**Well done! The course is completed.**

You completed the course AWS SimuLearn: Databases in Practice on 05/5/2025 at 03:51 pm.

Retake the course

ASSIGNMENT
Databases in Practice

TOTAL TIME 23:38:07

Solution Request

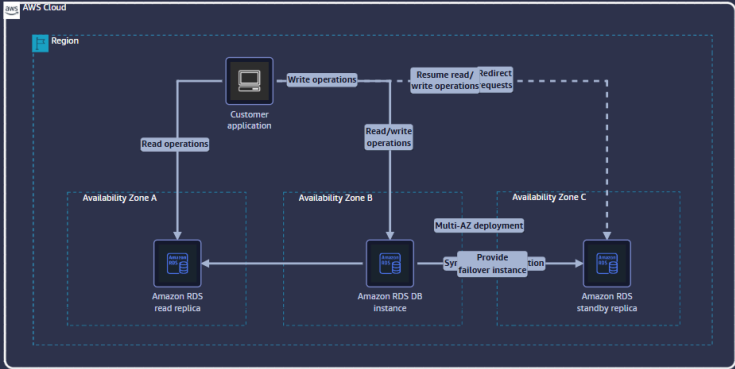
Improve the operational efficiency, availability, and performance efficiency of databases by using multiple Availability Zones and a read replica.


Practice Lab

- Learn about AWS database offerings.
- Launch an Amazon RDS instance.
- Configure a Multi-AZ deployment.
- Configure Amazon RDS backups.

DIY Goals

Create a read replica of your primary database by using a db.t3.xlarge instance.



 at skillbuilder.aws

DOWNLOAD CONTINUE

MEETING ATTENDEES

- Dr. Dee Tah, Data Scientist
- Technical Lead
- You

RDS can automatically back up your database, ensuring you can quickly recover in case of any issues. Monitoring with Amazon CloudWatch. This will allow your team to monitor database performance and health in real-time, reducing administrative overhead.

Dr. Dee Tah: Yes, Amazon RDS seems like an excellent solution to manage and scale our database infrastructure efficiently. It eliminates the need for manual setup and maintenance, provides high availability, and supports read replicas. Thank you for the recommendation.

Dr. Dee Tah: Thank you so much for the helpful information! It sounds like Amazon RDS could be the perfect solution to address my database management challenges. By using this service, I can efficiently manage and scale my MySQL database, prevent it from becoming a single point of failure, and offload read-only transactions. I'm excited to get started! Let's get the ball rolling and build this solution.

Build it on AWS!

DISCOVERED 100%

YOU DID IT!

Are you stuck? Provide the customer requirements, and I'll recommend the AWS service.

Dr. Newton

Soft skill hint: No hint available

Client Satisfaction Score: 79%

Score Dimensions: Communication Skills, Problem-Solving Skills, Customer Focus, Decision-Making Skills, Technical Knowledge

En un microservicio, aplicaría el patrón CQRS (Command Query Responsibility Segregation) separando claramente las operaciones de lectura y escritura. Para las **escrituras** (Commands), utilizaría la instancia principal de **AWS RDS**, que permite operaciones de lectura y escritura. Para las **lecturas** (Queries), aprovecharía una o varias **réplicas de lectura (read replicas)** de RDS, distribuyendo así la carga de consultas y mejorando el rendimiento del sistema. Esta separación facilita la escalabilidad y mantiene la integridad de los datos en arquitecturas basadas en microservicios.