

Amazon Relational Database Service (RDS)

- Relational databases are databases that organize stored data into tables. The associated tables have defined relationships between them.
- Amazon RDS is a **fully managed** database service for relational databases. This means that access to the underlying operating system is not allowed and software patches and management are handled by AWS.
- Databases Supported By RDS:
 - MySQL
 - PostgreSQL
 - Oracle
 - SQL (MS SQL Server)
 - Aurora
- What is Aurora?
 - Home grown Relational Database forked, and **fully compatible with MySQL**. It has five times better performance than MySQL and a lower price point than commercial databases.

RDS

- RDS is a fully managed Relational Database Service in the cloud
 - Does not allow access to the underlying operating system
 - Can connect to the database server itself as normal (i.e MySQL command line)
 - Ability to provision/resize hardware on demand for scaling
 - Multi-AZ deployments
 - Read Replicas (MySQL/PostgreSQL/Aurora)
- Currently supported Database engines
 - MySQL
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - Aurora
- Ability to provision/resize hardware on demand for scaling

RDS

- Instances
- Disk space Minimum 5GB | Maximum 3TB
- SSD vs. Provisioned IOPS
- Benefits of running RDS instead of your own instance
 - Automatic minor updates
 - Automatic backups
 - Not required to managed operating system
 - Multi-AZ with a single click
 - Automatic recovery in event of a failover

RDS

- Automatic AZ Failover, Multi-AZ synchronous replicates data to the backup instance located in another availability zone
 - Availability zone outage
 - Primary DB instance fails
 - Instance server type is changed
 - Manual failover initiated
 - Updating software version
- Backups are taken against the stand-by instance to reduce I/O freezes and slow down IF multi-az is enabled

RDS

- Backups: AWS provides automated point in time backups against the RDS database instance
 - Automated backups are deleted once the database instance is deleted and cannot be recovered
 - Backups on database engines only work correctly when the database engine is “transactional” but do currently work for all supported database types
 - MySQL requires InnoDB for reliable backups

- Read Replicas
 - MySQL/PostgreSQL/Aurora currently support
 - Uses native replication on by MySQL/PostgreSQL
 - Read replicas can be created from other read replicas
 - Multiple read replicas can have the same source
 - Read replicas allow for elasticity in RDS
 - Monitor replication lag using CloudWatch
 - Only supports InnoDB MySQL storage engine
 - Offload database tasks off of production
 - Can promote a read replica to a primary instance
 - MySQL
 - Replicate for importing/exporting data to RDS
 - Can replicate across regions

- Read Replicas: When to use them
 - High non-cached database read traffic (elasticity)
 - Running business function such as data warehousing
 - Importing/Exporting data into RDS
 - Rebuilding indexes
 - Ability to promote a read replica to a primary instance