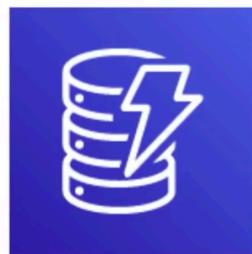


DynamoDB



A key-value and document database (**NoSQL**) which can guarantees **consistent reads and writes at any scale**.



Introduction to DynamoDB

What is NoSQL

NoSQL is database which is neither relational and does not use SQL to query the data for results

What is a Key/Value Store?

A form of data storage which has a key which references a value and nothing more

```
{ Title: 'S01E019 DS9 Duet' }
```

What is a Document Store?

A form of data storage which a nested data structure

```
{
  Series: 'DS9'
  Episodes: [
    {
      Season: 1,
      Episode: 19,
      Title: 'Duet'
    }
  ]
}
```



Introduction to DynamoDB

DynamoDB is a NoSQL **key/value** and **document** database for internet-scale applications.

Features

- Fully managed
- Multiregion
- Multimaster
- Durable database
- Built-in security
- Backup and restore
- In-memory caching

Specify your read and write capacity per second, it just works at **whatever capacity you need** without you tweaking anything.

Provisioned capacity

| | Read capacity units | Write capacity units |
|-------|---------------------|----------------------|
| Table | 100 | 100 |

Estimated cost \$58.04 / month ([Capacity calculator](#))

Provides

- Eventual Consistent Reads (default)
- Strongly Consistent Reads



All data is stored on **SSD storage** and is spread across **3 different regions**.



DynamoDB – Table Structure

| Primary Key | | | |
|---------------|----------|--|------------|
| Partition Key | Sort Key | | |
| IMDB ID | Year | Title | Box Office |
| tt0079945 | 1979 | Star Trek: The Motion Picture | 139000000 |
| tt0084726 | 1982 | Star Trek II: The Wrath of Khan | 97000000 |
| tt0088170 | 1984 | Star Trek III: The Search for Spock | 87000000 |
| tt0092007 | 1986 | Star Trek IV: The Voyage Home | 133000000 |
| tt0098382 | 1989 | Star Trek V: The Final Frontier | 63000000 |
| tt0102975 | 1991 | Star Trek VI: The Undiscovered Country | 96900000 |
| tt0111280 | 1994 | Star Trek Generations | 118000000 |
| tt0117731 | 1996 | Star Trek: First Contact | 146000000 |
| tt0120844 | 1998 | Star Trek: Insurrection | 117800000 |
| tt0253754 | 2002 | Star Trek: Nemesis | 67300000 |



DynamoDB – Reads

When data needs to be updated it has to write updates to all copies. **It is possible for data to be inconsistent** if you are reading from a copy which has yet to be updated. You have the ability to choose the read consistency in DynamoDB to meet your needs.



Eventual Consistent Reads (DEFAULT)

When copies are being updated it is possible for you to read and be returned an inconsistent copy

Reads are fast but there is no guarantee of consistent

All copies of data eventually become generally consistent within a second.

Strongly Consistent Reads

When copies are being updated and you attempt to read, it will not return a result until all copies are consistent.

You have a guarantee of consistency but the trade off is higher latency (slower reads).

All copies of data will be consistent within a second



DynamoDB *CheatSheet*

- **DynamoDB** is a fully managed **NoSQL** key/value and document database.
- Applications that contain large amounts of data but require predictable read and write performance while scaling is a good fit for DynamoDB
- DynamoDB scales with whatever **read and write capacity you specific** per second.
- DynamoDB can be set to have **Eventually Consistent Reads (default)** and **Strongly Consistent Reads**
- **Eventually consistent reads** data is returned immediately but data can be inconsistent. Copies of data will be generally consistent in 1 second.
- **Strongly Consistent Reads** will wait until data is consistent. Data will never be inconsistent but latency will be higher. Copies of data will be consistent with a guarantee of 1 second.
- DynamoDB stores 3 copies of data on SSD drives across 3 regions.