**RDS**

* Relational Database Service(RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud
* RDS is an Online Transaction Processing(OLTP) type of database
* Primary use case is a transactional database(rather than analytical)
* Best for structured, relational data store requirements
* Aims to be drop-in replacement for existing on-premise instances of the same databases
* Automated backups and patching applied in customer-defined maintenance windows
* Push-button scaling, replication and redundancy
* RDS supports the following database engines
  + Aurora
  + MySQL
  + MariaDB
  + Oracle
  + SQL Server
  + PostgreSQL
* RDS is a managed service and you do not have access to the underlying EC2 instance(no root access)
* The RDS service includes the following:
  + Security and patching of the DB instances
  + Automated backup for the DB instances
  + Software updates for the DB engine
  + Easy scaling for storage and compute
  + Multi-AZ option with synchronous replication
  + Automatic failover for Multi-AZ option
  + Read replicas option for read heavy workloads
* A DB instance is a database environment in the cloud with the compute and storage resources you specify
* Database instances are accessed via endpoints
* Endpoints can be retrieved via the DB instance description in the AWS Management Console, **DescribeDBInstances** API or **describe-db-instances** command
* By default, customers are allowed to have up to a total of 40 RDS instances(only 10 of these can be Oracle or MS SQL unless you have your own licenses)
* Maintenance windows are configured to allow DB instances modifications to take place such as scaling and software patching(some operations require the DB instance to be taken offline briefly)
* You can define the maintenance window or AWS will schedule a 30 minute window
* Windows integrated authentication for SQL only works with domains created using the AWS directory service - need to establish a trust with an on-premise AD directory
* Events and Notifications
  + RDS uses SNS to send RDS events via SNS notifications
  + You can use API calls to the RDS service to list RDS events in the last 14 days(DescribeEvents API)
  + You can view events from the alst 14 days using the CLI
  + Using the AWS Console you can only view RDS events for the last 1 day
* Use Cases, Alternatives and Anti-Patterns
  + The table below provides guidance on when best to use RDS and several other AWS database/data store services

Table

Description automatically generated

* Alternative to Amazon RDS
  + if your use case isn't suppored on RDS you can run databases on EC2
  + Consider the following points when considering DB on EC2
    - You can run any database you like with full control and ultimate flexibility
    - You must manage everything like backups, redundancy, patching and scaling
    - Good option if you require a database not yet supported on RDS such as IBM DB2 or SAP HANA
    - Good option if it's not feasible to migrate to AWS-managed database
* Anti-Patterns
  + Anti-patterns are certain patterns in architecture or development that are considered bad, or sub-optimal practices - ie there may be a better service of method to produce the best result
  + The following table describes requirements that are not a good fit for RDS
    - Table

      Description automatically generated
* Encryption
  + You can encrypt your RDS instances and snapshots at rest by enabling the encryption option for your RDS DB instance
  + Encryption at rest is supported for all DB types and uses KMS
  + When using encryption at rest the following elements are also encrypted:
    - All DB snapshots
    - Backups
    - DB instance storage
    - Read Replicas
  + You cannot encrypt an existing DB, you need to create a snapshot, copy it, encrypt the copy then build an encrypted DB from the snapshot
  + Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, Read Replicas and snapshots
  + A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same region
  + If the master and Read Replica are in different regions, you encrypt using the encryption key for that region
  + You can't have an encrypted Read Replica of an unencrypted DB instance or an uinecrypted Read Replica of an encrypted DB instance
  + Encryption/decryption is handled transparantly
  + RDS supports SSL encrtyption between applications and RDS DB instances
  + RDS generates a certificate for the instance
* DB Subnet Groups
  + A DB subnet group is a collection of subnets(typically private) that you can create in a VPC and that you then desginate for your DB instances
  + Each DB subnet should have subnets in at least two AZs in a given region
  + It is recommended to configure a subnet group with subnets in each AZ even for standalone instances
  + During the creation of an RDS instance you can select the DB subnet group and the AZ within the group to place the RDS DB instance in
  + You cannot pick the IP within the subnet that is allocated
* Bills and Provisioning
  + AWS Charge for:
    - DB instance hours(partial hours are charged as full hours)
    - Storage GB/month
    - I/O requests/month - for magnetic storage
    - Provisioned IOPS/month - for RDS provisioned IOPS SSD
    - Egress data transfer
    - Backup storage(DB backups and manual snapshots)
  + Backup storage for the automated RDS backup is free of charge up to the provisioned EBS volume size
  + However, AWS replicates data across multiple AZs and so you are charged for extra storage space on S3
  + For multi-AZ you are charged for
    - Multi-AZ DB hours
    - Provisionied storage
    - Double write I/Os
  + For multi-AZ you are not charged for DB data transfer during replication from primary to standby
  + Oracle and Microsoft SQL licenses are included or you can bring your own(BYO)
  + On-demand and reserved instance pricing is available
  + Reserved instances are defined base on the following attributes which must not be changed:
    - DB engine
    - DB instance class
    - Deployment type
    - Licemnse model
    - Region
  + Reserved instances
    - Can be moved between AZs in the same region
    - Are available for multi-AZ deployments
    - Can be applied to Read Replicas if DB instance class and region are the same
    - Scaling is achieved through changing the instance class for compute, and modifying storage capacity for additional storage allocation
  + Scalability
    - You can only scale RSD up(compute and storage)
    - You cannot decrease the allocated storage for an RDS instance
    - You can scale storage and change the storage type for all DB engines except MS SQL
    - For MS SQL the workaround is to create a new instance from a snapshot with the new configuration
    - Scaling storage can happen while the RDS instance is running without outage however there may be performance degradation
    - Scaling compute will cause downtime
    - You can choose to have changes take effect immediately, however the default is within the maintenance window
    - Scaling requests are applied during the specified maintenance window unless 'apply immediately' is used
    - All RDS DB types support a maximum DB size of 64 TiB except for SQL Server 16 TiB
  + Performance
    - RDS uses EBS volumes(never instance store) for DB and log storage
    - There are three storage types available: General Purpose(SSD), Provisioned IOPS(SSD) and Magnetic
    - General Purpose(SSD)
      * Use for Database workloads with moderate I/O requirement
      * Cost effective
      * Also called gp2
      * 3 IOPS/GB
      * Burst up to 3000 IOPS
    - Provisioined IOPS(SSD):
      * Use for I/O intensive workloads
      * Low latency and consistent I/O
      * User specified IOPS
    - For provisioined IOPS storage the table below shows the range of Provisioined IOPS and storage size range for each database engine
  + Table

