

NoSQL and Other Database Solutions in AWS

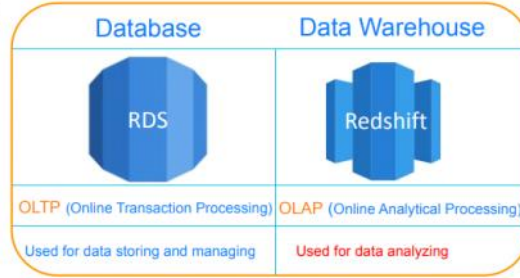


2 Amazon Redshift

Machine learning isimlerinde çok fazla aktif kullanıldığı için son zamanlardaki kullanım oldukça arttı. Çok büyük miktarda bilgiyi alıp, genel haliyle makineye öğretmeye çalıştığımız şey machine learning olarak geçiyor. Yapay zeka gibi durumlardan dolayı redshift gibi büyük depolama yerlerine ihtiyaç var. Redshift ilk çıktığı zaman rds gibi bir databasedi. Sonra serverless diye bir kavram aldı basını ilerledi. Bunun üzerine amazon hemen Aurora'ya serverless olarak sundu. Serverless dediğimiz kavram, server işleriyle bizim ilgilenmediğimiz, amazonun bizim verdiğimiz bilgilerle arkada işlediği olay. Serverlessin popülaritesinin artması üzerine redshiftin de serverlessini çıkarttılar.

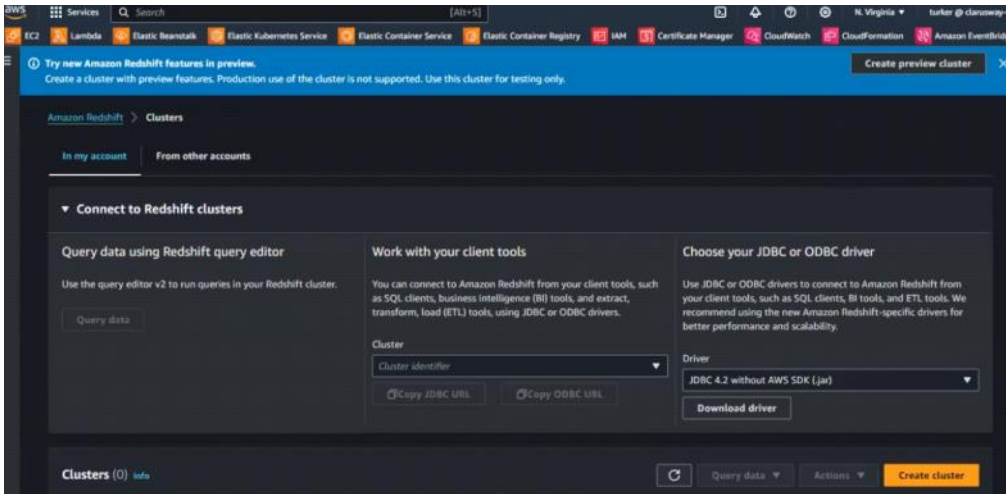
Rds te bizim seçtiğimiz saatler içerisinde bir backüplama yapıyordu. Onun devamında bir bakım penceresi açıyordu. Ama mariadb de mesela ortama yüklediğimiz durumda bunlar ortadan kalkıyordu. Biz bir ssh bağlantısı kurup database bağlanmak istediğimizde rds buna izin vermiyordu. Çünkü sadece database'e özel bir instance. O yüzden bir ssh bağlantısı vs yok. Ama bize verdiği endpoint üzerinden bu işlemleri yapabiliyorduk. Bağlanabiliyorduk yani. Redshiftte bu şekilde çalışıyor ama daha çok cluster yapısında. Cluster kumelenmiş demek. Bu cluster yapıda bir tane bu kumeyi yönetecek ana bir server oluyor master node adında. Ve altında da diğer işlemleri yapacak diğer nodelar bulunuyor. Bu yapıya bize cluster diyoruz, kumeler. BU YAPI BİZİM DEVOPS TARAFINDA DA KARSIMIZA ÇIKACAK. Master node bir tane olur ama worker node çok olur bu sistemde. Redshiftte bu cluster yapısıyla çalışıyor. Rds te ssh ile bağlanamadığımız gibi burada da bağlanamıyoruz. Ama burada bakım vs aws te. Burada bağlantı yine endpoint üzerinden. Workbenchte biz rds in içini gördük ve basit bir şekilde yönettik.

