



Databases on AWS

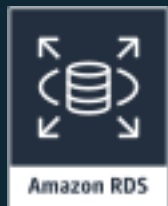
Yasser Quraishi, AWS Solutions Architect – yquraish@amazon.com

Jeff Hosley, AWS Account Manager – jhosley@amazon.com

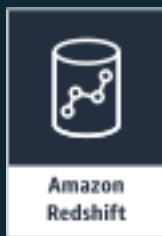
Agenda

- AWS Database Services
- Traditional vs AWS Data services model
- Amazon RDS
- Redshift
- DynamoDB
- ElastiCache
- Neptune

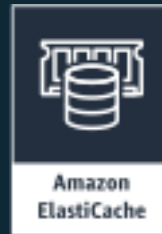
AWS Database Services



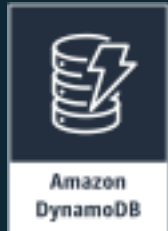
Managed Relational
Database Service



Petabyte-scale
Data Warehouse
Service



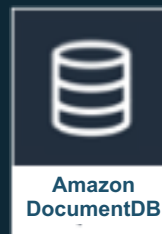
In-Memory key
Value Store
Service



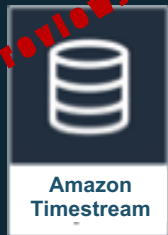
NoSQL Key Value
Oriented Service



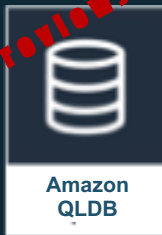
NoSQL Graph
Oriented Service



NoSQL Document
Oriented Service
Mongo Compatible



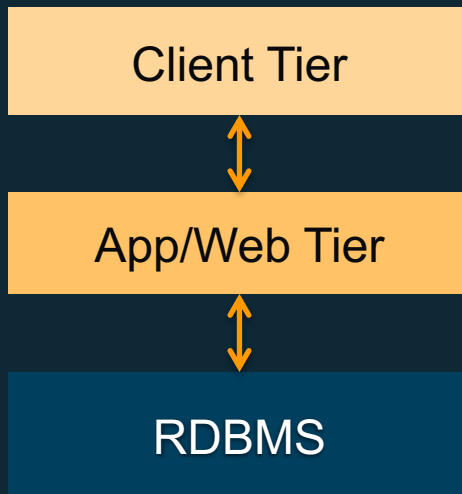
Time Series
Database Service



Quantum Ledger
Database Service

Traditional Database Architecture

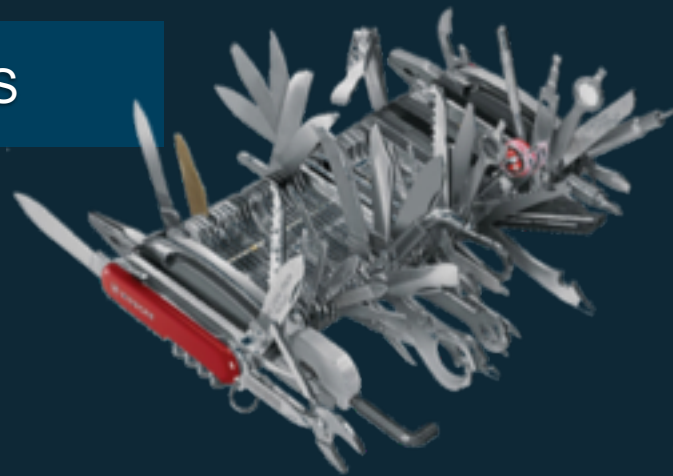
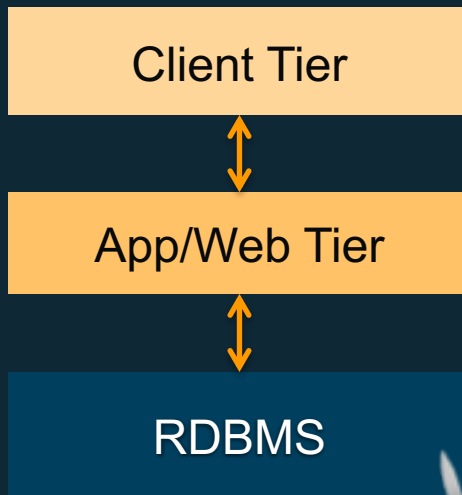
*one database
for all
workloads*



