

**Data**: Refers to all the single items that are stored in a database, either individually or as a set. Data in a database is primarily stored in database tables, which are organized into columns that dictate the data types stored therein.

**Database**: is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

**RDBMS**: A relational database management system (RDBMS or just RDB) is a common type of database that stores data in tables, so it can be used in relation to other stored datasets.

Examples of the most popular RDBMS are MYSQL, Oracle, IBM DB2, and Microsoft SQL Server database.

**NoSQL** : NoSQL databases (aka "not only SQL") are **non-tabular databases and store data differently than relational tables**. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph.

**Types of NoSQL or Non-Relational Databases:**

1. **Columnar Database:** - Cassandra , HBase
2. **Document Database(Script written in JSON or XML):** Mongo DB, Couch DB, Raven DB
3. **Key Value Database**: Redis, Dynamo DB, Tokyo Cabinet
4. **Graph Based Database**: Neo4J, Flock DB

**AWS Fully Managed Database Services:**

**In this type of AWS service is responsible for:**

1. Security and Patching
2. Automated Backup
3. Software updates for the DB engine.
4. AWS will take the responsibility to automatically redirect from primary DB to Stand-by : or simply Automatic Failover.
5. The stand by DB should not be in same AZ or different region. Stand by server should only be created at another AZ.
6. If you do not want to use the Standby DB than you can choose, Stand-alone DB, where you cannot create any Standby DB.
7. By default every DB has a weekly maintenance window. Retention Period(default: 1 to 35days in between)

**Settings managed by the Users:**

1. Managing DB settings.
2. Creating Relational DB Schema.
3. Database performance Tuning.

**Relational DB Engine Options in AWS:**

1. MS SQL server
2. MySQL -> Support 64TB of DB
3. Oracle
4. AWS Aurora -> High Throughput
5. Postgre SQL -> Highly reliable and stable
6. Maria DB -> MySQL compatible, up to 64TB

**Licensing Options:**

1. BYOL 🡪 Bring your own licence.
2. Licence included 🡪 from AWS, charged on hourly basis.

**RDS Limits:**

1. Up to 40 DB instances can be created per account.
2. 10 among these 40 can be Oracle or MS-SQL server under Licence included model.
3. Under BYOL model all 40 DB instances can be any DB engine you need.
4. All AWS RDS type instances are created in EBS volumes only for DB and log storage. So we cannot create an RDS instance to S3 buckets or to EFS.

**RDS instance Storage:**

1. RDS instance storage types can be the following:
   1. **General Purpose**: If your requirement is normal IOPS and minimal throughput.

Limit: Memory 🡪 min 20GB, max 🡪 16384 GB

* 1. **Provisional IOPS RDS Storage**: Use for high performance like OLTP (online transaction processing) workloads. Like banking transactions etc…

Limit: Memory 🡪 min 100GB, max 🡪 16384GB

**Templates Available in RDS:**

**Production**: Use defaults for high availability and fast, consistent performance.

**Dev/Test**: Outside of production envt.

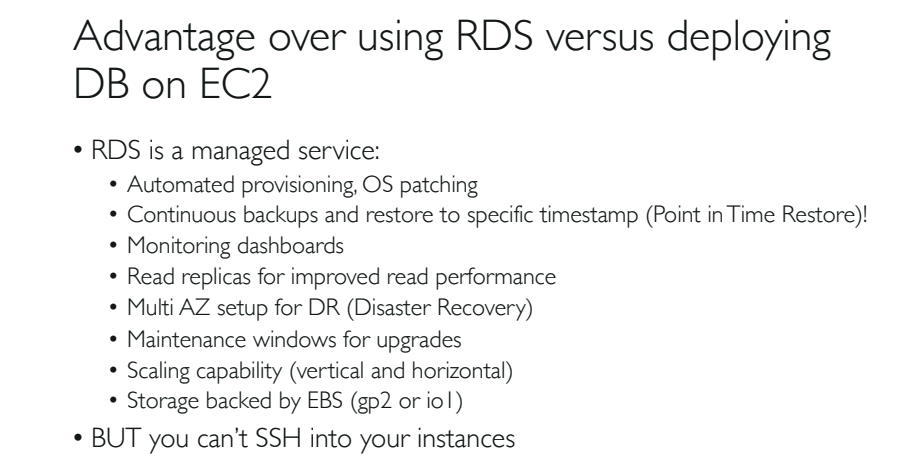
**Free Tier**: For free tier account holders. No Multi-AZ option not available means no standby option. Cannot select instance type.