► Amazon Redshift



- Since the **analyzing process** causes an extra workload on database we prefer to use data warehouse
- Amazon Redshift is a fully managed, cloud-based, petabyte-scale **data warehouse** service by Amazon Web Services (AWS).
- Amazon Redshift is an efficient solution to **collect and store** all your data to **analyze**.

OLTP transaction, OLAP ise analitik Transaction gunluk hizli islemler icin kullanilir. Mesela logg in yapcaklar icin hemen userdatadan passwordu ismini vs alsin. Sonra iki password kullanıcı adı uyuypmu birbirine bunu kontrol etsin. Bunlar gunluk basimiza gelebilecek seyler. Gunluk hemen elde edilir. Ama analitik dedigimiz sey daha buyuk noktalarda kullanilir. Butun veriler toplanir buralarda ve analitik bir veri elde edilir. Mesela bir urunun bir yil boyunca en cok hangi yas araligi kadinlar tercih etmis vs gibi. Bu zaman alir. Redshift bir data warehouse servistir. Bu sebepten OLAP kullanir. Fully manged bir servis. Yani aws tarafından yönetiliyor. Cloud tabanlı. Yuksek olceklenebilir. Rdsse gore daha olceklenebilir. Yuksek miktarda elinde tutabilen depolama servisleridir. Bir suru farkli servisle birlikte calisabilir. Mesela quicksize servisi birlikte kullanip gorsel bir sov yapilabilir. Ihtiyac olan islem dahilinde database mi data warehouse mu tercih edilecegine karar verilir.

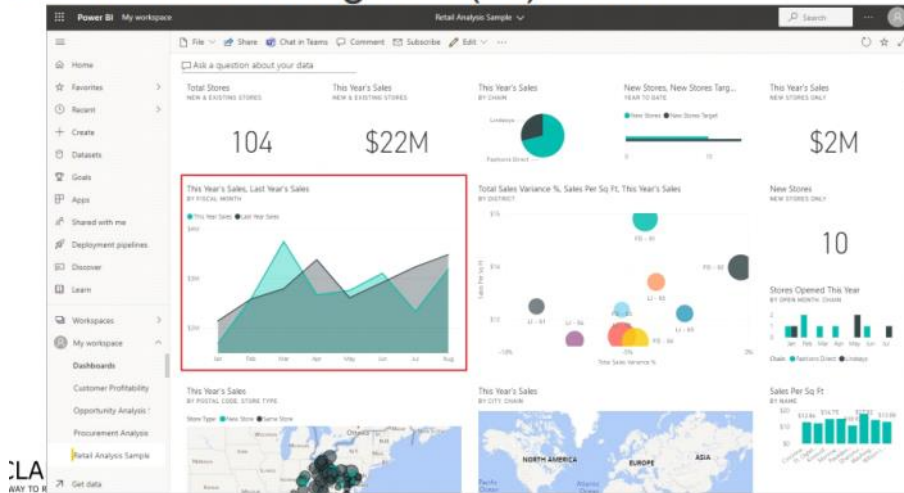


JDBS bu java database connectivity
ODBC open source connectivity
Bunlara gore baglanti yonlerini ayarlamamiz gerekiyor
Sql noskle gore daha hizliydi
ETL ise bizim yiginla data icinde almak istedigimiz veriyi extract edip (cekir), bu bilgiyi kendi kullanacagimiz sekilde transform edip(donusturup), istedigimiz yere load etme(yukleme) isine denir. (Extract Transform Load) BU KAVRAMI BILMEK ONEMLI. Amazonun ETL icin kullandigi servis ise ETL GLU.