    Description automatically generated
    - Magnetic:
      * Not recommended anymore, available for backwards compatibility
      * Doesn't allow you to scale storage when using the SQL Server database engine
      * Doesn't support elastic volumes
      * Limited to a maximum size of 4 TiB
      * Limited to a max of 1,000 IOPS
  + Multi-AZ and Read Replicas
    - Multi-AZ and Read Replicas are used for high availability, fault tolerance and performance scaling
    - The table below compares multi-AZ deployments to Read Replicas
  + Graphical user interface, text, application

    Description automatically generated
  + Multi-AZ
    - Multi-AZ RDS creates a replica in another AZ and synchronously replicates to it(DR only)
    - There is an option to choose multi-AZ during the launch wizard
    - AWS recommends the use of provisioned IOPS storage for multi-AZ RDS DB instances
    - Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable
    - You cannot choose which AZ in the region will be chosen to create the standby instance
    - You can view which AZ the standby DB instance is created in
    - A failover may be triggered in the following circumstances:
      * Loss of primary AZ or primary DB instance failure
      * Loss of network connectivity on primary
      * Compute(EC2) unit failure on primary
      * Storage(EBS) unit failure on primary
      * The primary DB instance is changed
      * Patching of the OS on the primary DB instance
      * Manual failover(reboot with failover selected on primary)
    - During failover RDS automatically updates configuration(including DNS endpoint) to use the second node
    - Depending on the instance class it can take 1 to a few minutes to failover to standby DB instance
    - It is recommended to implement DB connection retries in your application
    - Recommended to use the endpoint rather than the IP address to point applications to the RDS DB
    - The method to initiate a manual RDS DB instance failover is to reboot selecting the option to failover
    - A DB instance reboot is required for changes to take effect when you change the DB parameter group or when you change a static DB parameter
    - A DB parameter group is a configuration container for the DB engine configuraiton
    - You will be alerted by a DB instance event when a failover occurs
    - The secondary DB in a multi-AZ configuration cannot be used as an independed read node(read or write)
    - There is no charge for data transfer between primary and secondary RDS instances
    - Multi-AZ deployments for the MySQL, MariaDB, Oracle and PostgreSQL engines use Amazon's failover technology
    - Multi-AZ deployments for the SQL Server engine use SQL Server Database Mirroring(DBM)
    - System upgrades like OS patching, DB instance scaling and system upgrades are applied first on the standby, before failing over and modifying the other DB instance
    - In multi-AZ configurations snapshots and automated backups are performed on the standby to avoid I/O suspension on the primary instance
    - Read Replica Support for Multi-AZ:
      * RDS Read Replicas for MySQL and MariaDB support Multi-AZ deployments
      * Combining Read Replicas with Multi-AZ enables you to build a resilient DR strategy and simplify your database engine upgrade process
      * **A RR in a different region that the source database can be used as a standby database and promoted to become the new production database in case of a regional disruption**
      * This allows you to scale reads whilst also having multi-AZ for DR
    - The process for implementing maintenance activities are as follows
      * Perform operations on standby
      * Promote standby to primary
      * Perform operations on new standby(demoted primary)
    - You can manually upgrade a DB instance to a supported DB engine version from the AWS Console
    - By default upgrades will take effect during the next maintenance window
    - You can optionally force an immediate upgrade
    - In multi-AZ deployments version upgrades will be conducted on both the primary and standby at the same time causing an outage of both DB instance
    - Ensure security groups and NACLs will allow your application servers to communicate with both the primary and standby instances
  + Graphical user interface, application, Teams

