].

5. Scalability Comparison

a) Amazon S3

Amazon S3 can accommodate apps with changing storage needs because it scales quickly. S3 smart - tiering and lifespan rules enable it to transfer data to cheaper storage classes based on usage, saving money. S3 is ideal for large data storage and backup [25]. It handles extra data effortlessly.

b) Amazon RDS

Read copies and larger servers can be added as Amazon RDS grows. It can expand horizontally. However, scaling may need downtime that hinders app performance. RDS can handle everything from small apps to massive commercial systems because it can grow up and down and left and right [16].

c) Amazon DynamoDB

Amazon DynamoDB does different jobs without any intervention. It can do them alone. Data can be copied across regions using global tables. This improves data accessibility and disaster recovery. DynamoDB can grow quickly and manage a lot of traffic, making it ideal for apps that don't know how much work they'll get [21].

6. Security Comparison

a) Amazon S3

Amazon S3 provides many security measures, including data encryption in transit and at rest. Access control lists and bucket rules are included. S3 and AWS IAM grant users the right access. S3 ensures data safety and compliance by limiting access and security [12].

b) Amazon RDS

Amazon RDS protects data at rest with AWS KMS. Data transmission uses SSL/TLS. The IAM and RDS control who can view what. RDS provides VPC sharing and security groups to protect the network. RDS can handle private data and meet the regulations because it handles security and compliance [4].

c) Amazon DynamoDB

Data is encrypted when stored in AWS DynamoDB and sent via SSL/TLS. IAM lets the firm control access and improve security and reporting with VPC endpoints and AWS CloudTrail integration. Due to its security and compliance features, DynamoDB can manage sensitive data and follow laws [12].

7. Conclusion

Each AWS service has pros and cons. Amazon S3, a terrific way to transfer and store a lot of data, has numerous cheap and quick ways to add space. Due to its automated control tools and compatibility with many database systems, Amazon RDS can handle transactional and analytical tasks. Amazon DynamoDB is great for real - time, high - traffic apps since it works well and can be expanded. Understanding these differences helps businesses choose the best service. This will enable fast and affordable cloud storage and management.

References

[1] Shrestha, S., 2019. Comparing Programming Languages used in AWS Lambda for Serverless Architecture.

[2] Hassan, M., 2021. Public Cloud - Based Private Python Package Serving Platform.

[3] Heinonen, J., 2020. From Classical DW to Cloud Data Warehouse. *Obtenido de https: //helda. helsinki. fi/bitstream/handle/10138/322467/JyrkiHeinonen_Masters_Thes is_V1.0. pdf.*

[4] Müller, I., Marroquín, R. and Alonso, G., 2020, June. Lambada: Interactive data analytics on cold data using serverless cloud infrastructure. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp.115 - 130).

[5] Armenatzoglou, N., Basu, S., Bhanoori, N., Cai, M., Chainani, N., Chinta, K., Govindaraju, V., Green, T. J., Gupta, M., Hillig, S. and Hotinger, E., 2022, June. Amazon Redshift re - invented. In *Proceedings of the 2022 International Conference on Management of Data* (pp.2205 - 2217).

[6] Angle, I. C., 2022. Towards Cloud - Based cost - effective serverless information system.

[7] Petrovska, J. and Ajdari, J., 2019. Amazon's Role in the Field of Cloud Relational And noSQL Databases: A Comparison Between Amazon Aurora and DynamoDB. ISCBE 2019, p.214.

[8] Belcastro, L., Marozzo, F., Talia, D. and Trunfio, P., 2017. Big data analysis on clouds. Handbook of big data technologies, pp.101 - 142.

[9] Müller, I., Marroquín, R. and Alonso, G., 2020, June. Lambada: Interactive data analytics on cold data using serverless cloud infrastructure. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp.115 - 130).

[10] Kumari, A. and Sahoo, B., 2022. Serverless architecture for healthcare management systems. In *Handbook of research on mathematical modeling for smart healthcare systems* (pp.203 - 227). IGI Global.

[11] Fofanah, A. J., 2021. Review of Knowledge Management in Optical Networks, Lambda Architecture using Database Technologies in Cloud Settings. *International Journal of Scientific and Research Publications, 11* (8).

[12] Pathania, P. and Mithani, R. D., 2020, September. Sustainability in migrating workloads to public clouds. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (pp.166 - 169).

[13] Deb, M. and Choudhury, A., 2021. Hybrid cloud: A new paradigm in cloud computing. *Machine learning techniques and analytics for cloud security*, pp.1 - 23.

[14] Borhani, A. H., 2018. *Adaptive resource optimization of three - tier web applications running on the cloud* (Doctoral dissertation). .

[15] Berisha, B., Mëziu, E. and Shabani, I., 2022. Big data analytics in Cloud computing: an overview. *Journal of Cloud Computing, 11* (1), p.24.

[16] Sandhu, A. K., 2021. Big data with cloud computing: Discussions and challenges. *Big Data Mining and Analytics, 5* (1), pp.32 - 40.

[17] Ramachandra, M. N., Srinivasa Rao, M., Lai, W. C., Parameshachari, B. D., Ananda Babu, J. and Hemalatha, K. L., 2022. An efficient and secure big data storage in cloud environment by using triple data encryption standard. *Big Data and Cognitive Computing, 6* (4), p.101.

[18] Yilmaz, N., Demir, T., Kaplan, S. and Demirci, S., 2020. Demystifying Big Data Analytics in Cloud Computing. *Fusion of Multidisciplinary Research, An International Journal, 1* (01), pp.25 - 36.

[19] Tang, S., He, B., Yu, C., Li, Y. and Li, K., 2020. A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications. *IEEE Transactions on Knowledge and Data Engineering, 34* (1), pp.71 - 91.

[20] Tang, S., He, B., Yu, C., Li, Y. and Li, K., 2020. A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications. *IEEE Transactions on Knowledge and Data Engineering, 34* (1), pp.71 - 91.

[21] Moltó, G., Naranjo, D. M. and Segrelles, J. D., 2020. Insights from learning analytics for hands - on cloud computing labs in AWS. Applied Sciences, 10 (24), p.9148.

[22] Jumagaliyev, A. S., 2019. *A Modeling Language for Multi - tenant Data Architecture Evolution in Cloud Applications*. Lancaster University (United Kingdom).

[23] Singh Virdi, A., 2018. AWSLang: Probabilistic Threat Modelling of the Amazon Web Services environment.

[24] Rollino, S., 2022. Hands - on Comparison of Cloud Computing Services for Developing Applications.

[25] Stultiens, R., 2020. *Compliant but vulnerable: fixing gaps in existing AWS security frameworks* (Doctoral dissertation, MS thesis, Eindhoven University of Technology, Netherlands).