Traditional Database Architecture

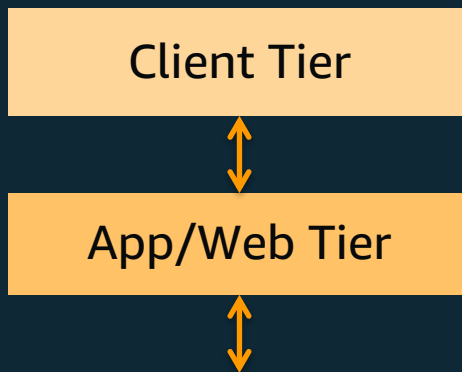
Key-value access
Complex queries
OLAP transactions
Analytics

*All forced into the
relational database*



AWS Data Tier Architecture

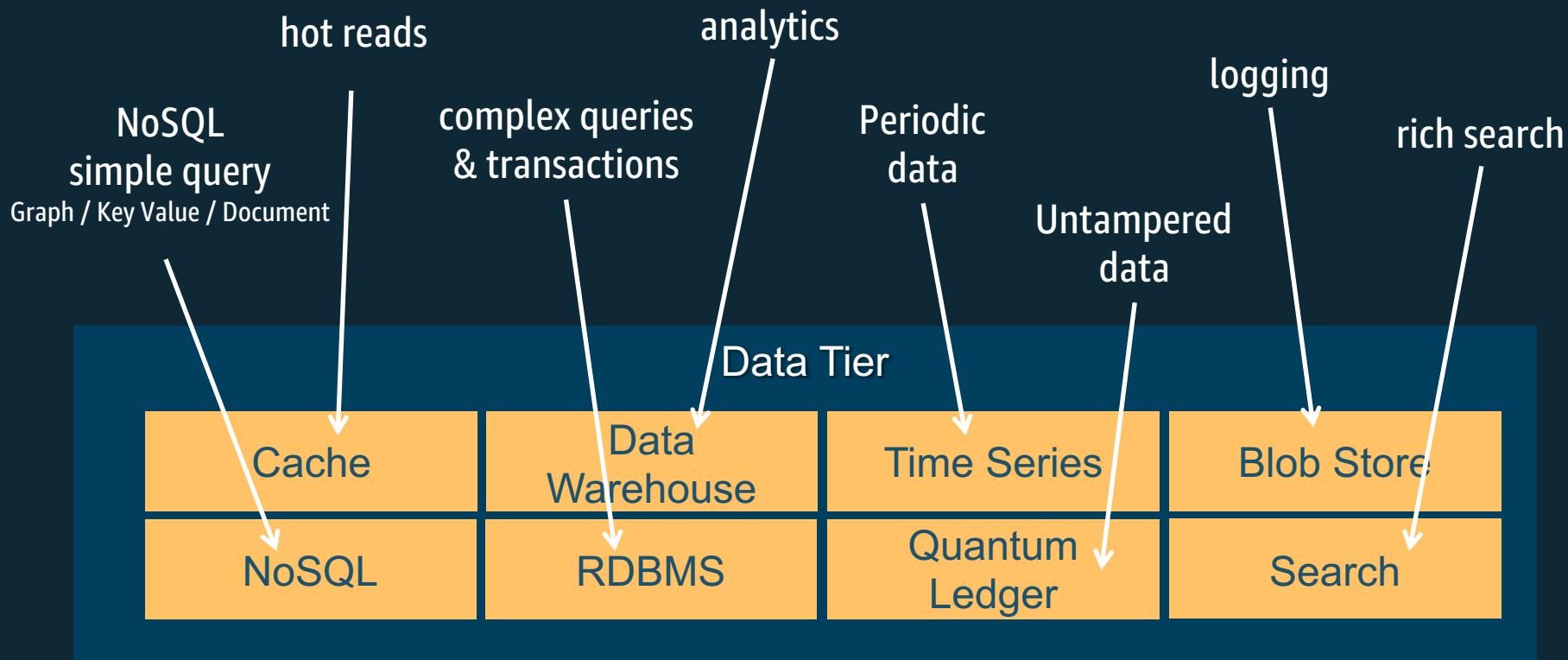
On AWS choose best database service for each workload



