

Database Services in AWS

- Potureddi Gowtham

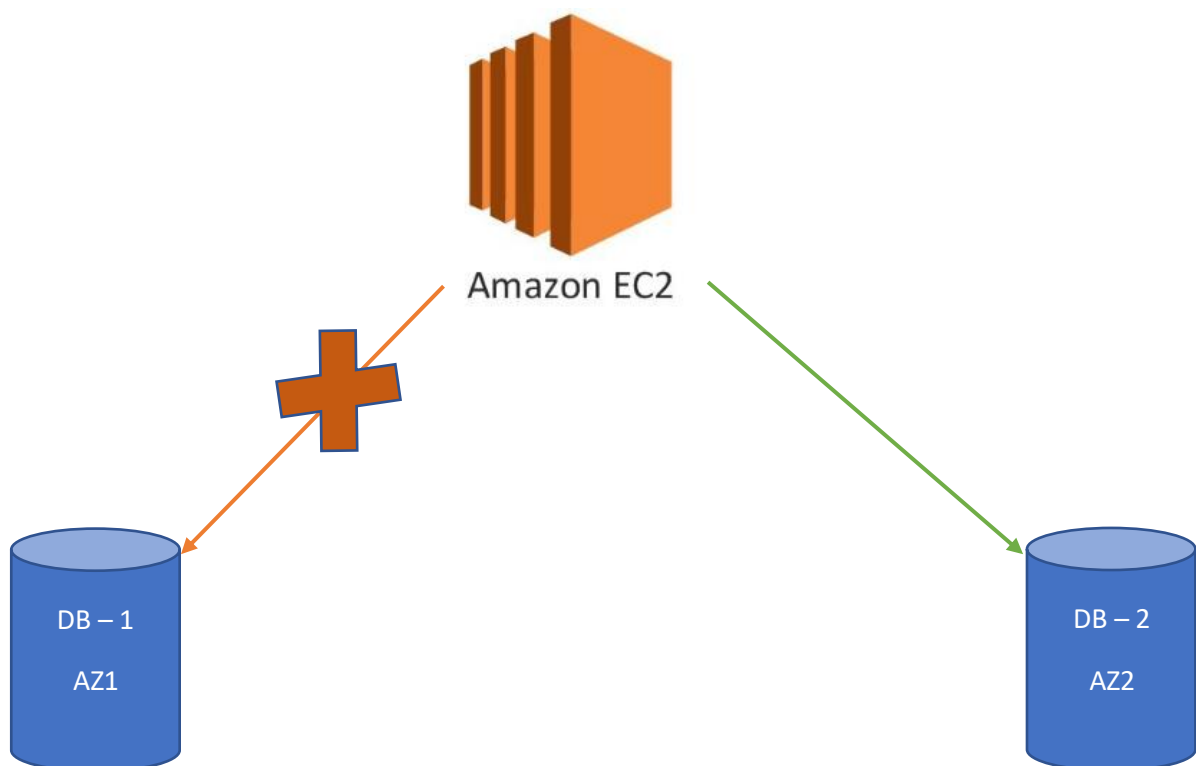
Relational Databases:

- These are just like excel or CSV files where we have so many rows and columns.
- A excel may have different work-sheets just like different tables in data base.
- Each table will have a relation with other table.
- Each column in a database table is called attribute.
- We will have a Primary Key in each table, to identify the information.
- Ex: registration number of a student, Surname etc. (which will be unique and will not have repetitions.)
- To build relationship between tables we have Foreign keys.

