# Which NoSQL database is right for you?

Jason Hunter

Principal Solutions Architect, DynamoDB
AWS

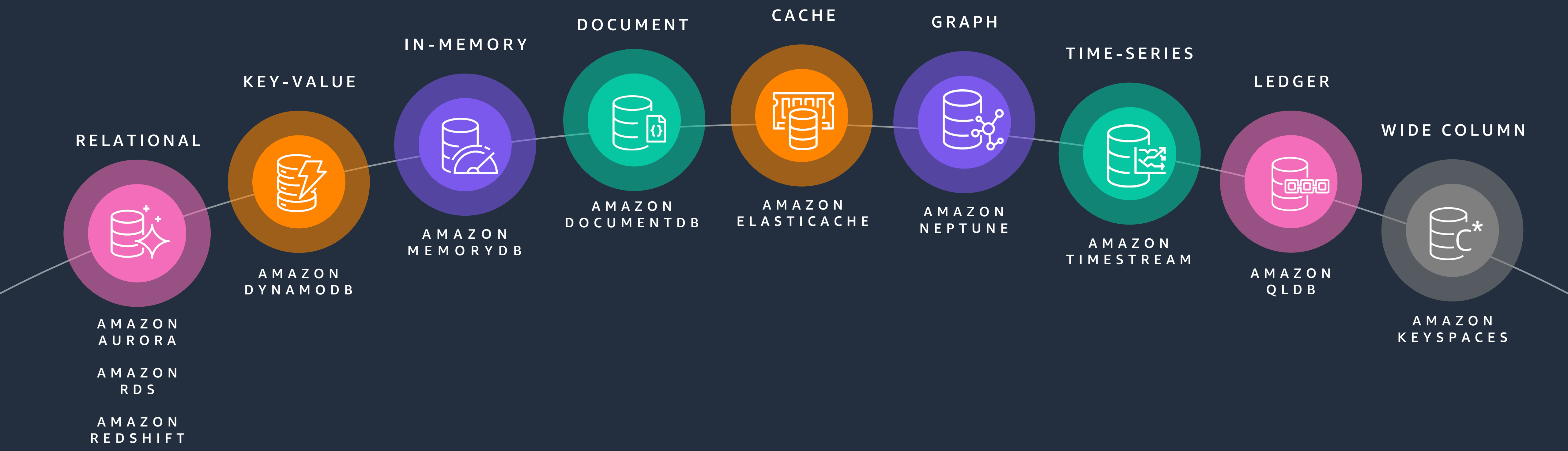
Siva Karuturi

Specialist Solutions Architect, In-Memory Databases
AWS

# Agenda

- Purpose-built databases
- In Memory, Key-Value, Document data types
- Review of AWS database services and key characteristics
- Choosing a database based on customer use case

# Broadest portfolio of purpose-built databases



- RELATIONAL: AMAZON AURORA, AMAZON RDS, AMAZON REDSHIFT
- KEY-VALUE: AMAZON DYNAMODB
- IN-MEMORY: AMAZON MEMORYDB
- DOCUMENT: AMAZON DOCUMENTDB
- CACHE: AMAZON ELASTICACHE
- GRAPH: AMAZON NEPTUNE
- TIME-SERIES: AMAZON TIMESTREAM
- LEDGER: AMAZON QLDB
- WIDE COLUMN: AMAZON KEYSPACES

[1] Read the article on Forbes.com | [2] Read the report

[2] Source: 2022 internet minute, DBTA, Boston Consulting

# Review of NoSQL databases

| | Amazon MemoryDB | Amazon ElastiCache | Amazon DocumentDB | Amazon DynamoDB |
|---|---|---|---|---|
| **Manageability** | Fully managed, durable, in-memory database service that delivers ultra-fast performance | Fully managed, durable, in-memory caching service that delivers ultra-fast performance | Fully managed, highly available and secure<br>**Cluster based** | Fully managed, highly available and secure<br>**Serverless -** No servers to manage |
| **Scale** | Performance at any scale | Performance at any scale | Highly scalable | Performance at any scale |
| **API** | Compatible with OSS Redis API | Compatible with OSS Redis & Memcached | Compatible with MongoDB | Proprietary API / PartiQL |
| **Category** | In Memory,<br>Key-Value | In Memory,<br>Key-Value | Document Databases | Key-Value |

# Amazon MemoryDB for Redis

# Amazon MemoryDB for Redis

Durable, in-memory Redis-compatible database, that is fully managed and purpose-built for:

- Modern applications with microservices architectures
- Accessing data with ultra-fast performance including microsecond read and single-digit millisecond write latency.
- Storing data across multiple AZs for high durability & fast recovery
- Enabling customers to quickly build applications using flexible and friendly Redis data structures and APIs like streams, hashes, lists and sets.
- Providing high concurrency to handle millions or requests per second.

# MemoryDB manageability & scalability

## **Manageability** - fully managed

- Launch new MemoryDB clusters using AWS Console, AWS CLI or SDK or using CloudFormation templates using pre-configured parameters
- MemoryDB automatically keeps your clusters up-to-date with new updates, and you can easily upgrade your clusters to the latest versions of Redis.