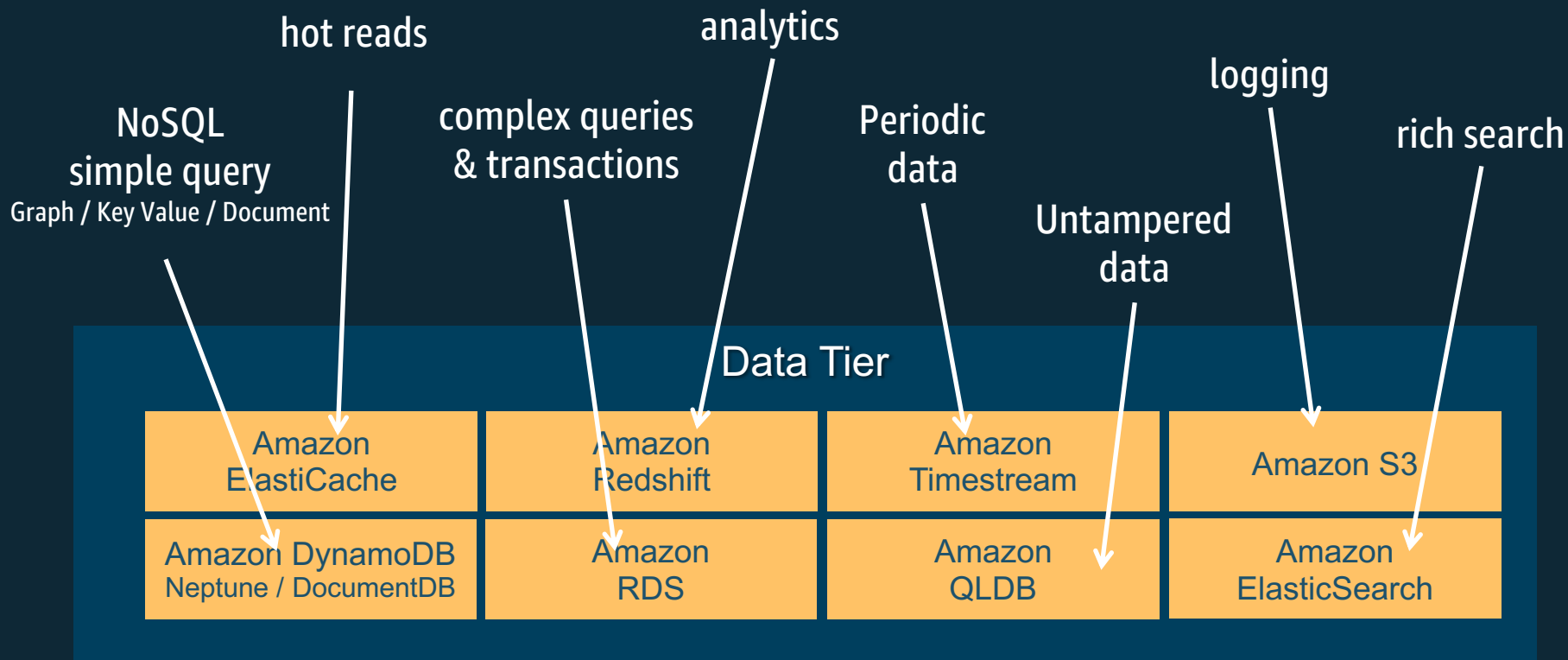
Data Tier

Cache	Data Warehouse	Time Series	Blob Store
NoSQL	RDBMS	Quantum Ledger	Search

Workload Driven Data Store Selection



AWS Database Services for the Data Tier



Relational Databases

Amazon Aurora, MySQL, PostgreSQL, Oracle, SQL Server,
MariaDB

Fully managed



If you host your databases on-premises

App optimization

Scaling

High availability

Database backups

DB s/w patches

DB s/w installs

OS patches

OS installation

Server maintenance

Rack & stack

Power, HVAC, net

you



If you host your databases in Amazon EC2

App optimization

Scaling

High availability

Database backups

DB s/w patches

DB s/w installs

OS patches

OS installation

Server maintenance

Rack & stack

Power, HVAC, net

you

OS installation

Server maintenance

Rack & stack

Power, HVAC, net



If you choose Amazon RDS

App optimization

Scaling

High availability

Database backups

DB s/w patches

DB s/w installs