**DB Instance Size:**

**Standard Class:** Includes m-class instances, max-46cpu, 384GB ram, EBS 14000mbps.

**Memory Optimized Class:** Includes r and x class instances, max- 96cpu, max-768GB ram, EBS 14000mbps.

**Burstable Class:** includes t class instances, max-8 vcpu, 32GB ram, EBS 1500mbps



**Multi AZ in RDS:**

1. We can select Multi-AZ option during RDS DB instance launch. We cannot enable this option once the RDS is created.
2. The main use of enabling Multi-AZ in RDS is that, we will have high availability and fault tolerance.
3. For example, if you have created an RDS with Multi-AZ option enabled in Mumbai region, than, Primary RDS will be created in any one AZ, let us say in 1a, and Stand-by AZ will be created in 1b or in 1c.
4. You do not have any option to select the location of standby RDS. It can be created on 1b or 1c.

**NOTE**: In RDS, primary and standby RDS will be created in same region but in different AZ’s.

1. Changes or updates made on primary RDS will automatically synchronize with the Stand-by RDS.
2. User does not have control over Standby RDS.
3. User cannot select the AZ or region of where to create the Stand-by RDS. AWS takes full control of it. However you can view the AZ of Standby server.
4. It can take 1 to few minutes to take data access on standby RDS, in cases of failover. From now onwards standby RDS will become Primary RDS.

**NOTE**: AWS recommends the use of ‘Provisioned IOPS’ instances for Multi-AZ RDS.

**In which situations Multi-AZ RDS triggers failover?**

1. In case of failure of Primary DB instance.
2. In case of AZ failure.
3. Loss of Network connectivity in Primary DB.
4. Loss of Primary EC2 instance failure.
5. EBS failure of Primary DB instance.
6. The primary DB instance settings are changed.
7. Patching the OS of the DB instance.
8. Manual Failover (ex: rebooting, maintenance etc.)

**Steps to take when Multi-AZ failover occurs:**

1. During failover, the CNAME of the RDS DB is updated to map the standby RDS ip address(do not use ip address of your DB RDS in programming, instead use the CNAME).
2. CNAME – Canonical Name Record.

**When do we perform Manual Failover?**

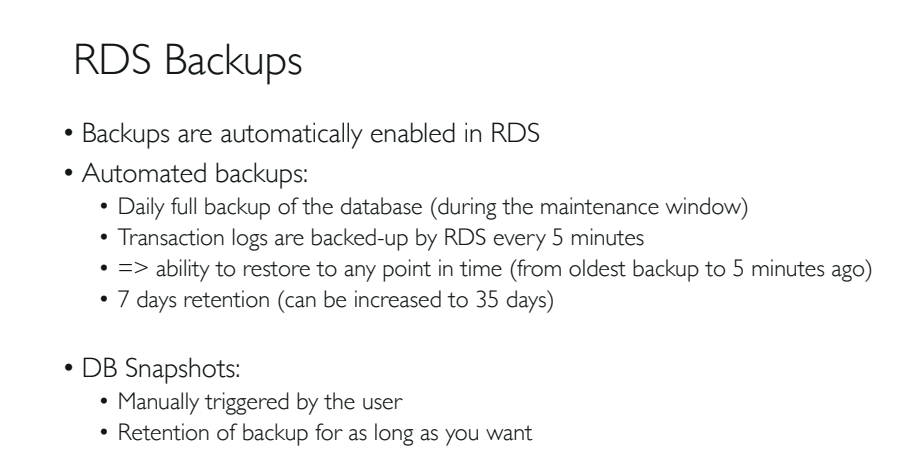
1. In case of rebooting.
2. We can reboot the RDS by selecting the option ‘Reboot with Failover’ on the Primary RDS DB instances.
3. Use Case: Whenever you change the DB settings, then there will be a good chance to reboot the RDS.

