

Benefits of Running Databases on Amazon Relational Database Service Over Traditional On-Premises Servers

Prof. Nishigandha Bhalekar

Assistant Professor, Department of Computer Application, MIT Arts Commerce And Science College
Alandi Pune, India.

Abstract

Amazon RDS is a web service provided by Amazon to provide easy way to operate and scale relational database in traditional database to cloud.

In Traditional or on-premises server, CPU, memory, storage comes together as a bundle. RDS stipulate managed database service for handling most management tasks. After elimination of manual task which is most tedious part, RDS focus on application and users. This paper delves the migration of database to the cloud. The study highlights introduction to amazon RDS and its comparative study with traditional database. It also throws a light on comparative analysis like cost, performance, security, maintenance etc by taking use cases.

Keywords: RDS, monolithic block

Introduction:

The migration of databases to the cloud has been a significant trend in recent years, driven by various factors such as scalability, flexibility, cost-effectiveness, and the growing adoption of cloud-based applications. The migration of databases to the cloud represents a strategic move for enterprises seeking to leverage the benefits of cloud computing to optimize their data management processes and drive business success.

Naturally, one might wonder: what is the optimal relational database management system (RDBMS) for my cloud requirements? Is it sufficient to migrate the system I've relied on for years, or might there be superior alternatives? Research by IDC indicates that the "lift and shift" strategy—simply transferring all applications and databases from internal data centers to the cloud without implementing automated management capabilities—poses its own set of challenges and is perhaps best considered as an initial phase of cloud migration. Numerous businesses are embracing "cloud-native applications," aiming to optimize cloud resources and minimize ongoing usage to manage costs effectively. This strategy often involves restructuring applications into microservices rather than monolithic blocks of code. It's logical to seek out a fully managed cloud-native database service to align with this transition.

In fact, there's a significant advantage in leveraging a cloud-native database service when transitioning functionality to the cloud, even without application restructuring. Amazon Relational Database Service (Amazon RDS) is specifically tailored to facilitate the setup, operation, and scalability of relational databases in the cloud.

Recently, AWS collaborated with IDC to assess the firsthand experiences of its customers, aiming to precisely quantify the benefits they've gained. The insights derived from this study form the basis of the findings presented here. This paper represents the review of impact of using Amazon RDS service on the cost, operations, and performance and comparative analysis of various organization wide use cases.

Introduction to Amazon RDS:

Amazon RDS is a relational database service utilized for storing, organizing, and granting access to interconnected data, essential for tasks such as data analysis, modeling, reporting, and various other business applications.

The benefits of Amazon RDS encompass its demonstrated performance as a fully managed, cost-efficient relational database solution, coupled with top-tier security measures. With a track record spanning over a decade, customers have relied on Amazon RDS for its exceptional availability, durability, scalability, and security, crucial for supporting their most vital workloads in the cloud. Whether beginners or seasoned cloud users, customers find Amazon RDS remarkably user-friendly for initiating and overseeing database operations.

Amazon RDS allows database administrators (DBAs) to redirect their attention towards innovation and creating business value. By automating routine administrative tasks associated with database management—such as provisioning, patching, and backups—Amazon RDS streamlines operations, freeing up valuable time for DBAs.

Amazon RDS offers support for eight different database engines, comprising five open-source and three commercial options. Open-source engines include Amazon Aurora PostgreSQL-Compatible Edition, Amazon Aurora MySQL-Compatible Edition, RDS for PostgreSQL, RDS for MySQL, and RDS for MariaDB. On the other hand, commercial engines encompass RDS for SQL Server, RDS for Oracle, and RDS for Db2.

AWS administers RDS in a manner that empowers customers to attain the performance levels and service level agreements (SLAs) they demand, all while keeping operational costs in check. Moreover, since AWS manages deployment specifics, databases on RDS are straightforward to configure and establish, thereby saving users both time and effort. This eliminates the need for the deployment expertise typically required when setting up databases in an on-premises data center.