OS patches

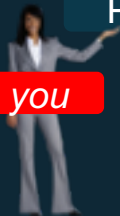
OS installation

Server maintenance

Rack & stack

Power, HVAC, net

you



Scaling

High availability

Database backups

DB s/w patches

DB s/w installs

OS patches

OS installation

Server maintenance

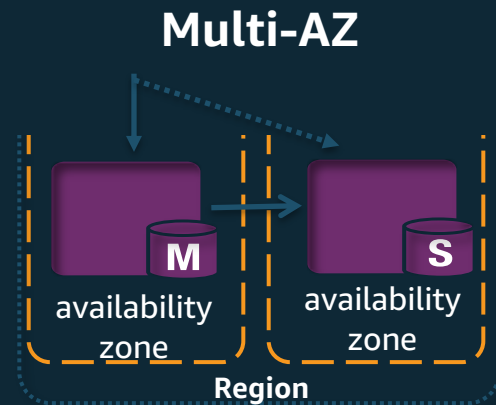
Rack & stack

Power, HVAC, net

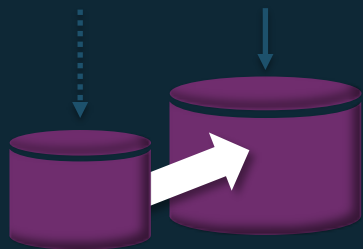


Key Amazon Amazon RDS Features

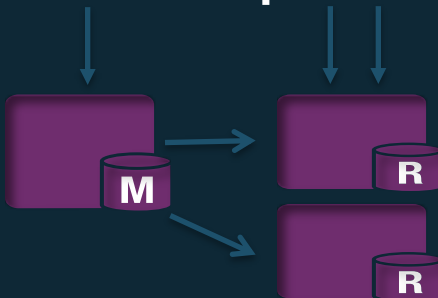
Amazon RDS Configuration	Improve Availability	Increase Throughput	Reduce Latency
Push-Button Scaling		✓	
Multi AZ	✓		
Read Replicas		✓	
Provisioned IOPS		✓	✓



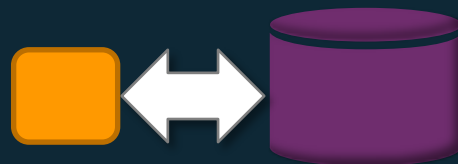
Push-Button Scaling



Read Replicas



Provisioned IOPS



Amazon RDS -- Aurora

Amazon Aurora is a MySQL-compatible relational database engine

- Built from the ground up to leverage AWS
- Speed and availability of high-end commercial databases
- Simplicity and cost-effectiveness of open source databases

