RDS Labs Assessment Report

Lab 1 – RDS Instance Creation and Basic Operations

Q1. What are the key differences you observed between the MySQL and PostgreSQL setups?

The setup process was almost identical in the AWS console, but the default ports differed (3306 for MySQL, 5432 for PostgreSQL), and authentication usernames also varied (`admin` for MySQL vs. `postgres` for PostgreSQL). PostgreSQL also uses slightly different CLI commands (`\l`, `\c`) compared to MySQL.

Q2. How long did it take for each instance to become available?

Both MySQL and PostgreSQL took around 10–15 minutes to provision and become available.

Q3. What would happen if you chose a larger instance class?

A larger instance class would improve performance (faster queries, better handling of concurrent connections) but at a significantly higher cost per hour. For lab purposes, the smallest class was sufficient.

Lab 2 – Multi-AZ Deployment and Failover Testing

Q1. How long did the Multi-AZ enablement process take?

It took about 8 minutes for the MySQL instance to transition to Multi-AZ and return to "Available."

Q2. What was the actual failover duration during testing?

The failover during reboot with failover took about 1–2 minutes, which is consistent with AWS expected behavior.

Q3. How does Multi-AZ affect your connection string?

The connection string remains unchanged. Applications still connect via the same endpoint, with AWS handling automatic redirection to the standby during failover.

Q4. What is the cost impact of enabling Multi-AZ?

Multi-AZ roughly doubles the cost since AWS provisions a standby replica in a different Availability Zone, though storage is shared.

Lab 3 – Read Replicas and Performance Scaling

Q1. What was the replication lag between master and replica?

The observed replication lag was about 0.01 seconds, essentially near real-time for a small dataset.

Q2. What happens when you try to write to a read replica?

Write attempts fail with a read-only error, as replicas are designed strictly for read scaling and cannot accept writes unless they are promoted.

Q3. How does cross-region replication affect data transfer costs?

Cross-region replication incurs inter-region data transfer costs in addition to the replica instance cost, which can become expensive with heavy workloads.

Q4. What are the use cases for promoting a read replica?

Disaster recovery when the primary fails; migrating workloads to another region; scaling reads, then converting to a write-capable DB for load balancing; blue/green deployment strategies.

Lab 4 – Automated Backups and Point-in-Time Recovery

Q1. How does backup retention period affect storage costs?

Longer retention increases backup storage costs, as more snapshots and log files are kept for recovery.

Q2. What's the difference between manual snapshots and automated backups?

Automated backups are continuous and support point-in-time recovery within the retention window, but are deleted when the instance is deleted. Manual snapshots must be created explicitly, persist even after the instance is deleted, and are commonly used for compliance or migration.

Q3. How long did the point-in-time recovery process take?

The restore process took about 17 minutes for the new DB instance to be created and available.

Q4. What are the limitations of point-in-time recovery?

Can only restore within the configured retention period. Creates a new DB instance rather than restoring the original in-place. Recovery time may be lengthy, depending on the database size. Requires automated backups to be enabled beforehand.