**Integration** with other AWS services such as CloudWatch, Amazon VPC, CloudTrail, and Amazon SNS for monitoring, security, and notifications.

## Scalability

- Scale your cluster horizontally by adding/removing shards or vertically by moving to larger/smaller node types.
- Each cluster can have up to 500 shards. Each shard has one primary node and up to five replica nodes.
- You can scale to more than 100 TB of storage per cluster (with 1 replica per shard)
- Your cluster continues to stay online and support read and write operations during a Scaling operation.

# MemoryDB key features

## Replication

Supports High Availability using Async Replication between primary and read replicas using Multi-AZ transaction log

## Indexing

Use Redis data structures to create

Secondary indexes on top level attributes

## Data Streaming

Supported through Redis Streams data structure

## ACID Transactions

Strong read after write consistency on primary nodes, with eventual consistency on replicas

Redis being single threaded provides consistency and isolation by design

High Data durability & Consistency using a Multi-AZ transactional log

# MemoryDB schema, API & connectivity

## Schema

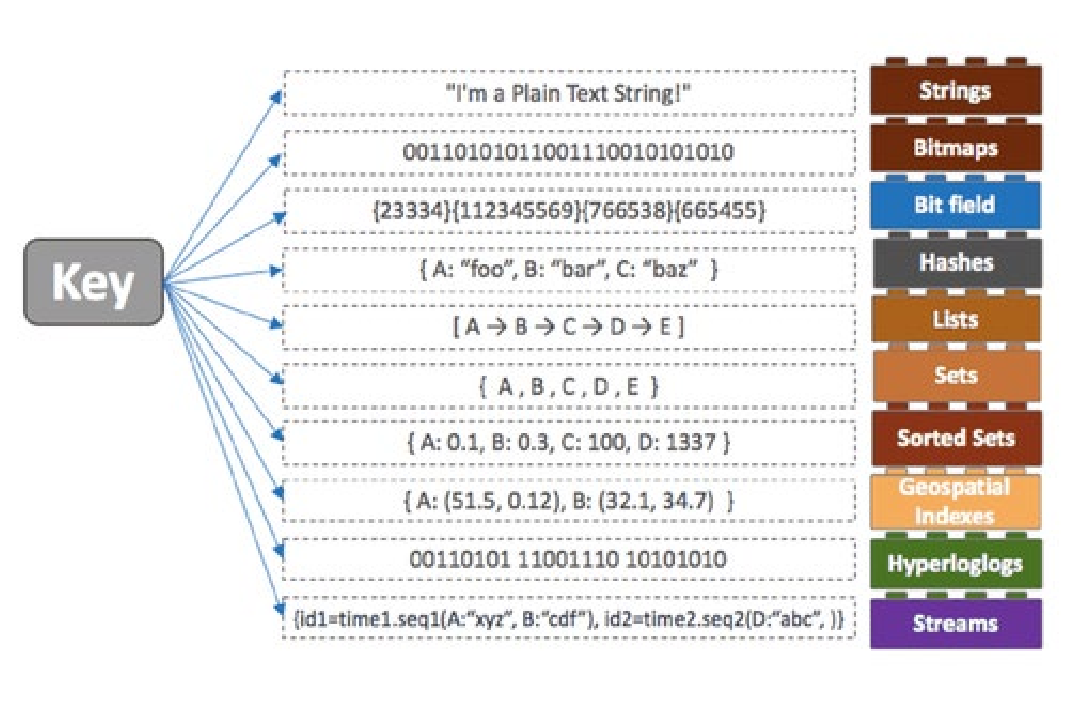
- Flexible in-memory key-value datastore leveraging redis data structures

## API

- Redis offers flexible APIs, commands, and data structures like streams, sets, and lists, to build agile and versatile applications

## Connectivity

- Accessible via Cluster Endpoint within VPC
- Supports VPC peering for private connectivity between VPCs
- Supports Hybrid Architecture using VPN and Direct connect



# MemoryDB pricing

Three dimensions for pricing:

- **On-Demand instance hours:** The amount of node capacity for a cluster, charged per instance-hour
- **Data Written:** Volume of data written to the MemoryDB cluster, charged per GB
- **Snapshot Storage:** Storage associated with the automated and user-initiated snapshots you take, charged per GB-month

Refer to MemoryDB pricing page for more details

# MemoryDB use cases

## Web and mobile

User content data stores
Session management
Geospatial indexing
Chat and message queues

## Retail

Customer profiles and accounts
Inventory tracking and fulfillment

## Gaming

Player data stores
Session history
Leaderboards

## Banking and finance

User transactions
Fraud detection

## Media and entertainment

User data stores
Real-time streaming

## IoT

Streaming device data
Operational insights

# MemoryDB customer story

"Our media platform uses a microservices architecture and we are looking for the right database to support some of our platform requirements. We're excited about the launch of Amazon MemoryDB, as it fulfills our need for a durable, in-memory database that supports our short and long-term architectural plans..”