**AWS RDS Backup and Retention Period:**

1. Whenever failover occurs, AWS RDS sends SNS notifications. This notification service is chargeable.
2. If you are using services like API calls or CLI to find out the RDS events occurred in the last 14days.
3. But using AWS console, you can find out only last one day or last 24hour events.
4. In case of :
   1. OS patching
   2. System upgrades
   3. DB scaling(horizontal or vertical)

The above said changes will be performed on standby first , than on primary RDS to avoid outage.

1. Similar way in Multi-AZ snapshots and automated backups are done on standby instance to avoid I/O suspension (inaccessibility) on Primary RDS.



**RDS Backups Explained in Detail:**

1. There are 2 ways to take backup and restore in AWS RDS DB instances.
   1. AWS RDS Automated Backup.
   2. User Initiated Manual Backup.
2. Either you can take backup like snapshot of entire DB instance or just the DB.
3. You can create storage volume snapshots of your entire DB instances.
4. Automated backed up data will be stored in different AZ for data durability or high availability.
5. All RDS backups will be stored in AWS S3. Whereas RDS default storage used is EBS.
6. In Multi-AZ environment, the backups will be taken from the standby instance.
7. In case of single AZ environment, the backup will be taken on the same RDS instance and till the backup process ends the RDS services will freeze.
8. The DB instance should be in ‘active’ state for Automated Backups. But can take manual backup but the process is different.
9. RDS automatically backs up the DB instances daily, by creating a storage volume snapshot of your DB instance (full daily snapshot) including DB transaction logs.
10. AWS does not charge for RDS backup process, but AWS will charge for the storage (S3) used for the backup.

**Extra points on Backup:**

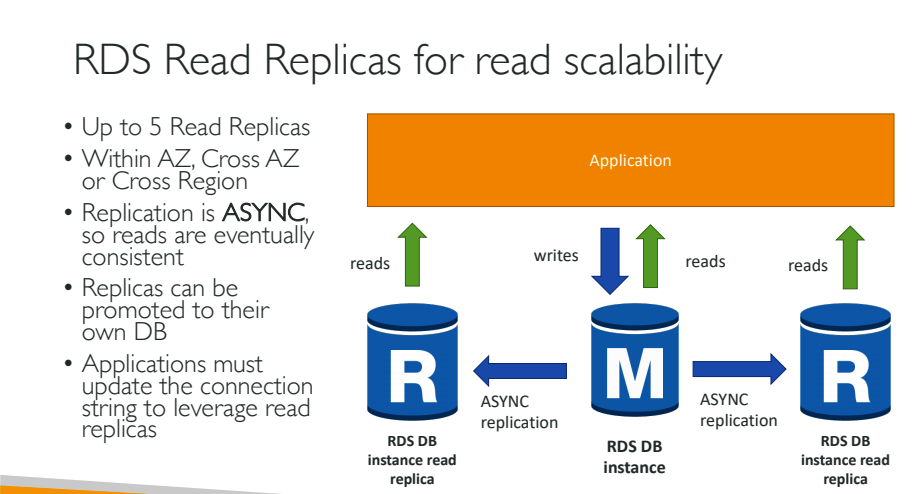
1. For Multi-AZ RDS, backups are taken from the standby DB instance for the following DB instances only – Maria DB, MySQL, Oracle, PostgreSQL.
2. Automated backup of RDS will be deleted when you delete your RDS DB instance by default. If we want we can change that default setting.
3. Retention period: Deletion of the backed up data.
   1. Range: 0 – 35days
   2. Default: 7days in AWS Console
   3. No backup: if set to zero
   4. Outage: if you change the previous retention period, say, changed from 5 to 10 days.
   5. In AWS CLI or API: Default retention period for every RDS is 1 day only.

