What is Database:

A database is a structured way to store, organize, and manage data so it can be easily accessed, updated, and analyzed. Think of it as a digital filing system where data is stored in tables, rows, and columns, like a spreadsheet. It helps applications and users retrieve or manipulate data efficiently.

Examples: Storing customer details for an e-commerce site or keeping track of bank transactions.

Types of Databases

1. **Relational Databases (SQL)**:
    - Stores data in tables with rows and columns.
    - Uses Structured Query Language (SQL) for managing data.
    - Examples: MySQL, PostgreSQL, Oracle.
2. **NoSQL Databases**:
    - Stores unstructured or semi-structured data (JSON, key-value pairs, documents).
    - Flexible schema for dynamic data.
    - Examples: MongoDB, DynamoDB.

Why Use a Database?

1. **Efficiency**:
    - Faster access and retrieval of data compared to files or spreadsheets.
2. **Scalability**:
    - Handles large volumes of data seamlessly.
3. **Data Integrity**:
    - Ensures accuracy and consistency with rules and relationships.
4. **Multi-User Access**:
    - Supports concurrent access by multiple users.
5. **Security**:
    - Provides advanced security controls to protect sensitive data.
6. **Backup and Recovery**:
    - Easy to back up and recover in case of failure.
7. **Querying and Analysis**:
    - Simplifies complex data searches and analytics with languages like SQL.

Databases are more robust, reliable, and efficient for structured, large, or growing datasets.

AWS RDS (Relational Database Service) Overview:

What is AWS RDS?

- AWS RDS is a **managed database service** that simplifies the setup, operation, and scaling of relational databases in the cloud.
- Supports various database engines:
  - **Amazon Aurora** (MySQL and PostgreSQL compatible)
  - **MySQL**
  - **PostgreSQL**
  - **MariaDB**
  - **Oracle**
  - **Microsoft SQL Server**

Why Use AWS RDS Instead of Setting Up Databases on EC2?

1. **Ease of Management**:
   - Fully managed service: No manual installation, patching, or backups.
   - Automated backups, snapshots, and point-in-time recovery.
2. **Scalability**:
   - Vertical scaling: Change instance types for more resources.
   - Horizontal scaling: Read replicas for improved read performance.
3. **High Availability**:
   - Multi-AZ deployments for automatic failover.
   - Automatically manages standby and primary instances.
4. **Performance**:
   - Optimized storage options like **General Purpose SSD**, **Provisioned IOPS SSD**.
   - Amazon Aurora offers superior performance compared to traditional MySQL/PostgreSQL.
5. **Security**:
   - Built-in encryption for data at rest (using AWS KMS) and in transit (using SSL/TLS).
   - Integration with AWS IAM for fine-grained access control.
   - Network isolation using VPC.
6. **Cost-Effectiveness**:
   - Pay-as-you-go model.
   - Saves costs by reducing administrative overhead.
   - No need to provision and maintain EC2 instances for database hosting.

7. **Monitoring and Metrics**:
    - Built-in monitoring via **Amazon CloudWatch**.
    - Database performance insights for query optimization.

Key Features of AWS RDS

1. **Database Engines**:
    - Choose from six popular relational database engines.
2. **Automated Backups**:
    - Daily snapshots and transaction logs enable recovery to any point within the retention period.
3. **Multi-AZ Deployments**:
    - Increases database availability with automatic failover.
4. **Read Replicas**:
    - Improves read performance by distributing read queries to replicas.
5. **Storage Options**:
    - **General Purpose SSD**: Balanced performance and cost.
    - **Provisioned IOPS SSD**: High-performance, low-latency workloads.
    - **Magnetic Storage**: Cost-effective for infrequent access.
6. **Maintenance**:
    - Automatic patching of the database engine.
7. **Security**:
    - Database encryption using AWS Key Management Service (KMS).
    - Access control via IAM and security groups.
8. **Support for VPC**:
    - Database instances can run in a Virtual Private Cloud for network isolation.
9. **Monitoring**:
    - CloudWatch metrics, alarms, and logs for detailed performance insights.

When to Use AWS RDS?

- **Production workloads** requiring high availability and durability.
- Applications needing managed database solutions to minimize administrative overhead.
- Workloads needing scalability for growing traffic.
- Scenarios requiring encryption for data compliance (e.g., healthcare, finance).
- Applications with heavy read workloads (use Read Replicas).

Key Differences Between RDS and Database on EC2

Feature	AWS RDS	Database on EC2

Management	Fully managed by AWS	Fully managed by you
Setup	Simple via console or CLI	Manual installation/configuration
Scalability	Auto-scaling supported	Manual scaling
Backups	Automated backups	Manual backups
High Availability	Built-in Multi-AZ support	Requires custom setup
Performance Tuning	Pre-configured optimizations	Fully manual
Cost	Pay for usage	Pay for EC2, storage, and management

Considerations When Using RDS

1. **Limitations**:
   - No OS-level access (unlike EC2).
   - Limited customization of database parameters compared to on-prem databases.
2. **Use Cases**:
   - **Transactional systems**: E-commerce, banking.
   - **Read-heavy applications**: Blogs, analytics.
   - **SaaS platforms** needing managed databases.
3. **Advanced Features**:
   - **Aurora Global Database**: Globally distributed workloads.
   - **RDS Proxy**: Connection pooling to improve app performance and availability.

RDS Best Practices

- Use **Multi-AZ deployments** for critical applications.
- Enable **automated backups** and **retention periods** for disaster recovery.
- Monitor **CloudWatch metrics** to optimize performance.
- Secure access using **IAM roles**, **security groups**, and **SSL/TLS**.
- Use **Read Replicas** for scaling reads without impacting the primary database.

Database setup In EC2:

```
$ sudo apt install mysql-server
```

```
$ sudo systemctl start mysql.service
$ mysql -u root -p
```

Create a DB:

```
$ CREATE DATABASE my_database;
```

```
$ USE my_database;
```

```
$ CREATE TABLE users ( user_id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), date_of_birth DATE );
```

```
$ INSERT INTO users (first_name, last_name, email, date_of_birth) VALUES ('John', 'Doe', 'john.doe@example.com', '1990-01-01');
```

```
$ SELECT * FROM users;
```

```
$ INSERT INTO users (first_name, last_name, email, date_of_birth) VALUES ('John', 'Doe', 'john.doe@example.com', '1990-01-01');
```

```
$ EXIT;
```

Connect to RDS from an EC2:

```
$ sudo apt-get install mysql-client
```

```
$ mysql -h database-1.c9y608sqq15u.us-east-1.rds.amazonaws.com -u admin -p
```

–x–

Laxmikanat Giri