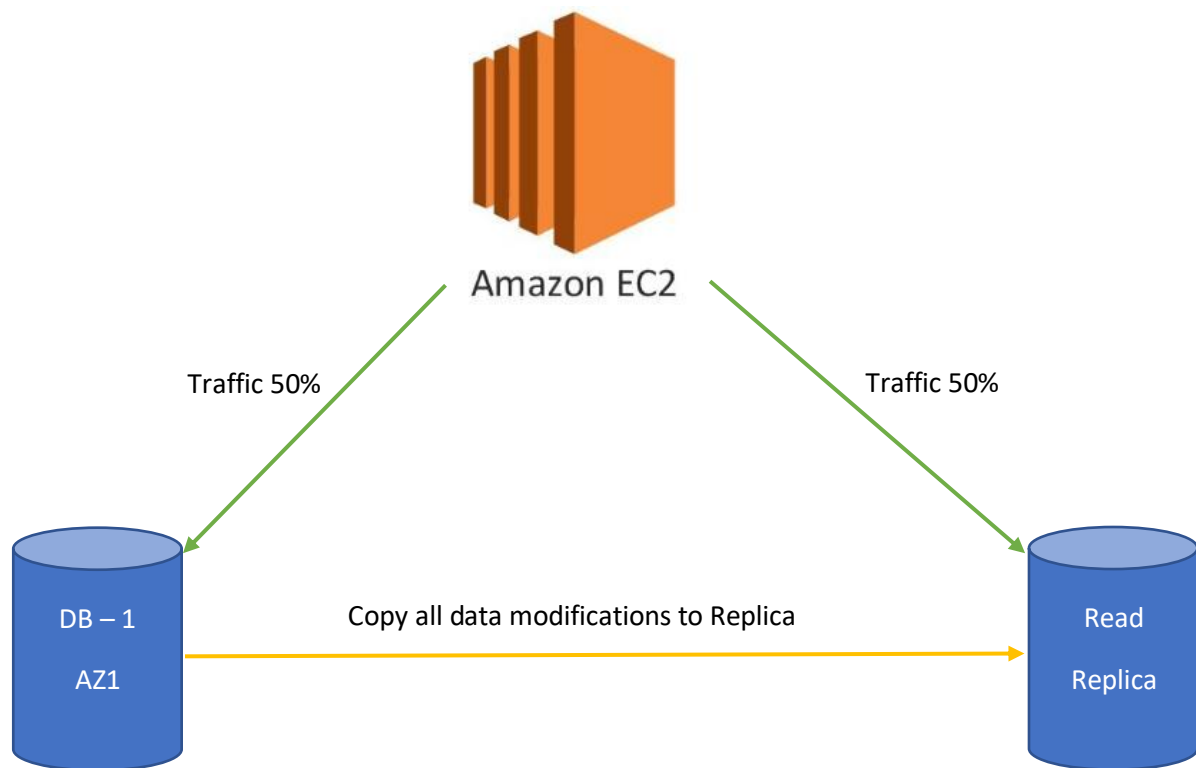
Relational Databases (RDS) in AWS:

Features of RDS:

- These Data-Base Instances can be running in more than one availability Zones.
- So, even if there is any problem with DB in one Zone, data can be taken from Another DB .
- All this will be done automatically and taken care by Amazon.



- We can also maintain Read Replicas for performance
- We first must maintain a replica from primary DB and then if you need to scale your DB, you can redirect some traffic to Read Replica.
- We can have many Replicas like this.
- Ex: if traffic is more scale DB using 5 replicas.



- RDS runs on a virtual machine, but you cannot SSH or login to these machines.

Backups with RDS:

Automated backups:

- This automated backup takes full daily snapshots of the DB.
- It will also take and store transactional logs throughout the day.
- If you try to do recovery, it will give the very latest snapshot.
- We can also get the needed snapshot according to our requirement.
- All these backups are stored in S3.

- Again, we need not pay for S3, we will be provided S3 free for the memory size of our instance.
- If our RDS is 25GB, we get S3 bucket of size 25GB.
- When the backups are going on we can expect a bit of latency.

Database Snapshots:

- These are done manually by us.
- Just to store the entire state of the DB.
- We can use these even after deleting the RDS instance.

Note:

- When we restore instance from both Automatic backups or Database snapshot, the result is a new RDS instance with new endpoint.