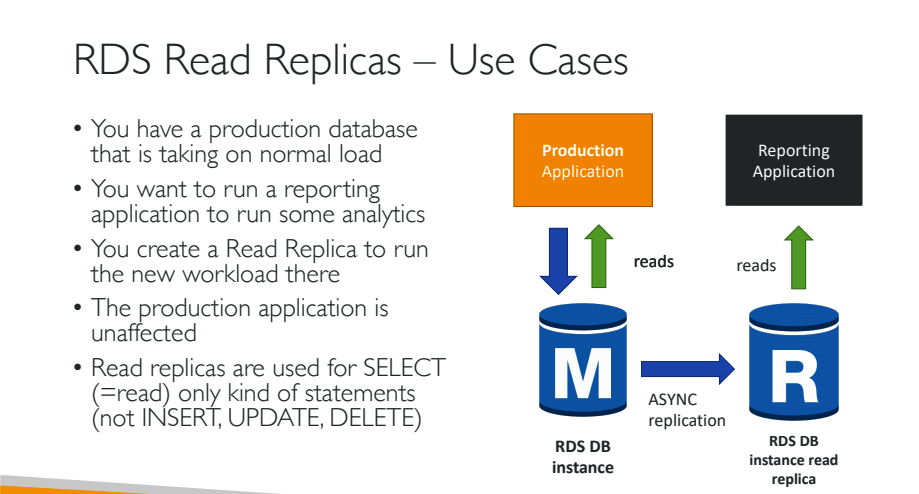
**NOTE**: In case of AWS Aurora, the default retention period is 1 day.

**NOTE**: you can increase retention period up to 35 days.

**RDS Multi-AZ Deployment – Maintenance:**

1. Firstly, perform maintenance on standby.
2. Than convert standby to primary, so that maintenance can be done on primary.
3. Steps to follow in console to modify the version of RDS:
   1. Go to console, Select RDS
   2. DB instance
   3. Modify DB
   4. Set DB engine version according to your requirement.
4. In Multi-AZ, version upgrade will be conducted on both primary and standby at the same time, which will cause an outage.
5. So, it is always recommended to do the version upgrade during ‘**maintenance window period**’. Mostly after 12am. But you can force do it if you want.



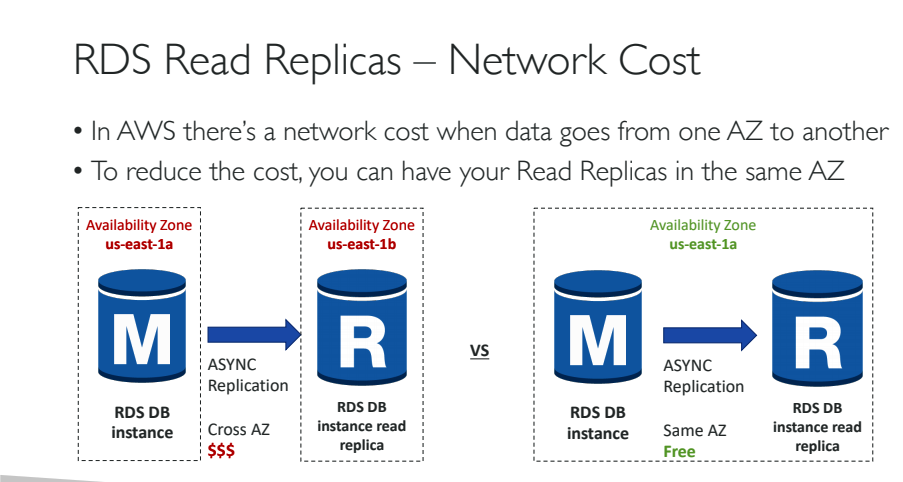


**Points to remember while AWS RDS Manual Snapshot and Recovery:**

1. In case of Manual snapshot, point-in-time recovery is not possible.
2. Whereas in case of Automatic backup, we can take backup up to 5min before data.
3. For example if you selected for an automated backup process at 4.30pm. You will get data backup till 4.25pm. The 5 min difference is called RPO [Recovery Point Objective].
4. Where as in manual backup the RPO can be 5min or 10min or 15min. Can’t say actually.
5. RTO: Recovery Time Objective means data backup process completion time from start to end.
6. Manual or Automated backup, the storage is in S3.
7. In case of automated backup, if you delete the RDS instance the backup will also get deleted automatically.
8. But, in case of manual backup the data from S3 will not be deleted automatically. It will remain. Billing will be continued.
9. It is always recommended to take a final snapshot of RDS instance before deleting it.
10. You can share Manual snapshot directly within other AWS account. Where as in automated you have to take a manual snapshot to share.
11. When you restore a DB instance using a snapshot, it means we are creating a new instance using the existing RDS snapshot. So, only default DB parameters and security groups are associated with the restored instance will be available in the new instance. Custom changes and parameters will not to available.
12. Cannot restore a RDS snapshot on an existing instance (imp).
13. At the time of restoring an RDS snapshot into a new RDS instance, you can select the storage type(Ex: Provisioned IOPS or General Purpose).

