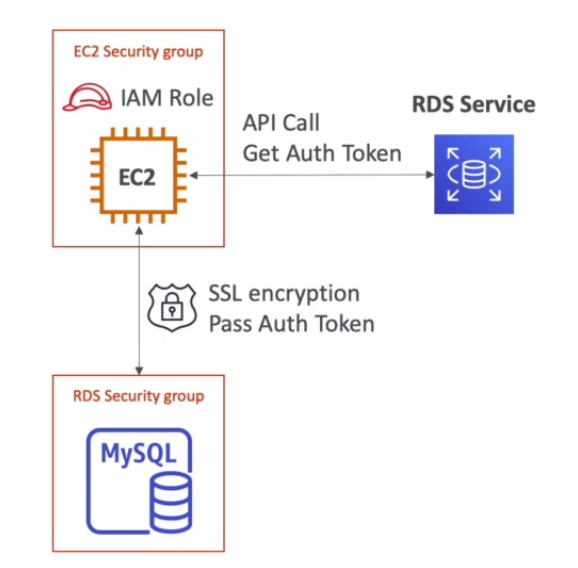
**AWS RDS**

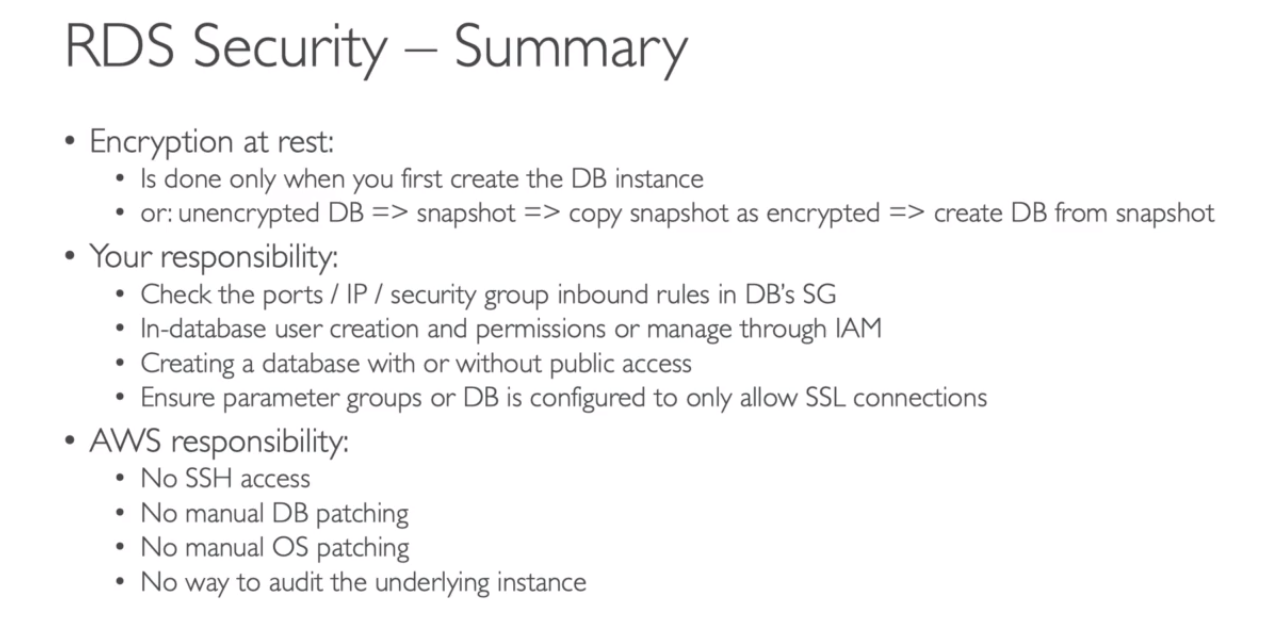
* RDS - relational database service
* Database serviee that uses sql
* DB’s managed by AWS in the cloud
  + Postgresql
  + Mysql
  + Oracle
  + Mariadb
  + Microsoft sql server
  + Aurora
* Why not deploy db on ec2?
  + Because RDS is a managed service
    - Automated services
    - Continuous backups
    - Monitoring dashboards
    - Disaster recovery
    - Scaling capabilities
    - Storage backed by EBS
  + DOnt get the background ec2 access
* **RDS BACKUPS**
  + Automated
  + DAily full backup
  + Transaction logs back up every 5 mins
  + Retention - 7days ( can be increased to 35 days)
  + **Snapshots-**  Manually triggered by the user, has retention for as long as you want
* **Storage Auto scaling**
  + Helps increase storage dynamically when running out of free space.
* HAve to set maximum storage threshold

| **RDS Read replicas** | **RDS Multi-AZ** |
| --- | --- |
| 5 read replicas- within AZ,cross AZ and cross region | For disaster recovery |
| REplication is async | Replication is syncronous |
| Replicas can be promoted to their own DB | Failover in casae of loss of az, ni manual intervention in apps |
| Only for read | Not used for scaling |
| Cost only for cross region |  |