► Amazon Redshift

- Amazon Redshift is a fully managed, **petabyte-scale data warehouse** service in cloud. It's a **very large relational database** traditionally used in big data applications.
- **Incredibly** big - it can hold up **16 PB** of data. This means you **don't have to split** up your large dataset.
- This database is **relational**. You use your **standard SQL** and business intelligence(BI) tools to interact with it.
- While Redshift is fantastic tool for BI applications, It's **not a replacement** for standard RDS databases.

► Business intelligence(BI) tools



► Exam Tips

- Amazon Redshift is **not a high available** service, meaning it only comes online **at one AZ**.
- If you want it in **multiple AZs**, you're gonna have to create **multiple copies**.
- Keep in mind it's for **BI applications**, it's **relational**, and it can store up to **16 PB** of data

3 ► AWS ElastiCache

Bize sunulan bir onbellek hizmeti yani cacheleme. Bu veritabanı olabilir, webserverlar olabilir(database ile webserver arasında bilgi alisveri olurken , bilgileri onbellege atip aradaki veri ulasimini hizlandiriyor. Webserver her seferinde databaseden bilgi almiyor hizli olmasi icin elasticacheten bilgi aliyor.), hdd diyebilir. Ama hdd le simdi artik kullanilmayacak kadar yavas. Onun yerine olmazsa olmaz sdd ler oldu. Ama her ikisinde ram hizina asla erismezler. Ram dedigimiz sey cachelemeyi yani hafizada tutup bilgiyi islemeyi hizli bir sekilde yapar. Sdd veya hdd dedigimiz olay rame gore daha yavas ama oraya atilan bilgiler kalici. Bilgisayarın acilip kapanmasıyla degisen bir bilgi olmuyor. ElastiCache dedigimiz sey ram takmis oluyoruz ve uygulamamızın islem hizi artiyor. Uygulama ve database surekli kontakt halindeler. Bu esnada yardimci olarak elasticache yapinca o komunikasyina destek atmıs oluyor. Boylece islem daha hizli olmus oluyor. Database in uzerindeki yuk azalmis oluyor ve performans artmis oluyor.

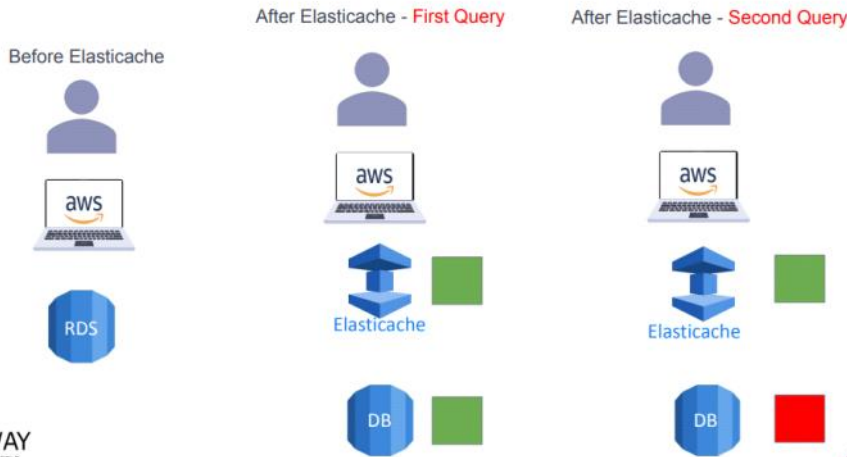
► AWS ElastiCache



Kullanıldığı iki engine var: Redis ve Memcached. Bunlar sadece amazonda değil diğer yerlerde de var. Amazon en çok kullanımları kendi bünyesi altına aldı parasıyla ve servis vermeye devam ediyor. İkisi de cachelemek için kullanılan servis ama farklı şekillerde kullanılabilir. Ortak ve farklı özellikleri var. Memcached daha çok multi işlerde performansı yüksekken, redis daha çok tekil işlemlerde. Redisin cacheleme hafızası memcachede göre daha yüksek.



- ElastiCache is an **In-Memory Cache service of AWS**.
- In-Memory Cache is a **temporary and fast** storage component. These components are used to reduce the workload of the main data storage device such as a database.
- AWS offers Redis and Memcached in-memory cache option which are popular in market.