— **Charles Zhao**, Manager, MCE Media Workflow Infrastructure - Netflix

Netflix is one of the world's leading entertainment services with more than 200 million members in over 190 countries enjoying TV series, documentaries, and feature films across a wide variety of genres and languages.

# Payment Processing use case - A leading global payments company with seamless, secure and innovative solutions to payment card issuers — with its solutions used over 1300+ financial institutions and retail companies in 200 countries

## Challenge:

To create a payment tokenization solution that can handle high throughput and meet ultra -fast read and write performance SLAs while reducing the footprint and transfer of sensitive data which comes under Payment Card Industry requirements.

## Solution:

Used Fully Managed Amazon MemoryDB for Redis **Service**to:

- Replace PAN/PII data with a random number token and store them in one central secured storage area.
- Store billion of tokens in memory with single digit millisecond write latency.
- Easily scale to thousands of RPS with minimum overhead and zero data loss durability

## Result

Low latency application that could store 4+ billion tokens and handle greater than 100,000 read IOPS and 20,000 write IOPS

# MemoryDB customers

# MemoryDB summary

✓ Fully managed, durable, in-memory database service that delivers microsecond read and single-digit millisecond write latencies with up to 160 million TPS per cluster.

✓ Flexible and friendly Redis data structures and APIs. Easily build new applications or migrate existing Redis applications with almost no modification.

✓ High Data durability using a Multi-AZ transactional log providing fast database recovery and restart.

✓ Strong read after write consistency on primary nodes, with eventual consistency on replicas

✓ Multi-AZ availability with automatic failover, and detection of and recovery from node failures.

✓ Support for up to 500 nodes and more than 100 TB of storage per cluster (with 1 replica per shard).

✓ Scale your cluster horizontally by adding/removing shards or vertically by moving to larger/smaller node types.

# Amazon ElastiCache

# Amazon ElastiCache: ultra-fast, in-memory caching service

## Redis & Memcached compatible

Fully compatible with open source Redis and Memcached

## Extreme performance

In-memory data store and cache for microsecond response times

## Secure and reliable

Network isolation, encryption at rest/transit, HIPAA, PCI, FedRAMP, multi AZ, and automatic failover

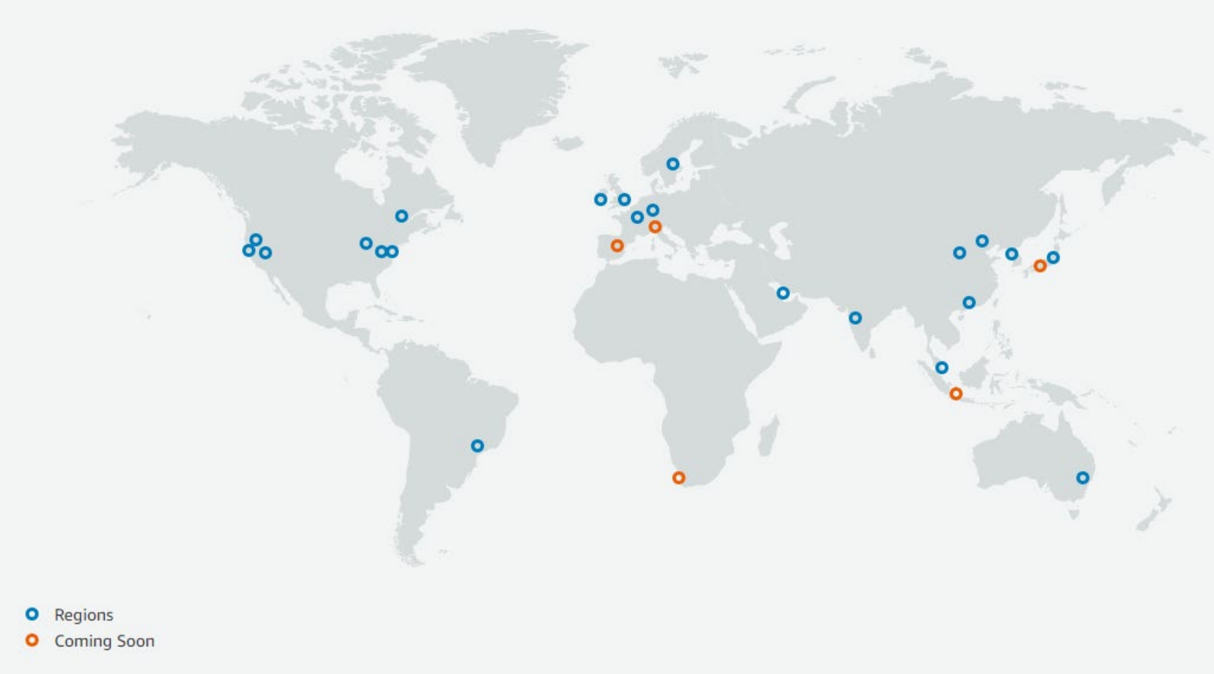
## Scalable and highly available

Scale writes and reads with sharding and replicas

# Amazon ElastiCache for Redis

BUILT ON THE FULL POWER OF AWS

Amazon ElastiCache for Redis is a fully managed, blazingly fast in-memory caching service for use as a Database Cache, Message broker & Queue with sub millisecond response times



