5️⃣

Lecture 5 - Databases in AWS - RDS, Aurora, Elasticache (1h)

Q&A about the previous lesson (3-5m)

RDS

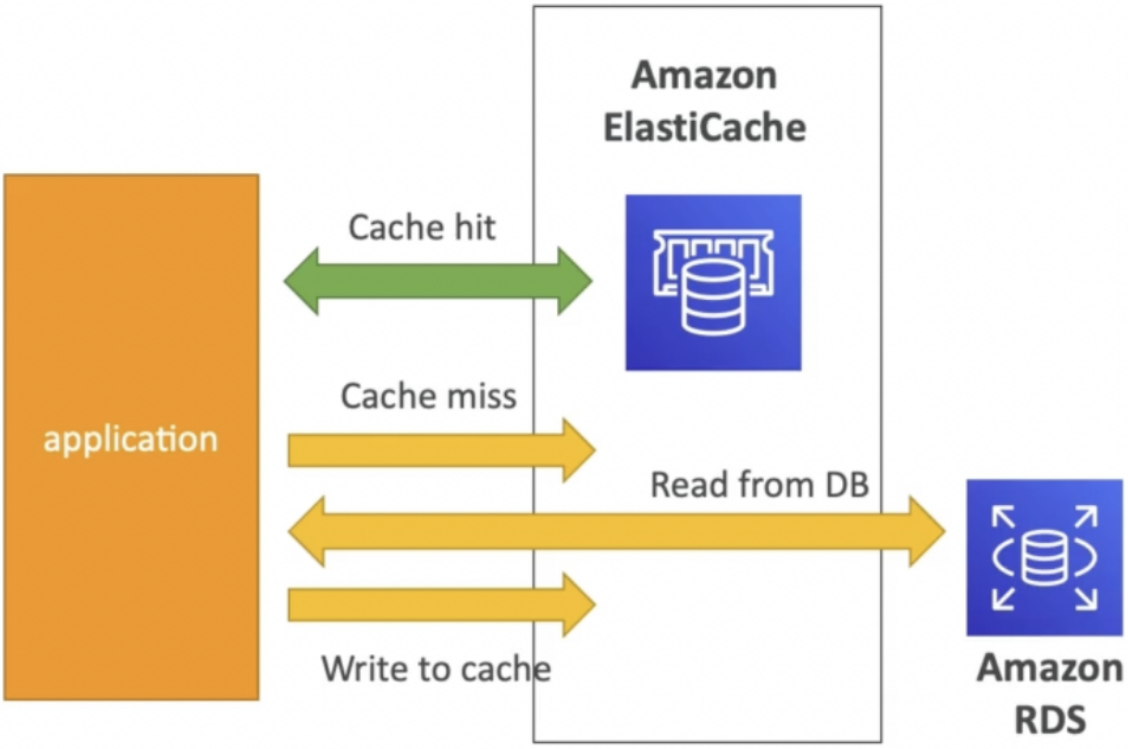
- Fully managed
  - no tuning required → slight customization with parameter groups https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html
  - no SSH access to the DB instances
  - OS updates and patches are done automatically during the maintenance window
  - automated backups - point in time restore
    - binlogs (every 5min)
    - full (daily)
    - or manual snapshots by user (+ final snapshot at DB instance deletion)
  - monitoring built-in
  - can be private and public → depending on the VPC and subnet settings
  - have security groups
  - multi-az
    - only for HA and failover - cannot interact with the standby instances - they're just sitting there until the failover
    - quite expensive → mutliple idle db instances + charge for mutli-az traffic
    - sync replication → STRONGLY consistent
    - DNS name isn't changed in case of failover - the app shouldn't see no difference
  - multi-region deployments
    - only for disaster recovery and local performance in that region
  - read-replicas https://aws.amazon.com/rds/features/read-replicas/#:~:text=Amazon RDS Read Replicas provide,for read-heavy database workloads.
    - offload read operations to the separate DB endpoint

        - up to 5 read replicas
        - can be promoted to the main db instance
        - async replication → EVENTUALLY consistent
    - auto-scaling (both vertical and horizontal)
        - storage auto-scaling
        - instance type change can be initiated by user
    - storage is backed by ebs
    - **encryption by KMS**
        - if master is not encrypted from the beginning - replicas are not encrypted
        - can be encrypted later with snapshots manipulation and restored from snapshot
    - integrates natively with the Secrets Manager → automated secrets rotation
    - IAM authentication is supported for some engines
- Supported engines
    - postgres
    - mysql
    - mariadb
    - mssql
    - oracle
    - aurora
        - mysql
        - postgres
        - *serverless*
- Snapshots → https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateSnapshot.html
    - used for data backup
    - cross-region replication
    - encypt-decrypt operations
    - incremental
- RDS proxy → https://aws.amazon.com/rds/proxy/
    - connection pooling
    - better performance
    - better scalability
    - also fully managed

- Amazon Aurora → https://aws.amazon.com/rds/aurora/features/
  Everything is the same as RDS, plus
  - features
    - more replicas (15 against 5), faster replication
    - x times better performance than open-source counterparts
    - almost instant faliover (30 seconds downtime)
    - 20% more expensive
    - better monitoring
    - backtrack - imporved point in time recovery
    - Overall more complex and worth learning separately
  - Aurora serverless → serverless configuration https://aws.amazon.com/rds/aurora/serverless/
  - Aurora global database → mutli-region deployment out of the box

Elasticache

- fully managed caching service
- engines
  - redis
  - memcached
- use cases
  - in-memory databases
  - cache for RDS
    - reduce the load
    - increase the performance
    - helps making apps stateless



- redis - supports multi-az and read-replicas
- the app should know how to work with it!
- For certification - know difference between redis and memcached architectural limitation
  - Redis - mutli-az with failover, read-replicas, backup and restore features (RDS)
  - memcached - multi-node for sharding, no HA, no backup and restore

Workshop

https://general-webapp.workshop.aws/lab2.html