Potential disadvantages compared to running PostgreSQL on-premise:

1. Cost
   - Ongoing Costs: AWS RDS involves ongoing costs for instance usage, storage, backup storage, and data transfer, which can be higher than the one-time hardware costs for on-premise setups.
   - Cost Management: Predicting and managing costs can be challenging due to variables like I/O operations and data transfer.

2. Limited Customization and Control
   - Configuration Restrictions: AWS RDS imposes restrictions on certain configurations and extensions, which might limit customization compared to full control on-premise.
   - Superuser Access: You don't get full superuser (root) access to the database, which can limit advanced configurations and optimizations.

3. Performance Overhead
   - I/O Latency: Network latency can affect performance, especially if your application and the database are not in the same AWS region.
   - Resource Contention: Shared infrastructure can sometimes lead to resource contention, impacting performance.

4. Vendor Lock-In
   - Migration Complexity: Moving from AWS RDS to another cloud provider or back to on-premise can be complex and time-consuming, leading to potential vendor lock-in.

5. Compliance and Security
   - Data Sovereignty: Regulatory requirements might mandate data to be stored on-premise, especially for sensitive or classified data.
   - Shared Responsibility: While AWS provides security for the infrastructure, you're still responsible for database security, including encryption and access management.

6. Network Dependency
   - Internet Dependency: Access to your database is dependent on internet connectivity. Any issues with your internet connection can affect access to the database.
   - Bandwidth Costs: High data transfer can lead to significant bandwidth costs.

7. Backup and Recovery Constraints
   - Backup Control: While AWS RDS automates backups, it may not provide the same level of control or customization as on-premise solutions.
   - Restore Time: Restoring large databases can take significant time, which might not meet your RTO (Recovery Time Objective).

8. Upgrades and Patching
   - Automatic Updates: AWS RDS handles maintenance tasks, including patching and minor version upgrades, which can sometimes cause disruptions if not planned properly.
   - Downtime: Maintenance windows and forced updates can lead to unplanned downtime.

9. Feature Lag
    - Delayed Availability: New PostgreSQL features and versions might be available later on AWS RDS compared to on-premise setups, as AWS needs time to integrate and test new releases.

10. Environment Constraints
    - Testing and Development: On-premise setups allow for more flexibility in creating isolated environments for testing and development, while AWS RDS environments might have limitations or additional costs.