- Available in all AWS regions
- Multi-AZ auto failover
- Cross region replication with Global Datastore
- Up to 500 nodes per cluster & 340 TB database
- Data tiering to expand storage capacity
- Scheduled snapshot support
- Scale out/in (sharded configuration)
- Scale up/down (all configurations)

# ElastiCache key features

## Replication

- Supports High Availability using Async Replication between primary and read replicas
- Supports cross region replication with Global Datastore

## Storage

- Leverage Data tiering to expand storage beyond memory into local attached storage

## Indexing

- Use Redis data structures to create Secondary indexes on top level attributes

## Data Streaming

- Supported through Redis Streams data structure

## ACID Transactions

- Strong read after write consistency on primary nodes, with eventual consistency on replicas

# ElastiCache manageability & scalability

## **Manageability** - fully managed

- Launch new ElastiCache clusters using AWS Console, AWS CLI or SDK or using CloudFormation templates using pre-configured parameters
- ElastiCache automatically keeps your clusters up-to-date with new updates, and you can easily upgrade your clusters to the latest versions of Redis.

**Integration** - with other AWS services such as CloudWatch, Amazon VPC, CloudTrail, and Amazon SNS for monitoring, security, and notifications.

## Scalability

- Scale your cluster horizontally by adding/removing shards or vertically by moving to larger/smaller node types.
- Autoscaling – Automatically add shards or replicas based on pre-defined rules
- Each cluster can have up to 500 shards. Each shard has one primary node and up to five replica nodes.
- You can scale to more than 340 TB of storage per cluster
- Your cluster continues to stay online and support read and write operations during a Scaling operation.

# ElastiCache schema, API & connectivity

## Schema

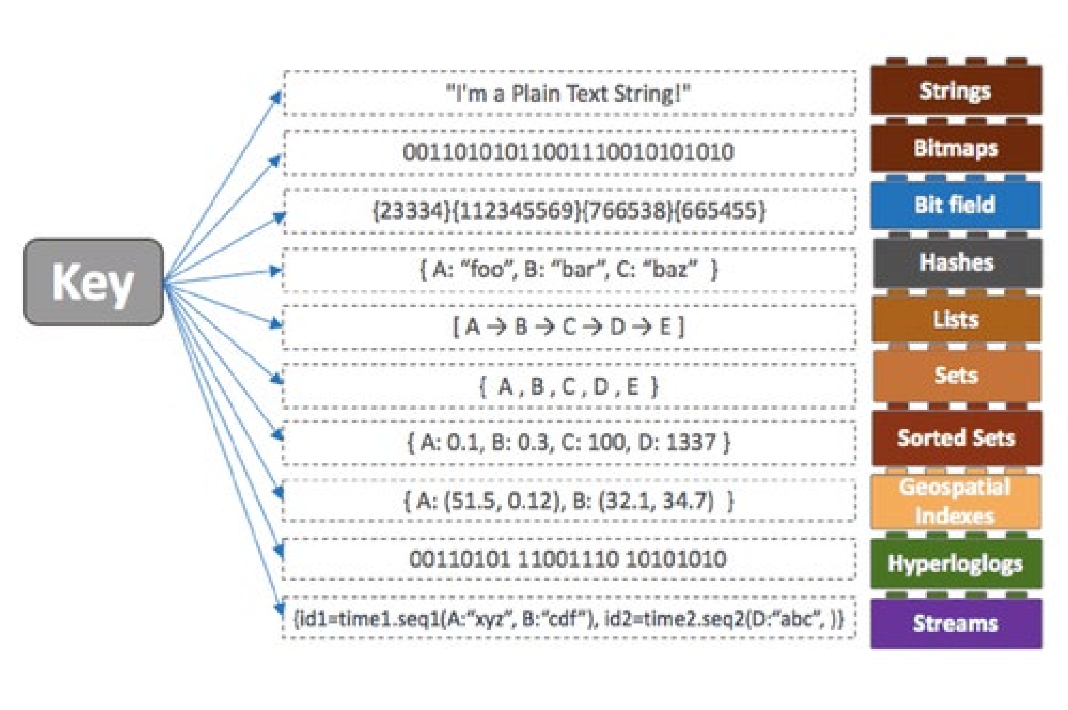
- Flexible in-memory key-value datastore leveraging redis data structures

## API

- Redis offers flexible APIs, commands, and data structures like streams, sets, and lists, to build agile and versatile applications

## Connectivity

- Accessible via Cluster Endpoint within VPC
- Supports VPC peering for private connectivity between VPCs
- Supports Hybrid Architecture using VPN and Direct connect



# ElastiCache pricing

Three dimensions for pricing:

- **On-Demand instance hours:** The amount of node capacity for a cluster, charged per instance-hour
- **Data transfer:** Volume of data transferred from your application to the ElastiCache cluster, charged per GB
- **Backup Storage:** Storage associated with the automated and user-initiated snapshots you take, charged per GB-month

Refer to ElastiCache pricing page for more details

# Amazon ElastiCache Use Cases

Caching

Real-time analytics

Gaming leaderboards

Geospatial

Media streaming

Session store

Chat apps

Message queues

Machine learning

# ElastiCache customer story

## Challenge

Peloton’s leaderboard requires high elasticity, low latency, and real-time processing to deliver customizable rider data for the community of users riding together virtually from the comfort of their homes.