DB's we can use:

- SQL SERVER
- ORACLE
- MYSQL
- PostgreSQL
- Amazon Aurora
 - Made by amazon
- MariaDB

NoSQL Data-Bases in AWS:

- These are not in the form of tables, rows and columns.
- They are in the form of JSON.
- These are like collections just like Tables
- Documents just like rows Row
- Key value pair just like Fields

```

{
  name: Joe,
  url: '...',
  stream:
  [{
    user: {name: Jane, url: '...'},
    title: 'today',
    body: 'go fly a kite',
    likes: [
      {user: {name: Lu, url: '...'}},
      {user: {name: Joe, url: '...'}}
    ],
  }]
}

```

AWS DynamoDB:

- This is a NoSQL DB made by Amazon.
- This supports document and key-value data models.
- This DB can be used for many different use-cases like:
 - Mobile apps
 - Web apps
 - Game development
 - IOT apps.
- Data is stored in an SSD.

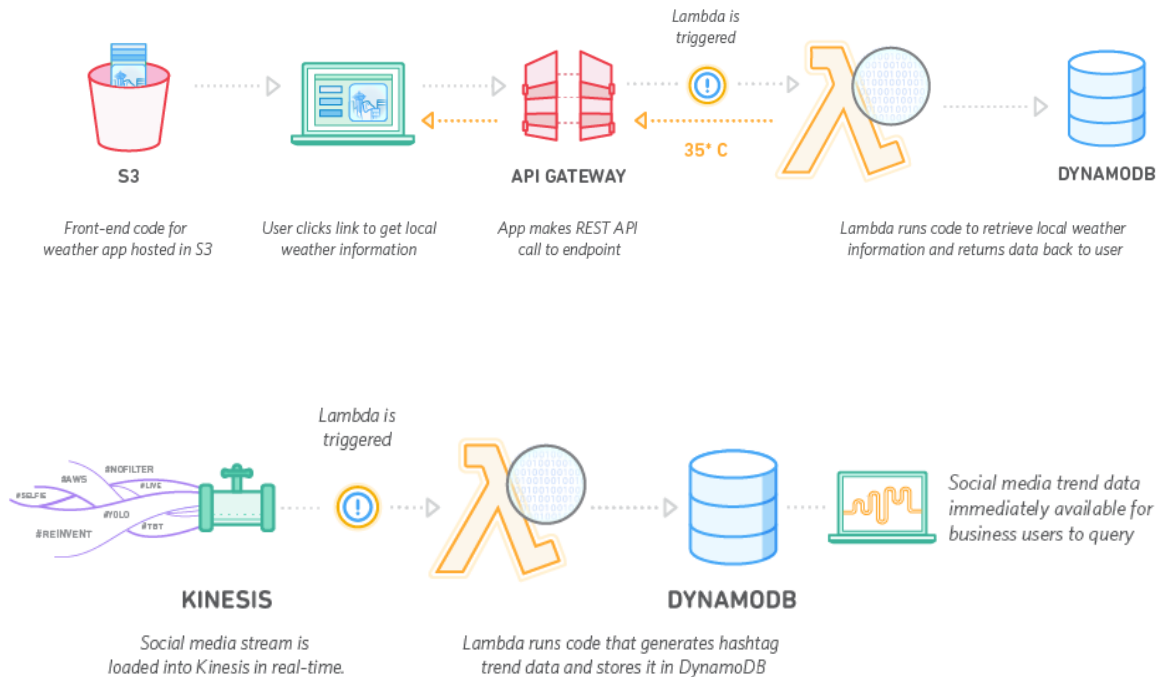
There are 2 different reads:

Eventual Consistent Reads:

- If we make a write in DB, we can read that after a second or two.

Strongly Consistent Reads:

- Unlike Eventual Consistent read we can read the data in less than a second.
- Reads are very fast.
- So, based on the use-case we can select the Eventual or Strong read types.