Comparative Analysis:

1. Cost Considerations

- With Amazon RDS, there's no initial commitment required. Users simply incur a monthly charge for each database instance they launch.
- Amazon RDS Reserved Instances offer the flexibility to reserve a DB instance for either a one- or three-year term, resulting in a substantial discount compared to the On-Demand Instance pricing for the same DB instance.
- It facilitates convenient and cost-effective utilization of databases for development and testing purposes, particularly when the database doesn't need to remain operational continuously.
- Amazon Aurora is specifically engineered to provide cost-effective pricing suitable for a wide range of applications characterized by low to moderate I/O usage.
- The AWS Free Tier allows you to initiate your journey with Amazon RDS at no cost for specific Single-AZ Instance databases, up to 750 hours per month. Additionally, it includes 20 GB of

General-Purpose SSD storage (gp2) and 20 GB of automated database backup storage per month, all provided for free for one year.

- For commercial engines, Amazon RDS provides various licensing models: Bring Your Own License (BYOL), Bring Your Own Media (BYOM), or License Included (LI). With BYOL and BYOM options, users can leverage their existing commercial software licenses for use with Amazon RDS.

Use Case: Build Web and Mobile Applications

Intuit Mint is a free personal financial management service used by more than six million consumers in the United States and Canada. The service consolidates a customer's financial information, including bank accounts, credit cards, and bills, and presents it in a unified platform. Mint additionally offers bill reminders and payment services, enabling users not only to view their finances but also to take necessary actions on them.

The Director of Application development and Operations of Intuit Mint, Sean McCluskey, says “Using Amazon RDS for MYSQL, we no longer need to spend time and money tuning IOPS to get strong database performance. By being in the cloud, we no need to worry about hardware acquisition costs. Ultimately, we have reduced our cost by 25%.”

2. Scalability

- You have the flexibility to adjust the compute and memory resources supporting your deployment, either up or down, with a maximum capacity of 128 vCPUs and 4,096 GiB of RAM. Scaling operations for compute typically finalize within a few minutes.
- As your storage needs expand, you have the option to provision additional storage. The Amazon Aurora engine will automatically increase the size of your database volume to accommodate growing storage requirements, up to a maximum of 128 TiB or a maximum size that you define.
- Amazon RDS Read Replicas simplify the process of elastically scaling out beyond the capacity limitations of a single DB instance, especially for read-heavy database workloads. By creating one or more replicas of a source DB instance, you can efficiently handle high-volume application read traffic from multiple copies of your data, thereby enhancing aggregate read throughput.
- Amazon RDS Proxy enhances database efficiency and enhances application scalability, security, and resilience to database failures by enabling applications to pool and share connections with the database.
- Amazon Aurora Serverless is a configuration option for Aurora that operates on an on-demand basis and offers automatic scaling capabilities.

Use Case: Cognizant

Cognizant is globally recognized as a premier provider of business and IT services, offering technology, consulting, and operational support to assist organizations in their digital transformation endeavors.

“The launch of Amazon Aurora Serverless has been highly anticipated by the market. It complements several of our offerings, including Cognizant Adaptive Data Foundation, a solution which enables the creation of cloud-based, holistic data strategies requiring insights, scale and speed”, says Arjun Varadaraja, Vice President, AI & Analytics, Cognizant.

3. Performance

- Amazon RDS Optimized Reads are engineered to deliver accelerated database performance, enabling up to 2X faster query processing without incurring any additional costs.

- To enhance the performance of your Amazon RDS database and applications, you have the option to incorporate a cache from Amazon ElastiCache, a managed, in-memory caching service, directly into your database through the Amazon RDS Console.
- Amazon RDS provides users with a selection of two distinct storage types. For most use cases, Amazon RDS General Purpose storage is suitable for database workloads. This SSD-backed storage option ensures a consistent baseline of 3 IOPS (Input/Output Operations Per Second) per provisioned GB and allows for bursting up to 3,000 IOPS above the baseline as needed.

Use Case:

Tonkean is a software-as-a-service startup specializing in simplifying the orchestration of intricate business processes for companies. Upon migrating to managed Amazon RDS for MySQL, its development team achieved time and resource savings, enabling a sharper focus on innovation and enhancing the reliability and availability of its product.

Afik Udi, Senior Manager of Production and Infrastructure – Tonkean, says “All the data that drives the engine at Tonkean is inside Amazon RDS. We rely heavily on the performance of our database and the stability of the service housing it. And we’re fully confident in AWS.”

4. Security

- Amazon RDS permits you to encrypt your databases using keys that you manage through AWS Key Management Service (KMS).
- Amazon RDS supports SSL for securing data in transit, as well as Transparent Data Encryption in SQL Server and Oracle databases to ensure data remains encrypted at rest.
- AWS strongly recommends running your database instances within Amazon Virtual Private Cloud (VPC). This enables you to isolate your database within a virtual network and establish connections with your on-premises IT infrastructure using industry-standard encrypted IPsec VPNs (Virtual Private Networks).