## Solution

- **Use Amazon ElastiCache** for Redis as an in-memory data store to  provide scale and performance to process streaming data for on-demand, live leaderboard.
- Required millisecond response time for real-time leaderboards with thousands of users and up to 20k requests/sec
- Use ElastiCache Redis as a in-memory data store for real-time data computation
- Use RDS MySQL as the persistent data store

https://aws.amazon.com/solutions/case-studies/peloton/

# Amazon ElastiCache customers

## Software & internet

Adobe

intuit

sumologic

duolingo

## Ride-hailing

Grab

Careem

lyft

## Financial services

Capital One

experian

NuData Security mastercard

KICKSTARTER

SOCIETE GENERALE

Traiana

## Gaming

zynga

BANDAI NAMCO Studios

UBISOFT

EPIC GAMES

## Media and entertainment

hulu

COMCAST

The Walt Disney Company

mlbam

A&E

News Corp

BuzzFeed

## Social media

ancestry

cookpad

Tapjoy

## Telecommunications

vodafone

gogo

igaworks

## Travel

airbnb

Expedia

## Industrials & services

ThermoFisher

GE

McDonald's

## Logistics & operations

here

esri

## Publishing

BBC

ZIFF DAVIS

PBS

## Other

Fender

instacart

PELOTON

aws

# ElastiCache summary

✓ Fully managed, in-memory caching service that delivers sub millisecond response times

✓ Flexible and friendly Redis data structures and APIs. Easily build new applications or migrate existing Redis applications with almost no modification.

✓ Strong read after write consistency on primary nodes, with eventual consistency on replicas

✓ Multi-AZ availability with automatic failover, and detection of and recovery from node failures.

✓ Support for up to 500 nodes and more than 340 TB of storage per cluster

✓ Scale your cluster horizontally by adding/removing shards or vertically by moving to larger/smaller node types. Leverage autoscaling.

✓ Cross region replication with Global Datastore

✓ Data tiering to expand storage capabilities beyond memory

# JavaScript Object Notation (JSON) with Redis

ElastiCache and MemoryDB are fully managed services that customers use to run Redis and have data sets that use JSON, an interchange format that's popular due to its flexibility and ease of use.

Historically, if a customer wanted to load, store, and access JSON in an AWS-native service for Redis,

- They would have to write, deploy, and manage custom code to convert JSON into a data structure that Redis supports, such as a String, Hash, or SortedSet.
- Alternatively, they could store their data in a database designed specifically for JSON documents, and write code to have their Redis applications interact with these systems.

These approaches often increase complexity and cost and negatively impact application performance.

# New JSON data type support for ElastiCache for Redis

- Build applications requiring the low latency and high throughput performance with powerful Redis APIs on JSON
- Natively store, fetch, and update JSON data inside ElastiCache clusters
- New set of commands for manipulating JSON structures
- No Serialization or Deserialization necessary
- Each JSON document is stored as the value pertaining to a unique Redis key
- Available for ElastiCache in all regions where ElastiCache is available, for Redis v6.2 and above
- The JSON data type and associated commands are available for a number of popular Redis clients, including redis-py for Python, node_redis for Node.JS, and Lettuce for Java

# Amazon DynamoDB

# DynamoDB manageability & scalability

DynamoDB is **serverless**

- No maintenance windows (99.99% and 99.999% uptime)
- No versions to manage, always on the latest version
- No need to make difficult sizing decisions
- Looks like a web endpoint behind which "magic" happens

Virtually unlimited **scaling**

- Runs on many thousands of servers
- Used by Amazon.com, peaking at 89 million requests per second during Prime Day
- With a good partition key, can scale automatically up and down
- Can scale to 0 (with no throughput charges) then hit 2x past peak the next second

# DynamoDB differentiating features

## Global Replication

- Active-active global table with ability to read and write in all regions

## Indexing

- Sort key and secondary indexes provide many query patterns beyond key-value

## Change Data Capture

- Take action on mutation events with DynamoDB Streams and Kinesis Data Streams, with built-in Lambda trigger support

## ACID Transactions

- Full ACID guarantees for up to 25 updated items, including across tables

# DynamoDB schema, API & connectivity

## Schema

- Flexible – multiple data types are often held in the same table, must only define the partition key and optionally sort key (often named PK/SK to keep them generic), can store JSON