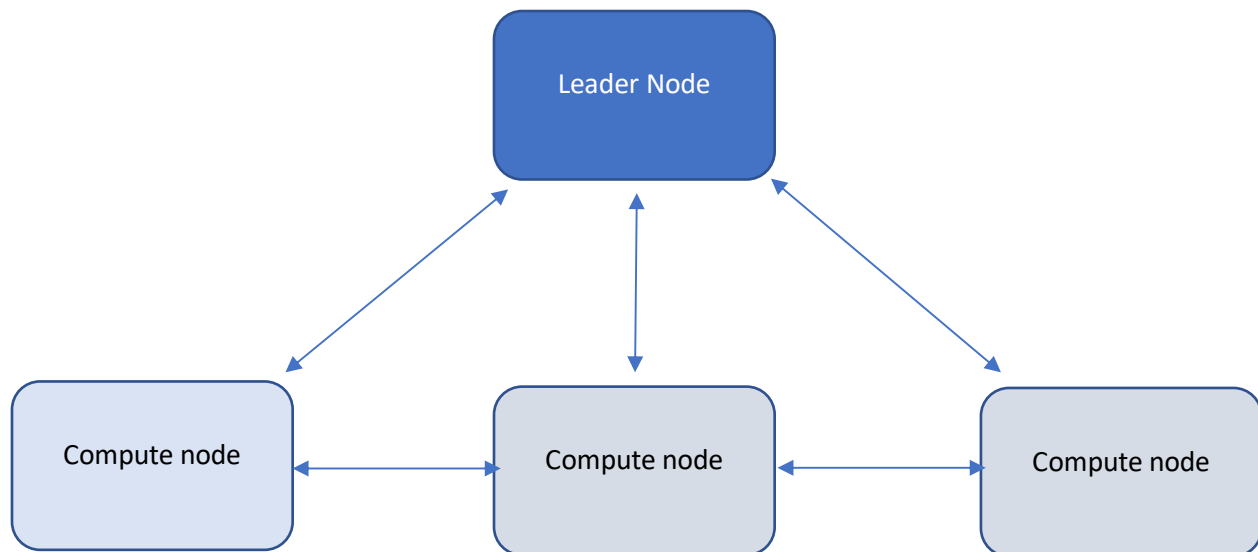
Data Warehousing:

- This is used for Business intelligence.
- We can collect data from various sources and use it for providing various business insights.

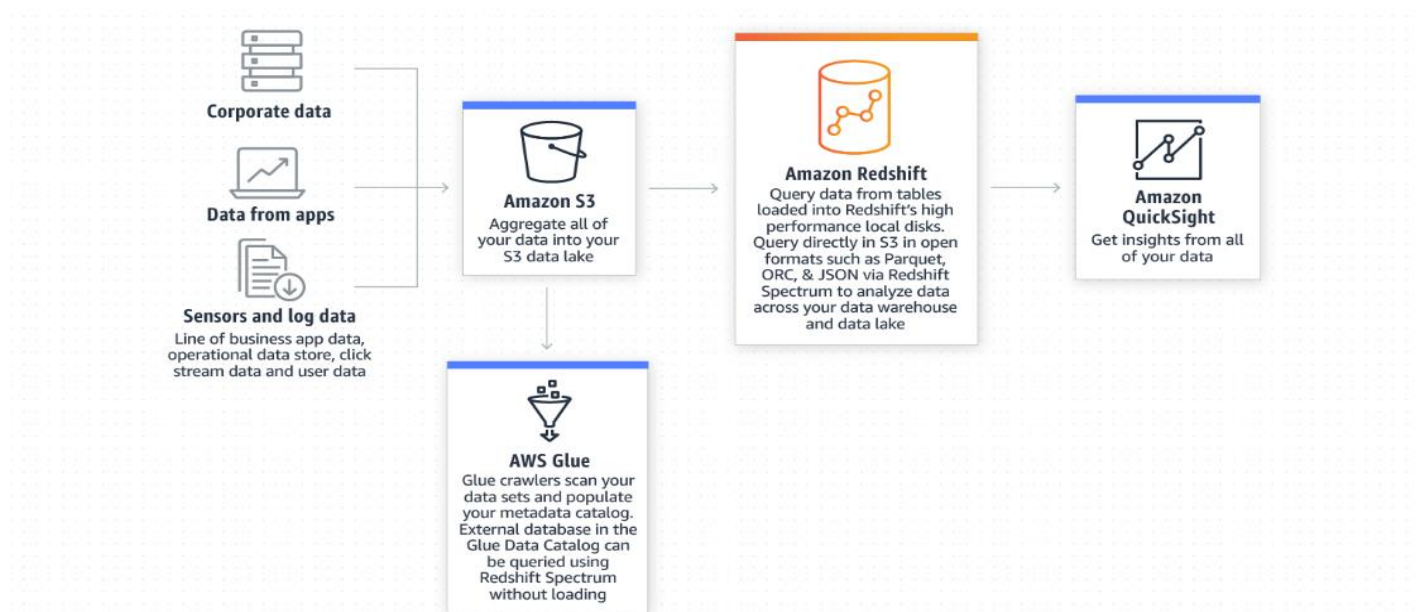
AWS RedShift:

- This is for data warehouse services in cloud.
- This can manage petabytes of data.
- This can be used for OLAP (Online analytical processing) to acquire insights for your business and customer needed.
- To perform OLAP, we need to apply multiple quires and perform analytical operations over it.
- Redshift can efficiently handle all these irrespective of the dataset size.
- We can build a cluster and set needed number of nodes.
- After a cluster is built we can upload data and then perform data analysis queries.
- We can use advanced Compression over the data.
- So, we can reduce memory usage effectively by applying these compression techniques.

- Single node with 160GB size.
- Multi node
 - Leader node (To manage all the slave nodes.)
 - Compute nodes (To store data and perform queries.)
 - We can have up to 128 compute nodes.



- We can apply massive parallel processing using multiple nodes.
- Redshift automatically distributes data and query load within the nodes.
- So, with the increase in warehouse we can increase number of nodes.
- We can take multiple backups and they will be stored in S3 in another region for disaster recovery.

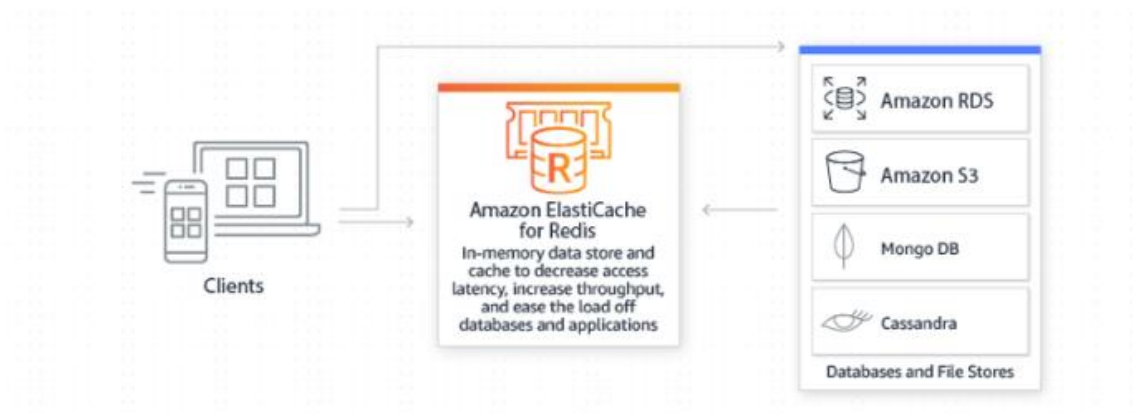


AWS Aurora:

