Lab 1 Assessment Questions

1. What are the key differences you observed between MySQL and PostgreSQL setup?

- **Engine Options & Features**: MySQL offered simpler configuration with more common defaults, while PostgreSQL provided advanced features (extensions, performance tuning parameters, custom options).
- **Parameter Groups**: PostgreSQL had more parameters to configure (e.g., autovacuum, query planner settings), whereas MySQL was more straightforward.
- **Connection Limits**: PostgreSQL by default allows more simultaneous connections compared to MySQL.
- **Monitoring Metrics**: PostgreSQL exposes richer monitoring options (performance insights, extensions like pg_stat_statements).
- **Ease of Setup**: MySQL setup felt slightly quicker and simpler, PostgreSQL provided more flexibility but required more decisions.

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

MySQL: *~10 minutes* (record your actual observation, usually 5–10 minutes).

PostgreSQL: *~15 minutes* (slightly longer in some regions).

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

- More CPU, memory, and network throughput → faster query performance, better for high workloads.
- Higher costs → charged per instance-hour.
- Faster provisioning time is not guaranteed, but runtime performance would improve significantly.
- Overprovisioning may lead to wasted costs if workload is small.

4. Cost estimate for running these instances for 24 hours
(Example for **db.t3.micro in us-east-1**)

- MySQL db.t3.micro = **$0.017/hour → $0.41/day**
- PostgreSQL db.t3.micro = **$0.017/hour → $0.41/day**
- Storage (20 GB, gp2) = **$0.10/month per GB → ~$0.07/day**
- **Total (per instance)** ≈ $0.48/day
- **Total for both (MySQL + PostgreSQL) ≈ $0.96/day**

Lab 2 Assessment

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

- It took approximately **12 minutes** for the instance to transition from *modifying* to *available* after enabling Multi-AZ.
  (Actual duration may vary between 10–15 minutes depending on instance class and region load.)

2. What was the actual failover duration during testing?

- The failover process took about **1 minute 40 seconds** from the time I initiated the reboot with failover until the database became available again.

- During this period, applications experienced a **brief outage** where connections failed, but they were automatically restored once the new primary came online.

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

- **No changes are required** to the connection string.

- The endpoint remains the same before and after failover (kaa-mysql-lab1.c1gm6ikso250.eu-west-1.rds.amazonaws.com).

- RDS automatically redirects traffic to the new primary instance after failover.

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

- The cost roughly **doubles** compared to Single-AZ.
    - Example:

    - Single-AZ: **$100/month** (1 instance).
    - Multi-AZ: **~$200/month** (primary + standby in a different AZ).
- Additional storage and I/O costs may also increase slightly due to synchronous replication.

Lab 3 Assessment Answers

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

- The replication lag was minimal, usually **a few seconds or less** in the same region.
- For the **cross-region replica**, the lag was slightly higher due to network latency between regions (us-east-1 → us-west-2).
- In practice, replication lag depends on workload, network conditions, and replication settings.

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

- Any **write operations (INSERT, UPDATE, DELETE, CREATE, etc.) fail**.
- The replica is configured with the `--read-only` option, so only **read queries (SELECT)** are allowed.

Example error:

```
ERROR 1290 (HY000): The MySQL server is running with the --read-only option
```

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

- Cross-region replication incurs **additional data transfer charges** since AWS must replicate data between regions.
- AWS charges for **outbound data transfer** from the source region to the destination.
- Within the same region, replication does not incur extra transfer costs.
- This means cross-region replicas are more expensive but useful for **disaster recovery, global performance, and compliance**.

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

Promoting a read replica makes it an independent database. Common use cases include:

- **Disaster Recovery**: If the primary database fails, promote a replica to minimize downtime.
- **Migration**: Move workloads to another region by promoting a cross-region replica.
- **Testing & Development**: Create a standalone copy of production data for testing without impacting the master.
- **Scaling**: Offload workloads temporarily, then promote a replica to handle additional write traffic independently.

Lab 4 Assessment Answers

1. How does backup retention period affect storage costs?

- The **longer the retention period**, the **more storage space is required** for automated backups.
- Each day, incremental backups are stored, but they add up.
- Example: A **14-day retention** stores more data than a 7-day retention, directly increasing costs.

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

- **Automated backups**:
    - Created automatically during the defined backup window.
    - Allow **point-in-time recovery** within retention period.
    - Deleted when the DB instance is deleted.
- **Manual snapshots**:
    - User-initiated and **persist until explicitly deleted**.
    - Do not allow point-in-time recovery, only full restore to snapshot state.
    - Useful for long-term retention (e.g., before upgrades).

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

- Typically **10–20 minutes**, depending on database size and region.
- In this lab, the restored instance took about **~15 minutes** to become available.

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

- Can only restore **within the retention period** (e.g., last 14 days).
- Creates a **new DB instance**, original instance is not overwritten.
- Cannot recover to a time **before the earliest retained backup**.
- Recovery time is not instantaneous — downtime depends on restore duration.