Encryption

1. Introduction
   1. Encryption in Amazon RDS (Relational Database Service) provides enhanced security by protecting data at rest and in transit
   2. RDS encryption uses the AWS Key Management Service (KMS) to encrypt your database, backups, snapshots, and read replicas.
2. Purpose of RDS Encryption
   1. Protect sensitive data in the database by making it inaccessible without proper authorization
   2. Ensure compliance with industry standards, regulatory requirements, and security best practices
   3. Secure backups and snapshots associated with the encrypted instance
   4. Encrypt data in transit to prevent unauthorized access while data moves within and outside the AWS network
3. How RDS Encryption Works
   1. Database Storage — Data stored in the database's storage is encrypted
   2. Automated Backups and Snapshots — All automated backups and manual snapshots are also encrypted with the same encryption key
   3. Read Replicas — If you create read replicas of an encrypted DB instance, those replicas are also encrypted
   4. Database Logs — Logs generated from an encrypted RDS instance are also encrypted
4. Setting Up Encryption
   1. RDS encryption must be enabled when creating a new DB instance.
   2. It cannot be added to an existing, non-encrypted DB instance
      1. Create an unencrypted snapshot of your existing DB instance
      2. Copy the snapshot with encryption enabled
      3. Restore the encrypted snapshot to a new encrypted DB instance
   3. You select an AWS KMS key (either the default key managed by AWS or a customer-managed key) when setting up encryption
   4. Default AWS KMS Key — AWS automatically manages a default key, which is convenient if you want encryption without managing your own keys
   5. Customer-Managed Key
      1. If you require additional control, you can create and manage your own KMS keys, defining specific permissions and access policies.
      2. This option gives you more flexibility and security but requires additional management
5. Data in Transit Encryption
   1. In addition to encryption at rest, RDS also supports encryption for data in transit.
   2. This feature uses SSL/TLS encryption to secure data transmitted between your applications and RDS instances
   3. RDS provides a certificate for SSL/TLS connections
   4. You configure your application to use SSL/TLS, ensuring that data sent between the app and RDS is encrypted
6. Considerations and Limitations
   1. Cannot Encrypt Existing Instances
   2. Performance Impact