► AWS ElastiCache



Burası bizim çalıştığımız bir ec2 uygulamamız olsun. Bu normalde sürekli olarak arada hiçbir uygulama olmadan rds ile iletişime geçiyordu. 100 kisiye girse 1000 kisi de girse sürekli rds ile giriyordu. Fakat biz onune bir elastiCache kurduk diyelim. Sonrasında uygulamız ilk sorgusunda önce database'e gidiyor. Sorgulama atıyor çünkü daha içinde tutacağı bilgiye dataya sahip değil. Daha sonra biz ayarladığımız bilgileri yüklüyoruz. Ve böylece bir sonraki adımda artık bellekte bilgileri olduğu için database'e sorgu atmıyor. Bu durumda su olmuş oldu: aynı sorulara artık database değil elastiCache cevap verecek ve database üzerindeki yük azalmış olacak ve performans artmış olacak. Dolayısıyla bizim kullandığımız web uygulamamızın performansı artmış olacak. Fiziksel olarak bir RAM gibi düşünebiliriz.

CLARUSWAY
WAY TO REINVENT YOURSELF

Redis 			Memcached 
Sub-millisecond latency	+	+	Sub-millisecond latency
User friendly syntax	+	+	User friendly syntax
It supports many different programming languages C, C++, java, python, etc.	+	+	It supports many different programming languages C, C++, java, python, etc.
Redis supports strings, lists, sets, sorted sets, hashes, bit arrays, and hyperloglogs	+	-	Memcached supports only strings
It doesn't support multithreaded architecture	-	+	It supports multithreaded architecture. It means that it has multiple processing cores. This allows you to handle multiple operation.
It supports Snapshot	+	-	It doesn't support Snapshot
It supports Replica	+	-	It doesn't support Replica

Farklı tiplerle uğrarsak redis kullanabiliriz. Multithreaded için memcached Snapshot ile çalışacak redis Replica alacak redis

► AWS ElastiCache

- ElastiCache gives you a bit more flexibility. It can front just about any database, but really **excels** being placed in **front of RDS**.
- Redis can be more than just a cache - it can be a standalone database as well (**More Than a Cache**.)
- **Memcached** >> No Failover or Multi-AZ supported
- **Redis** >> Failover or Multi-AZ supported

Asil olayimiz bu kisim.
Devopsta siklikla kullanacagimiz icin iyi bilmemiz gerek.
SINAVDA DAHA COK CIKAN BR SERVIS.

1 ► Amazon DynamoDB

► DynamoDB What is DynamoDB?



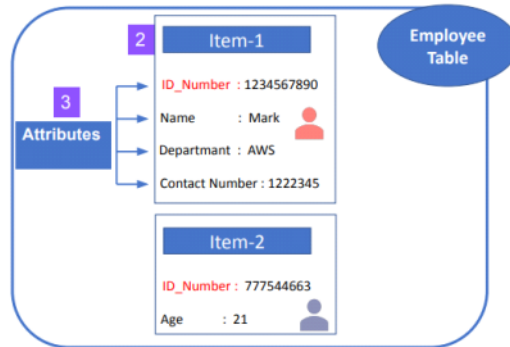
NoSQL tarzı bir database.
Mongodb gibi engine'leri var, mariadb vardı mesela burada da mongodb.
Ama burada engine seçilmiyor. Dynamodbye ihtiyaçlarını giriyosun o gerekli olanı yapıyor. Burada sadece bilgileri giriyoruz. Geri kalan tüm işlemleri(instance açma,, bağlantı kurma, tablo oluşturma vs) hepsini o yapıyor
Noqlde join fonksiyonu calismaz sadece dynamodb de degil.

- Amazon DynamoDB is a **NoSQL database** service
- Unlike RDS, you don't need to stick pre-determined schema. Instead of Schema, DynamoDB uses **flexible tables**.
- Amazon DynamoDB is a **fully-managed** database. (also **Serverless**)
- DynamoDB doesn't **have Join function**.

► DynamoDB

Structure of DynamoDB?

- 1- **Table** is a collection of data.
- 2- Each table consist of **items**. In the Picture, **item** represents a **person**.
- 3- **Attributes** are specific feature of the items.