## API

- Simple get/put/query/scan proprietary API as well as PartiQL
- If you can express something, it will execute quickly
- Large variety of client languages, most have nice syntactic sugar

## Connectivity

- Public, per region https endpoint or VPC GW endpoint for private connectivity from a VPC
- Rich integration with other AWS services

# DynamoDB pricing

## Pricing

- Primarily storage (in GB-months), reads (4K blocks), and writes (1k blocks)
- Eventual consistency is charged 1/2 of strong consistency
- Can be "On Demand" (pay per request) or "Provisioned" (specify a consumption rate)
- Reserved capacity provides discounts over 50% for annual, over 77% for 3 year

Refer to DynamoDB pricing page for more details

# DynamoDB use cases by industry

CUSTOMERS RELY ON DYNAMODB TO SUPPORT THEIR MISSION-CRITICAL WORKLOADS

## Banking and finance

Fraud detection

User transactions

Mainframe offloading

(Capital One, Vanguard, Fannie Mae)

## Ad tech

User profile stores

Metadata stores for assets

Popular-item cache

(AdRoll, GumGum, Branch, DataXu)

## Gaming

Game states

Leaderboards

Player data stores

(Riot Games, Electronic Arts, PennyPop)

## Retail

Shopping carts

Workflow engines

Customer profiles

(Nordstrom, Nike, Zalando, Mercado Libre)

## Software and internet

Metadata caches

Ride-tracking data stores

Relationship graph data stores

(Uber, Lyft, Swiggy, Snap, Duolingo)

## Media & Entertainment

User data stores

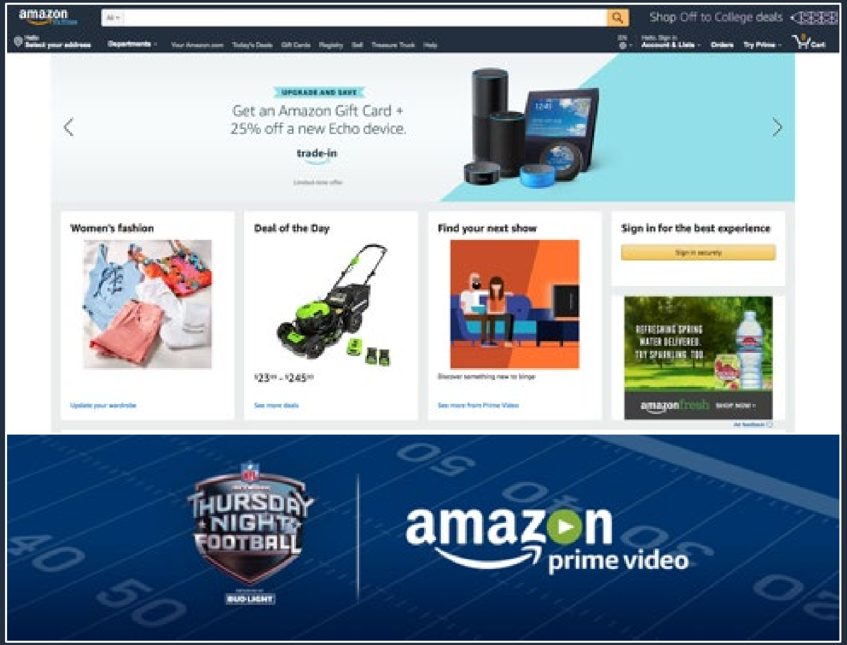
Media metadata stores

Digital rights management stores

(Airtel Wynk, Amazon Prime, Netflix)

aws

# amazon



Over the course of the 66-hour Prime Day in 2021, these [Amazon.com properties and systems] made **trillions of API calls while maintaining high availability with single-digit millisecond performance, and peaking at 89.2 million requests per second**.

DynamoDB powers multiple high -traffic Amazon properties and systems including Alexa, the Amazon.com sites, and all Amazon fulfillment centers.

The internal Amazon.com Herd system supports 100s of millions of active workflows.

Migrated from Oracle to DynamoDB

- **Improved customer experience:** Workflow processing delays dropped from 1 second to 100 milliseconds.
- **Reduced cost:** Scaling and maintenance effort dropped 10 times.
- **Reduced complexity and risk:** Retired more than 300 Oracle hosts.