    Description automatically generated
  + Read Replicas
    - Read replicas are used for read heavy DBs and replication is asynchronous
    - Read replicas are for workload sharing and offloading
    - Read replicas provide read-only DR
    - Read replicas are created from a snapshot of the master instance
    - Must have automated backups enabled on the primary(retention period>0)
    - Only supported for transactional database storage engine(InnoDB not MyISAM)
    - **Read replicas are available for MySQL, PostgreSQL, MariaDB, Oracle, and Aurora(not SQL Server)**
    - For the MySQL, MariaDB, PostgreSQL and Oracle database engines, RDS creates a second DB instance using a snapshot of the source DB instance
    - It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance
  + **Aurora** employs an SSD-backed virtualization storage layer prupose-built for database workloads
    - You can take snapshots of PostgreSQL read replicas but cannot enable automated backups
    - You can enable automatic bakcups on MySQL and MariaDB read replicas
    - You can enable writes to the MySQL and MariaDB Read Replicas
    - **You can have 5 read replicas of a production DB**
    - You cannot have more than four instances involved in a replication chain
    - You can have read replicas of read replicas for MySQL and MariaDB but not for PostgreSQL
    - Read replicas can be configured from the AWS Console or the API
    - You can specify the AZ the read replica is deployed in
    - The read replicas storage type and instance class can be different from the source but the compute should be at least the performance of the source
    - You cannot change the DB engine
    - In a multi-AZ failover the read replicas are switched to the new primary
    - Read replicas must be explicitly deleted
    - If a source DB instance is deleted without deleting th replicas, each replica becomes a standalone single-AZ DB instance
    - You can promote a read replica to a primary
    - Promotion of read replicas takes several minutes
    - Promoted read replicas retain:
      * Backup retention window
      * Backup window
      * DB parameter group
    - Existing read replicas continue to function as normal
    - Each read replica has its own DNS endpoint
    - Read replicas can have multi-AZ enabled and you can create read replicas of multi-AZ source DBs
    - Read replicas can be in another region(uses asynchronous replication)
    - This configuration can be used for centralizing data from across different regions for analytics
    - Graphical user interface, diagram, application

      Description automatically generated
  + DB Snapshots
    - DB Snapshots are user-initiated and enable you to back up your DB instance in a known state as frequently as you wish, and then restore to that specific state
    - Cannot be used for point-in-time recovery
    - Snapshots are stored in S3
    - Snapshots remain on S3 until manually deleted
    - Backups are taken within a defined window
    - I/O is brefly suspended while backups initialize and may increase latency(applicable to single AZ RDS)
    - DB snapshots that are performed manually will store even after the RDS instance is deleted
    - Restored DBs will always be a new RDS instance with a new DNS endpoint
    - Can restore up to the last 5 minutes
    - Only default DB parameters and security groups are restored - you must manually associate all other DB parameters and SGs
    - It is recommended to take a final snapshot before deleting an RDS instance
    - Snapshots can be shared with other AWS accounts
  + High Availabilty Approaches for Databases
    - If possible, choose DynamoDB over RDS because of inherent fault tolerance
    - If DynamoDB can't be used, choose Aurora because of redundancy and automatic recovery features
    - If Aurora can't be used, choose Multi-AZ RDS
    - Frequent RDS snapshots can protect against data corruption or failure and they won't impact performance of Multi-AZ deployment
    - Regional replication is also an option, but will not be strongly consistent
    - If the database runs on EC2, you have to design the HA yourself
  + Migration
    - AWS Database Migration Service helps you migrate databases to AWS quickly and securely
    - Use along with the Schema Conversion Tool(SCT) to migrate databses to AWS RDS or EC2-based databases
    - The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database
    - The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases
    - SCT can copy databse schemas for homogenous migrations(same database) and convert schemas for heterogenous migrations(different database)
    - DNS is used for smaller, simpler conversions and also supports MongoDB and DynamoDB
    - SCT is used for larger more complex datasets like data warehouses
    - DMS has replication functions for on-premise to AWS or to Snowball or S3