ID number bos olamaz.
Primary keyler üzerinden biz join fonksiyonunu uygulayabiliyoruz.

```
CREATE TABLE 'clarusway'. 'Personal_Info_2' (
  'ID_number' INT NOT NULL,
  'Name' VARCHAR(45) NULL,
  'Surname' VARCHAR(45) NULL,
  'Gender' VARCHAR(45) NULL,
  'Age' INT NULL,
  'Department' VARCHAR(45) NULL,
  PRIMARY KEY ('ID_number'));
```

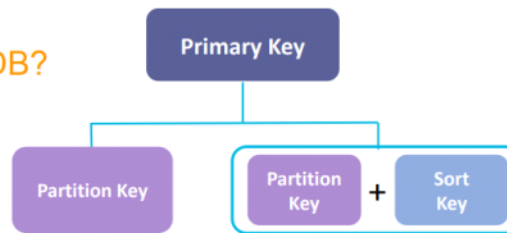
Tablo bizim olmazsa olmazımız. Burada employee diye adlandırılmış. Alt daldaki itemler var. Bir esya yani. Eşyanın özellikleri farklı olabiliyor itemlerin yani. ID number olmak zorunda. Itemi bizim ayarladığımız primary key olması lazım ve id numberi primary key olarak seçiyoruz. Bizim personal infoda böyle bir şey yoktu. Hepsi aynı attribute'leri sahip olmak zorundaydı.

CLARUSWAY

Unlike RDS, you can enter different **attributes** for each people.

► DynamoDB

Structure of DynamoDB?



Primary key, o itema ulaşabilmek için belirlediğimiz unique bir özellik. Tek olduğu zaman partition key deniliyor ya da baska bir aratma yöntemi daha koymussak partition key + sort key.

DynamoDB uses **Primary Keys** to **uniquely identify each item** in a table. When you create a table, in addition to the table name, you must specify the primary key of the table.

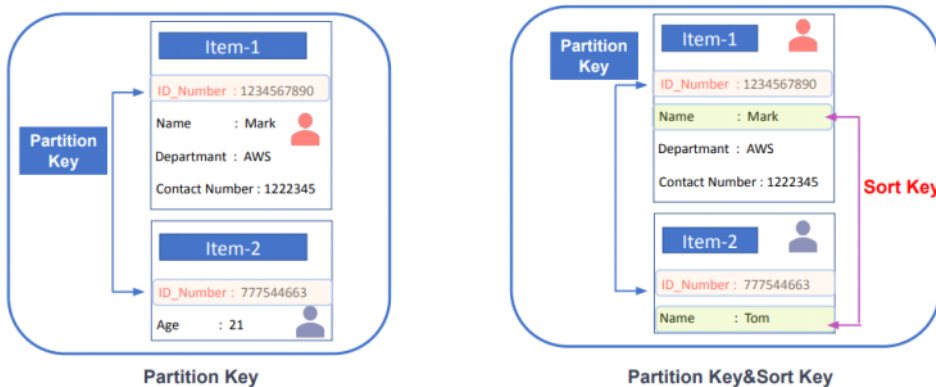
The **sort key** is used to sort and order **items** in a partition. Multiple items with the same partition key value are feasible, but they should have different sort key values. This means you can have multiple items with the same partition keys, but the sort key can not be the same.

There are two different kinds of Primary Key model : **Partition Key** and **Partition Key & Sort Key**.

CLARUSWAY

► DynamoDB

Structure of DynamoDB?



İlkinde partition olarak id number.
İkincisinde hem id hem sort key olarak name seçilmiş mesela.



► DynamoDB

Capacity Modes

Eventually Consistent Reads

- When you read data from a DynamoDB table, the response might not reflect the results of a recently completed write operation. The response might include some stale data.
- If you repeat your read request after a short time, the response should return the latest data.

Strongly Consistent Reads

- When you request a strongly consistent read, DynamoDB returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful.

► DynamoDB

Capacity Modes

DynamoDB RCU

- Read Capacity tells us how much data can be read from a DynamoDB table. Read Capacity is measured in RCUs. (Read Capacity Unit)

DynamoDB read requests can be either strongly consistent, eventually consistent, or transactional.