- This is a relational database made by amazon.
- This DB can provide 5 times better performance than MySQL.
- Memory plan starts from 10GB and can extended to 64TB
- Compute resources can also be scaled up to 32vCPUs and 244GB of Memory.
- We can have 2 copies in each availability zones, with minimum of 3 AZs.
- So, totally 6 copies are maintained.
- Aurora storage is self-healing.
- Data blocks and disks are continuously scanned for errors and repaired automatically.
- We can have up to 15 read replicas and 5 read replicas to MySQL DB.
- We can take backups and snapshots for these DB.
- And interestingly there won't be any latency or impact when we are doing backups or snapshots.
- We can also share snapshots with other AWS accounts.

AWS ElastiCache:

- This is a webservice that makes us deploy, operate and scale an in-memory cache in the cloud.
- This can improve performance with fast retrievals.
- When there is a query getting repeatedly triggered its cached and used next.
- Ex:
 - If there is an online shopping portal and a product got some very good discount.
 - Users are repeatedly searching about that product.
 - So instead of getting the info always from the DB, we can cache and use it.
 - Means, store that data somewhere and use it when triggered again.

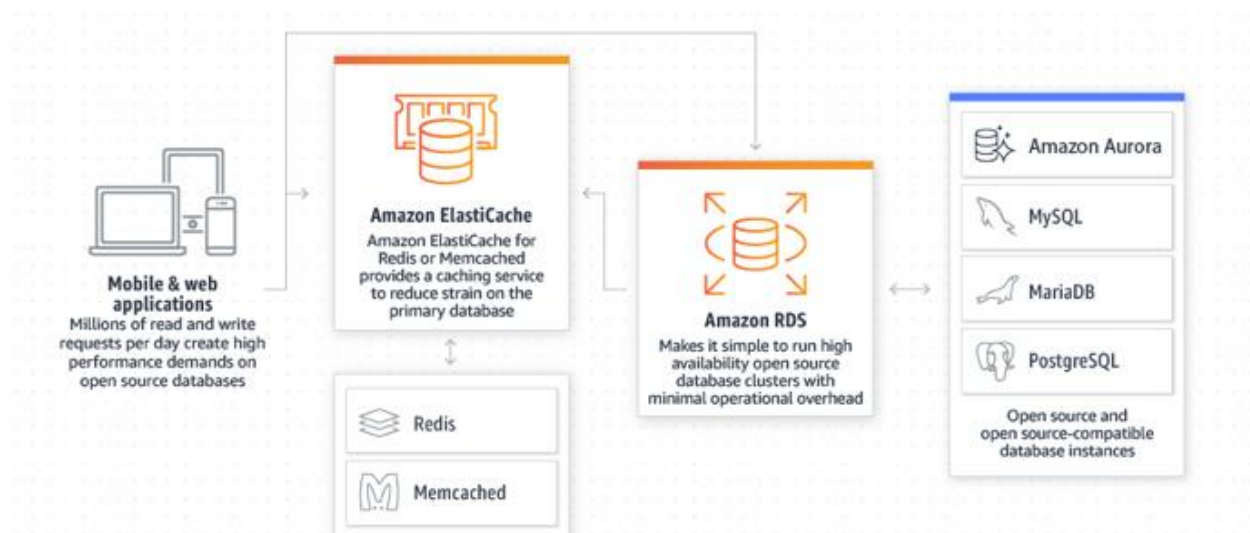


This has 2 types:

- Memcached
- Redis

Requirement	Memcached	Redis
Simple Cache to offload DB	Yes	Yes
Ability to scale horizontally	Yes	No
Multi-threaded performance	Yes	No
Advanced data types	No	Yes
Ranking/Sorting data sets	No	Yes
Pub/Sub capabilities	No	Yes
Persistence	No	Yes
Multi-AZ	No	Yes
Backup & Restore Capabilities	No	Yes

Transactional Database



Gaming Application