# DynamoDB summary

- Serverless, unlimited scale, reliable single-digit millisecond latency, even for millions of reads and writes per second, or spiky workloads
- Key-value, with a sort key and secondary indexes for alternative views
- Simple get/put/query/scan proprietary API

- Supports strong and (half price) eventual consistency, active-active cross-region replication, ACID transactions, and streams to observe all data changes
- Doesn't directly support aggregations, geospatial, text-based queries
- Supports push-button data export to S3 for analyzing the data with other tools
- Has an optional API-compatible write-through caching layer in front called DAX for sub-ms reads

# Hundreds of thousands have chosen DynamoDB

expedia group
lyft
amazon
airbnb
Adobe
COMCAST
UNDER ARMOUR
mlbam
Arcesium
Avalara
camerite

twilio
TRD
tinder
zynga
FANDUEL
SUPERCELL
Storybulbs
armut
BLACK & VEATCH
CEQUENS

duolingo
dataxu
REDFIN
VidRoll
AdRoll
mapbox
PLANGRAM
Amplitude
Autotask
boomtrain
CLEARCODE

clutch
Clever
CYLANCE
ČESKÁ SPOŘITELNA
Rue La La
druva
embraco
EasyStore
eMoney
etobee
EVRYTHNG

ITT
JUMIA
JUMP RAMP GAMES
KEMPPI
KOCH
Maas GLOBAL
LA NACION
miovision rethink traffic
MARKSYSTEMS
NEWFORMA
Nishitetsu

TTI
playkot
proov by scalabill.it
Q
REALITY REFLECTION
RUMBLE
Reltio
rentcars.com
DOORDASH
SCANIA

triumph learning
TRIOMOBIL
ticketea
ZOOMDATA
VIVID GAMES
tuya
UNIwise
ViaSat
whitepages
Workiva

FASHION VALET
Australian Government Geoscience Australia
ZOONA LET'S MAKE IT REAL
Financial Engines
Global Intellect Service
ibotta
frantishex
HIJUP
IGG Software
shop pr shoppers 24/7

GAMETIME
hydrane
Iot-Inc.
SPOKEO
Optimizely
Paragora
PEOPLEADMIN
STARTDATE LABS
SendGrid
Serenova shine bright

ServiceChannel
TalkingData
天润融通 证券代码：835653
THINSLICES
BYJU'S The Learning App
SGK A Matthews International company

https://aws.amazon.com/dynamodb/customers/

# Amazon DocumentDB (with MongoDB compatibility)

# DocumentDB manageability & scalability

## Manageability - fully managed

- Cluster based, managed servers
- Simple management – similar to other managed services

## Scalability – highly scalable

- Millions of reads/sec with up to 15 read replicas
- Tens of thousands of writes/sec with single primary
- Storage scales automatically up to 64 TiB
- Resizing and scaling out reads takes minutes due to separation of compute and storage

# DocumentDB key features

## Global Replication

Global cluster with 1 primary region (reads and writes) and up to 5 secondary regions (reads)

Secondary region can be manually promoted in less than a minute to become primary

## Indexing

Various index types can be created on top level, nested and array fields

## Data Streaming

Supported through Change Streams

## ACID Transactions

Operations on single item are atomic, isolated and durable

ACID transactions on multiple documents are supported now in 4.0

# DocumentDB schema, API & connectivity

## Schema

Flexible – Document-oriented, no need to define any fields for a collection

Unique _id field is automatically added for every document and its value can be specified by the application, or auto generated by the database

## API

Compatible with MongoDB

Customers can use existing MongoDB code, drivers and open source tools from the MongoDB ecosystem

Rich querying capabilities including querying on nested and array fields, aggregations, geospatial and joins

## Connectivity

Cluster located in a VPC and can only be accessed from within that VPC, or through private connectivity to that VPC like VPC peering and SSH tunneling

# DocumentDB pricing

## Pricing

Compute instances, priced per second running

Data read/written to the storage volume, priced by millions of IOs

Storage, priced by GB-month

Backup storage, priced by GB-month

Refer to DocumentDB pricing page for more details

# DocumentDB use cases

Migrations of existing MongoDB workloads, moving to managed

Rich querying capabilities for JSON documents

- Aggregations

- Ad hoc querying

- Index on any Key

Customers who want to use open source APIs

# DocumentDB customer story

## Challenge

When Plume, a leader in the smart home services industry, reached database scalability limits in MongoDB, it faced complicated application redesigns that would require onboarding new infrastructure experts.

## Benefits

Greatly reduced the need for in-house database management

Drastically increased scalability and functionality without increasing the workload of Plume staff.

" On one of our clouds, users are making like 50,000 API calls a minute, and Amazon DocumentDB is not really blinking an eye "

—**Robert Miller,** software development lead for public APIs, Plume

# DocumentDB key customers

amazon
BBC
freshop
Easy Online Groceries
HABBY
FINRA
eHealth®
Plume®
DOW JONES
punchh®
SAMSUNG
zulily
Capital One

https://aws.amazon.com/documentdb/customers

# DocumentDB summary

✓ Fully managed document database that supports tens of thousands of writes per second and millions of reads per second with millisecond latency

✓ Cloud native architecture - separation of compute and storage with single writer and up to 15 read replicas

✓ API compatible with MongoDB, great for migration of MongoDB workloads or other workloads that require a Document database with flexible schema and querying capabilities

✓ Rich querying capabilities – choice between strong and eventual consistency, indexing top level, nested and array fields, ACID transactions, aggregations, geospatial

✓ Supports read-only cross-region replication, streams to observe all data changes