- An *eventually consistent* read request of an item up to 8 KB requires *one read* request unit.
- A *strongly consistent* read request of an item up to 8 KB requires *two read* request unit.
- A *transactional* read request of an item up to 8 KB requires *four read* request units.

DynamoDB Write Capacity

- DynamoDB Write Capacity tells us how much data can be written to a DynamoDB table. Write Capacity is measured in WCUs.
- WCU or "Write Capacity Unit" represents one write per second, for an item up to 1 KB in size.

► DynamoDB

Secondary Indexes

- In a relational database, an index is a data structure that lets you perform fast queries on different columns in a table. You can use the CREATE INDEX SQL statement to add an index to an existing table, specifying the columns to be indexed.
- You can create one or more secondary indexes on a dynamodb table. A *secondary index* lets you query the data in the table using an alternate key, in addition to queries against the primary key. DynamoDB doesn't require that you use indexes, but they give your applications more flexibility when querying your data.
- After you create a secondary index on a table, you can read data from the index in much the same way as you do from the table.

DynamoDB supports two kinds of indexes:

- Global secondary index – An index with a partition key and sort key that can be different from those on the table.
- Local secondary index – An index that has the same partition key as the table, but a different sort key.

Each table in DynamoDB has a quota of 20 global secondary indexes (default quota) and 5 local secondary indexes.

► DynamoDB Streams

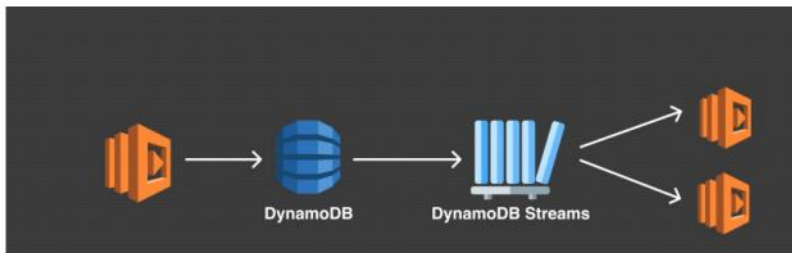
DynamoDB Streams is an optional feature that captures data modification events in DynamoDB tables. The data about these events appear in the stream in near-real time, and in the order that the events occurred.

Each event is represented by a *stream record*. If you enable a stream on a table, DynamoDB Streams writes a stream record whenever one of the following events occurs:

- A new item is added to the table: The stream captures an image of the entire item, including all of its attributes.
- An item is updated: The stream captures the "before" and "after" image of any attributes that were modified in the item.
- An item is deleted from the table: The stream captures an image of the entire item before it was deleted.

► DynamoDB Streams

- Time-ordered sequence of item-level changes in table
- Stored for 24 hours
- Inserts, updates, and deletes
- Combine with Lambda functions for functionality like stored procedures
- FIFO time sequence



Amazon DynamoDB Streams

DigitalCloud
TRAINING



1. Application inserts / updates / deletes item
2. A record is written to the DynamoDB stream
3. A Lambda function is triggered
4. The Lambda function writes to CloudWatch Logs

► Global Tables

it's a way of replicating your DynamoDB tables from one region to another.

You will need **DynamoDB streams turned on** in order to enable this.

Managed Multi-Master, Multi-Region Replication

- Globally distributed applications
- Based on DynamoDB streams
- Multi-region redundancy for disaster recovery or high availability
- No application rewrites
- Replication latency under **1 second**

► DynamoDB Accelerator (DAX)

What is DynamoDB Accelerator (DAX) ?

In-Memory Cache : DAX can reduce DynamoDB response times from milliseconds to microseconds.