**RDS Encryption:**

1. You cannot encrypt an existing un-encrypted DB instance.
2. To encrypt and existing RDS instance, first create new encrypted instance and migrate your data to the newly created instance.
3. (or) You can restore from a backup/snapshot into a new encrypted RDS instance.
4. RDS supports ‘**Encryption-at-Rest’** for all DB engines using KMS.
5. What actually is encrypted when data is at rest:
   1. All the snapshots of RDS.
   2. Backups of storage at S3.
   3. Data on EBS volume.
   4. Read Replicas created from the snapshot.



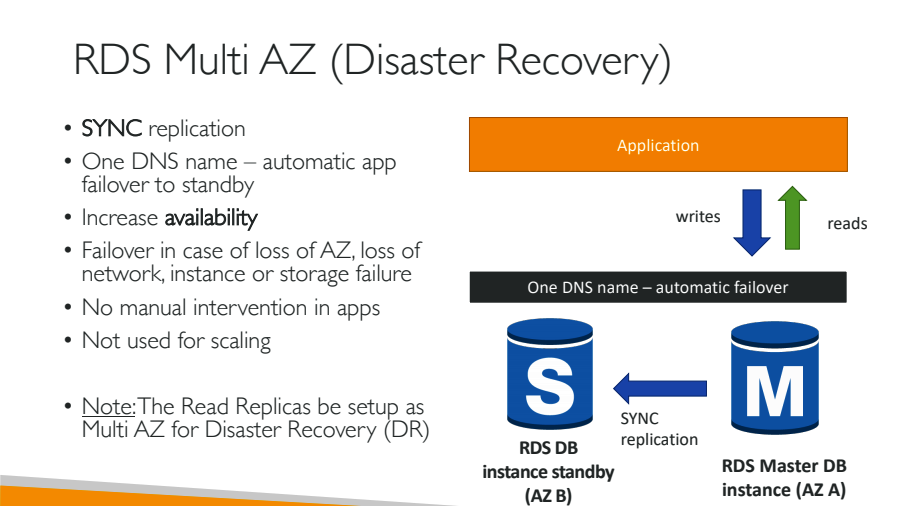
**RDS Billing:**

For standalone RDS instances:

1. No upfront cost except reserve instance.
2. you have to pay only for :
   1. DB instance type per hour (partial hour charged as full hour).
   2. Storage GB/month(EBS volumes)
   3. Data transfer to RDS using internet. Per GB(migrations or backup).
   4. If retention period increases then billing cycle increases.

Charges for Multi – AZ instances:

* 1. Multi-AZ DB charges per hours( Means have to pay for primary and standby instances)
  2. Double write I/O charges. Data written in Primary DB will also get written in Standby. So 100GB written in primary, 100 GB written in standby will cost you both.
  3. No read charges.