# Review of NoSQL databases

| | Amazon MemoryDB | Amazon ElastiCache | Amazon DocumentDB | Amazon DynamoDB |
|---|---|---|---|---|
| **Manageability** | Fully managed, durable, in-memory database service that delivers ultra-fast performance | Fully managed, durable, in-memory caching service that delivers ultra-fast performance | Fully managed, highly available and secure<br>**Cluster based** | Fully managed, highly available and secure<br>**Serverless -** No servers to manage |
| **Scale** | Performance at any scale | Performance at any scale | Highly scalable | Performance at any scale |
| **API** | Compatible with OSS Redis API | Compatible with OSS Redis & Memcached | Compatible with MongoDB | Proprietary API / PartiQL |
| **Category** | In Memory,<br>Key-Value | In Memory,<br>Key-Value | Document Databases | Key-Value |

# Review of NoSQL databases / cont.

| | Amazon MemoryDB | Amazon ElastiCache | Amazon DocumentDB | Amazon DynamoDB |
|---|---|---|---|---|
| **Schema** | Flexible | Flexible | Highly flexible (schema less) | Flexible |
| **ACID Transactions** | No* | No | Yes | Yes (25 items) |
| **Connectivity** | VPC Only | VPC Only | VPC Only | Public and VPC GW |
| **Global Replication** | No | Yes (Active-Passive) | Yes (Active-Passive) | Yes (Active-Active) |

**MemoryDB for Redis provides strong read after write consistency on primary nodes with high data durability using Multi-AZ transaction log*

# Which database should I pick?

*Note: The following examples are not the only criteria for choosing a datastore*

# … I need to create personalized customer experiences and manage user profiles and preferences with microsecond read and single-digit millisecond write latency.

MemoryDB for Redis

(Stores data in-memory across up to hundreds of nodes for ultra-fast performance)

# … I'm looking to query on nested fields or arrays and don't want to remodel the data

DocumentDB

(DynamoDB would require data remodeling)

## … I need to use aggregations (average, sum, group by)

DocumentDB

(DynamoDB does not support aggregation functions)

# … I want to run high concurrency streaming data feeds to ingest user activity and support trillions of requests per day

MemoryDB for Redis

(Access data with ultrafast performance – Process more than 13 trillion request per day and over 160 million requests per second)

# … I need to run ad-hoc queries

## DocumentDB

DocumentDB flexible querying supports joins, aggregations, and the query optimizer dynamically selects the index appropriate for the query

## … I want to build applications quickly using versatile data structures like streams, lists, and sets to build content data stores, chat and message queues, and geospatial indexes for demanding, data-intensive web and mobile applications that require low latency and high throughput

ElastiCache, or MemoryDB for Redis

(Depending on your data durability requirements, you can use ElastiCache as a Redis-compatible cache, or MemoryDB to store data in-memory up to hundreds of nodes for ultra-fast performance)

# … I'd like to store large records and don’t want to remodel and break them into smaller ones

## DocumentDB

Record size limit is 400KB in DynamoDB, 16MB in DocumentDB.
Using DynamoDB for large records would require breaking the record into multiple ones, or storing part of the record on S3

## … I want to build player data stores, keep session history and leaderboards for gaming applications that require massive scale, low latency, and high concurrency to make real-time updates

MemoryDB for Redis, or possibly ElastiCache

MemoryDB stores data in-memory across up to hundreds of nodes for ultra-fast performance, while ElastiCache can be added as a cache to your existing database

# … I'd love to reduce the operational overhead of my self-managed MongoDB cluster…

DocumentDB

Migrating to DocumentDB will be simplest since it's compatible with MongoDB APIs

# … I want to stream data changes from the database

DynamoDB or DocumentDB

Both support change data capture to streams

## … I'm looking for a database with cross-region replication which allows to concurrently write and read across multiple regions…

DynamoDB

DocumentDB secondary regions are read-only

# … I have spiky workload with potentially idle periods

DynamoDB

On-Demand capacity mode, or provisioned capacity mode with autoscaling, allow to dynamically adapt the capacity to the traffic and avoid provisioning for peak consumption

# … I have a dataset larger than 64TiB

DynamoDB

DynamoDB supports datasets with multiple petabytes

# … I have a workload with at least hundreds of thousands of writes/sec…

DynamoDB

DynamoDB allows practically unlimited write rates

# … I'm looking for high throughput of strongly consistent reads…

DynamoDB, possibly DocumentDB

DynamoDB throughput is virtually unlimited.
DocumentDB can provide consistent reads only from the primary,
so throughput is limited.

# … I use a relational database for my application but as it grows, queries have gotten slower

## ElastiCache for Redis

You can boost the performance of your applications
by adding an in-memory caching layer to your relational database for
sub-millisecond latency

… I have an application that was initially built for one region that has grown to have users around the world, and I’m seeing performance issues with read requests

## ElastiCache for Redis (with Global Datastore)

ElastiCache offers Global Datastore where data can be replicated across regions with latency of typically under a millisecond per read

# Thank you!

Jason Hunter
jzhunter@amazon.com

Siva Karuturi
maravisk@amazon.com