Use case:

Classle is a cloud-based social learning platform designed to facilitate connections between students, experts, and professionals across academic, research institutes, and industry sectors. Amazon RDS serves as both the data warehouse and transactional database for Classle's operations.

Vaidya Nathan, Founder and CEO, Classle says "The flexibility, reliability, and elasticity were the reasons for the initial decision to use AWS. Over the past two years, other services coming from AWS like Amazon RDS confirm that the decision was the right one."

5. Management and Maintenance

- AWS gives ways to access the capabilities of a production-ready relational database in minutes with either through management console or programmatically.
- Amazon RDS database instances come preconfigured with parameters and settings tailored to the selected engine and class, ensuring optimal performance and compatibility.
- Amazon RDS reduces your administrative workload by handling undifferentiated tasks like software patching, backups, provisioning, and scheduled maintenance operations. It ensures that the relational database software supporting your deployment remains current with the latest patches.
- Amazon RDS offers Amazon CloudWatch metrics for your database instances at no extra cost.
- Amazon RDS seamlessly integrates with AWS Config to bolster compliance efforts and strengthen security measures. It achieves this by logging and auditing alterations to the configuration of your

DB instance, encompassing parameter groups, subnet groups, snapshots, security groups, and event subscriptions.

- Amazon RDS Blue/Green Deployments enable you to execute database updates with enhanced safety, simplicity, and speed, all while ensuring zero data loss.
- During the promotion of your staging environment, Blue/Green Deployments halt write operations to both the blue and green environments until the switchover process is finalized.

Use Case:

Established in 2008, Airbnb, headquartered in San Francisco, operates as a community marketplace offering over 7 million accommodations and more than 40,000 unique Experiences for customers to book via the company's website or its iOS and Android applications.

Xinyao Hu, an engineering manager at Airbnb, sheds light on the Amazon RDS Replay machine developed by his team. This infrastructure enables the team to replay genuine transactions directed to nearly 100 different databases. Airbnb employs this system for purposes such as disaster recovery, load testing, and conducting advanced analytics in Hive at an extensive scale.

Conclusion

Study participants reported that they are leveraging Amazon RDS to lower the overall costs, including staff time requirements, and for running their databases while also benefiting from improved database agility, scalability, and performance, reliability.

References:

1. <https://aws.amazon.com/solutions/case-studies/intuit-mint/?pg=ln&sec=c>
2. <https://aws.amazon.com/rds/aurora/customers/>
3. Amazon RDS documentation: <https://aws.amazon.com/rds/>
4. "Comparing Cloud-based vs. On-premises Database Management Systems" by John S. Davis II and Peter Tierno (2016).
5. "Cloud Databases vs. On-Premises Databases: A Performance Comparison" by Abhijeet Khade and Neeraj Gupta (2018).
6. "Security Considerations for Cloud vs. On-Premise Databases" by Lucas Tamagna (2019).
7. "Cost Comparison: Cloud vs On-Premises" by AWS Whitepaper (2020).
8. "On-Premises vs. Cloud Database: A Comparative Study on the Basis of Performance" by Vinod Kumar Sharma and Dr. Harish Sharma (2021).
9. <https://aws.amazon.com/solutions/case-studies/cathay-pacific-rds-case-study/?pg=ln&sec=c>
10. <https://aws.amazon.com/rds/reserved-instances/>
11. <https://aws.amazon.com/rds/pricing/>
12. https://aws.amazon.com/solutions/case-studies/?customer-references-cards.sort-by=item.additionalFields.sortDate&customer-references-cards.sort-order=desc&awsf.customer-references-location=*all&awsf.customer-references-industry=*all&awsf.customer-references-use-case=*all&awsf.language=language%23english
13. https://aws.amazon.com/solutions/case-studies/tonkean-rds-case-study/?trk=em_a134p000006v19zAAA&trkCampaign=pac_q221_opensource_email1_CTA1_casestudy_tonkean&sc_channel=em&sc_campaign=%7b%7bprogram.name%7d%7d&sc_medium=em_%7b%7bcampaign.id%7d%7d&sc_outcome=PaaS_Digital_Marketing&sc_content=AWS_Webpage