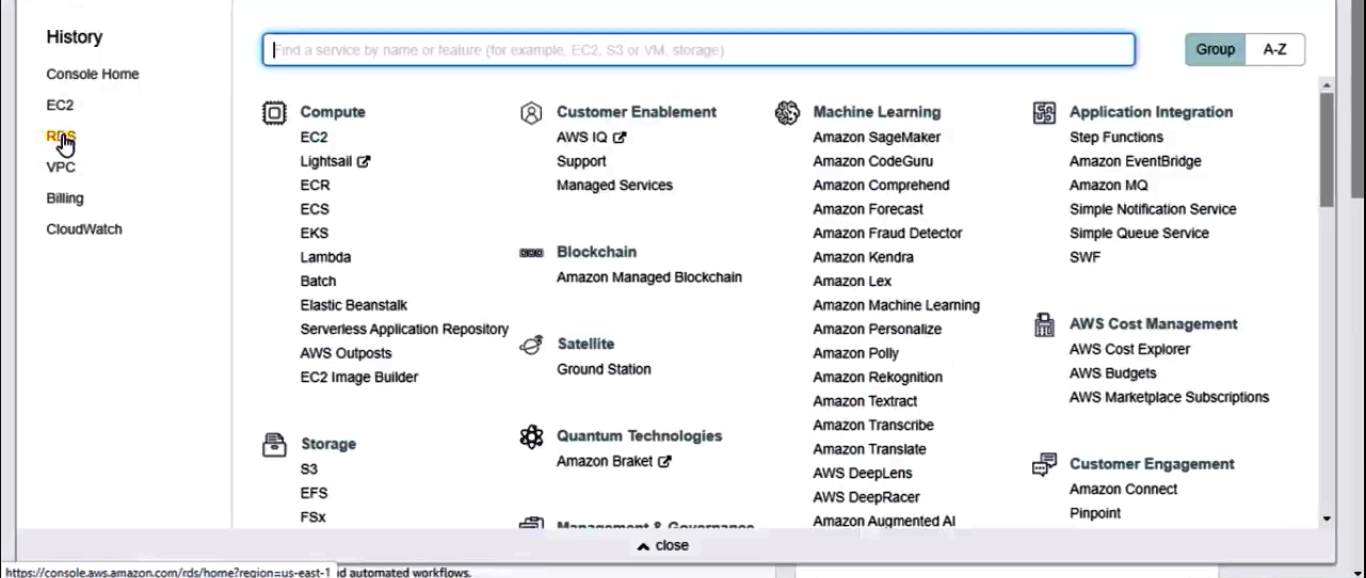


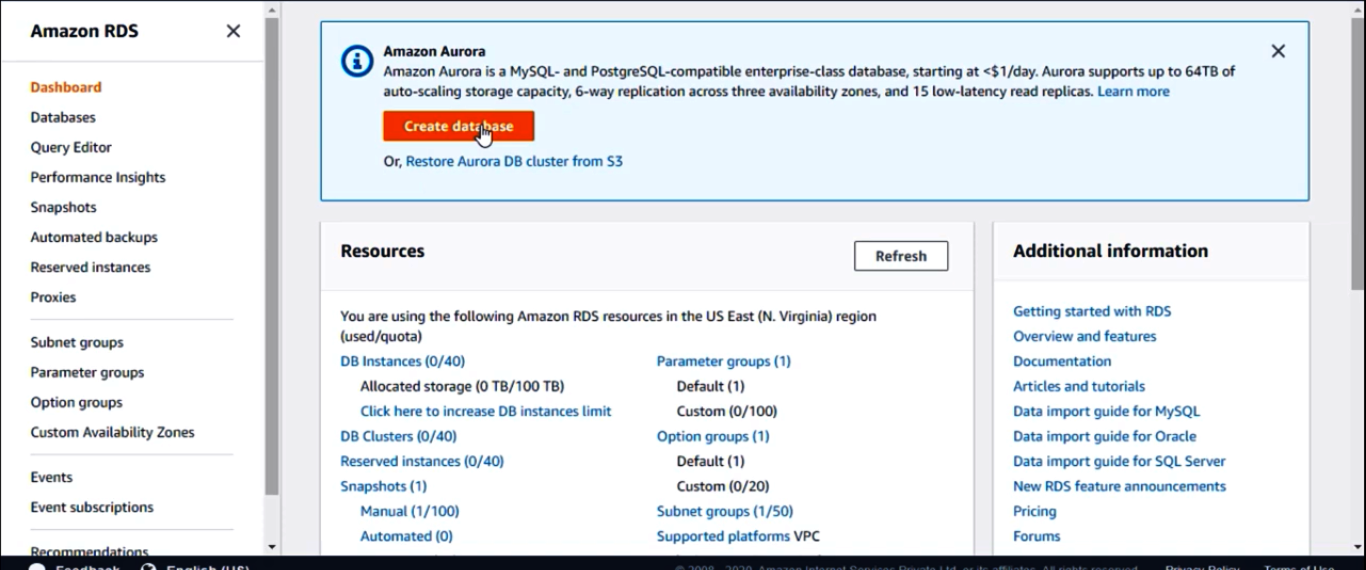
Hands On:

Task: How to access MySQL database from Linux Instance

Steps:

1. Create RDS DB instance.
2. Access this RDS instance is using Putty or MobaXterm.
3. Observe the billing cycle.
4. Delete RDS instance and Linux instance.





Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

For more info on creating and maintaining the Amazon RDS, follow the following documentation link:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\_MySQL.html