**Read replicas can be setup as multi AZ**

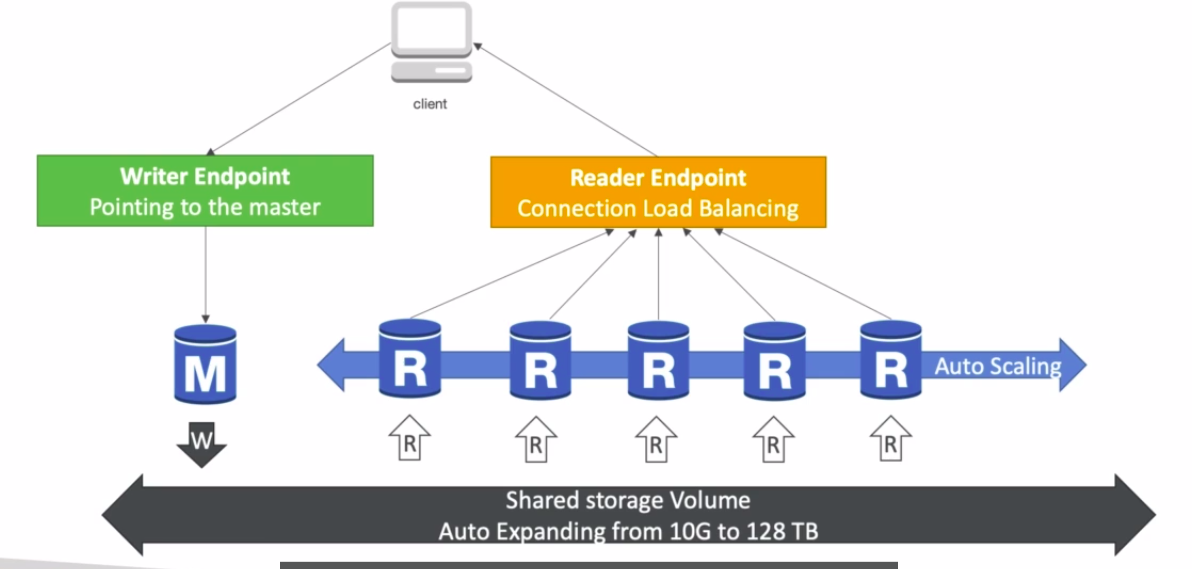
**RDS security**

* At rest
* In flight
* Encrypting rds backups
* To encrypt an unencrypted RDS
* IAM
* **At rest**
  + Can encrypt master and read replicas with AWS KMS
  + If master is not replicated, read replicas **cannot** be replicated
  + Transparent data encryption available for oracle and sql server
* **In flight encryption**
  + SSL
  + To enforce ssl- rds.force\_ssl = 1 or ‘Require SSL’
* **Encrypting RDS backups**
  + Snapshots of encrypted RDS are encrypted
  + SNapshots of unencrypted Rds are unencrypted
  + Can copy a snapshot into an encrypted one
* **TO encrypt and unencrypted**
  + Create a snapshot
  + Copy snapshot into an encrypted one
  + Restore DB from the new snapshot
  + Migrate applications to the new databases and delete the old DB
* **Network Security**
  + Usually deployed within a private subnet
* **Access management**
  + **IAM** Policies help control who can manage AWS RDS
  + Username and password can be used
  + IAM based auth for MYsql and PostgreSQL
    - Generate an auth token through IAM and api calls(15 mins lifetime)
    - 
    - Users are centrally managed



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Amazon Aurora**

* proprietary
* AWS cloud optimized, 5xMySQL, 3xPostgres
* Automatically grows in increments of 10GB upto 128 TB.
* CAn have 15 replicas
* Costs 20% more but is more efficient also
* **Read scaling**
  + 6 copies of your data across 3 AZ- 4 for writes and 3 for reads
  + Replication+Self healing + auto expanding
  + 1 master + 15 replicas
* There is a writer endpoint for master and there is a reader endpoint that client can connect to for url of read replicas
* 
* Shared storage volumes, Reader end point , writer endpoint imp
* Aurora security
  + Same as RDS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Amazon elasticache**

* **Is used to get managed REdis or Memcached**
* Helps reduce loads for read intensive workloads
* Makes your application stateless.
* Need to make heavy application code changes
* Application writes the session data into Elasticache
* **Redis vs Memcached**
* **Redis**
  + Multi AZ with auto failover
  + Read replication , has high availability
  + Persistent
  + Backup and restore failures
* **Memcached**
  + Multi-node for data partitioning of data(sharding)
  + No replication, not highly available
  + Non persistent
  + No backup and restore
  + Multi threaded architecture