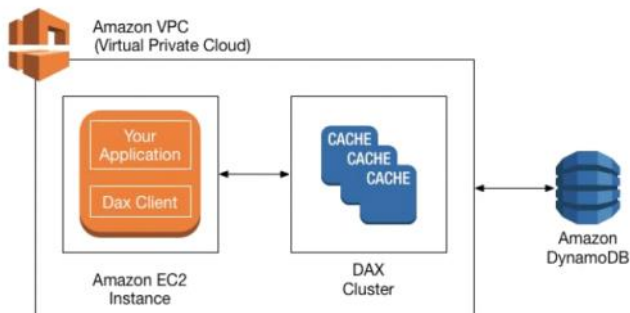
Fully managed, high available, in-memory cache

10x performance improvement

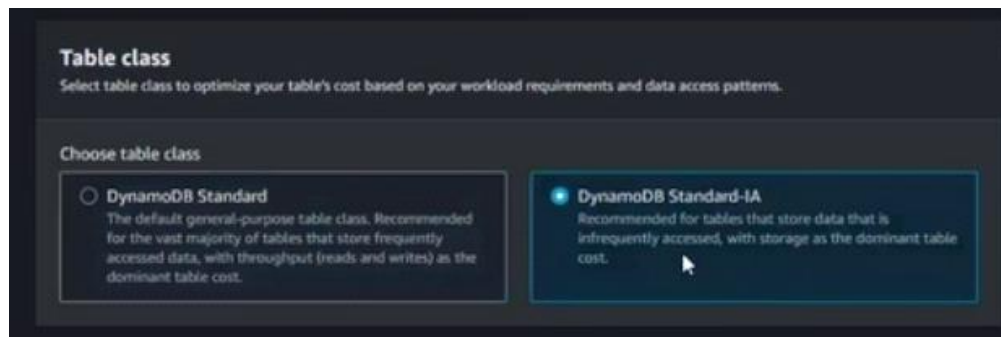
Reduces request time from millisecond to **microseconds** – even under load

Compatible with DynamoDB API calls

► DynamoDB Accelerator (DAX)



SLADICMAV https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX_concepts.html



Aynı s3teki gibi. Tabloya çok mu erişilecek az mı erişilecek? Ona göre fiyatlandırma sağlıyor.

▼ Capacity calculator

Average item size (KB)

1

Item read/second

1

Read consistency

Eventually consistent

Item write/second

1

Write consistency

Standard

Read capacity units

1

Write capacity units

1

Region

us-east-1

Estimated cost

USD 0,59 / month

Read consistency databaseden okunacak bilginin tutarlılığı, write consistency ise bu database yazılacak bilgiler.burada daha çok okuma işlemi yapılır. Yeni bilginin girmesinden ziyade daha çok var olan bilginin okunması yapılır.

Read consistency

Eventually consistent

Eventually consistent

Strongly consistent

Transactional

Eventually consistent sunu yapar:cok hizli cevap verir fakat o an giren bilgiyi gosteremeyebiliriz diyor hizli cevap verdigi icin. Strongly consistent ise once bilginin girilmesini bekliyor ve sonra cevap verme kismina geciyor. Transactionalda ise read ve write islemi var. Yani mesela para gonderilecek. Bunun icin hem senin hesabının okunup yazilabilmesi hem karsi tarafın hesabın okunması gozukmesi lazim. Karsi tarafın hesabında problem olunca işlem yarım kalır.bu yuzden transactionalda olay iki taraftanda alınan bilgiler dogrultusunda isleme gecmesi seklinde gercekleşiyor.

Saniyede 10 item yazilip okunması istersek mesela fiyatlandırma degisir.

Capacity mode

☐ Provisioned

Manage and optimize your costs by allocating read/write capacity in advance.

☒ On-demand

Simplify billing by paying for the actual reads and writes your application performs.

► Maximum table throughput - optional

Specify maximum on-demand read or write (or both) throughput for tables and secondary indexes to help keep your throughput usage and costs bounded. By default, maximum table throughput does not apply and on-demand throughput is only limited by the default DynamoDB table quotas.

Ondemandte butun secenekler gidr. Cunku ne kadar ihtiyac varsa o kadar kullanir. On demand en pahali secim.

Secondary indexes

Delete

Create local index

Create global index

Name	Type	Partition key	Sort key	Projected attributes
------	------	---------------	----------	----------------------

No indexes

Use secondary indexes to perform queries on attributes that are not part of your table's primary key.

Create global index

Dataya ulasmak yerine daha fazla secenek olsun istiyorsak bu kisma ihtiyac duyuyoruz.