Retains compatibility with MySQL 5.6

Up to 5x better performance than MySQL

At a price point 1/10 of a commercial database

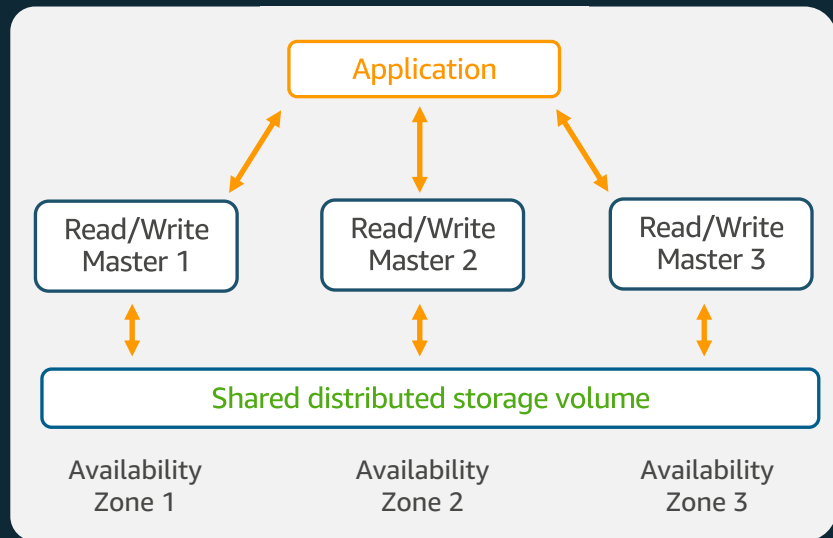
- Data is transparently replicated 6 ways among 3 Availability Zones
- Encryption at rest and in transit
- Add up to 15 Replicas

Compatibility with PostgreSQL

Aurora Multi-Master

First relational DB service with scale-out reads and writes, across multiple datacenters

Scale out both reads **and** writes



Zero application downtime from ANY instance failure

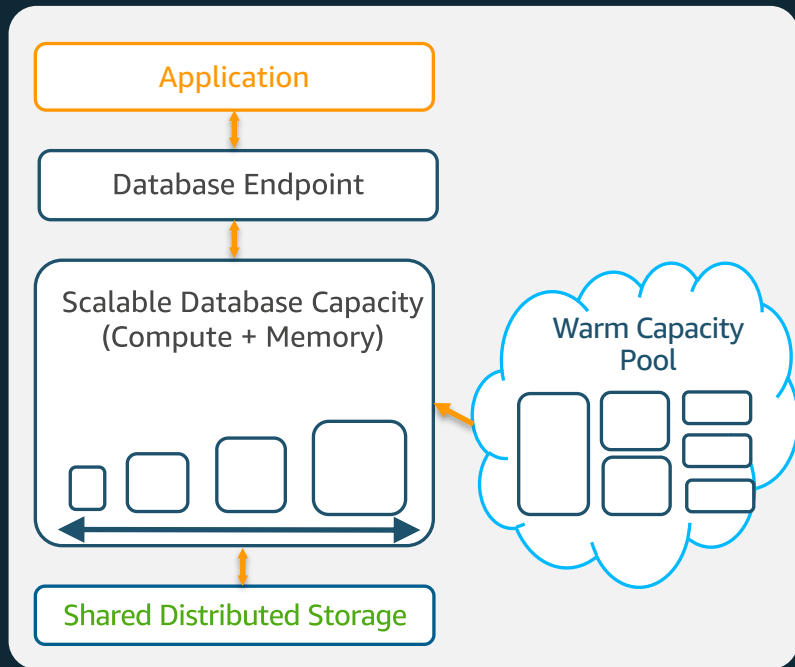
Zero application downtime from ANY AZ failure

Faster write performance and higher scale

Sign up for single-region multi-master preview today;
multi-region multi-master **coming in 2019**

Aurora Serverless

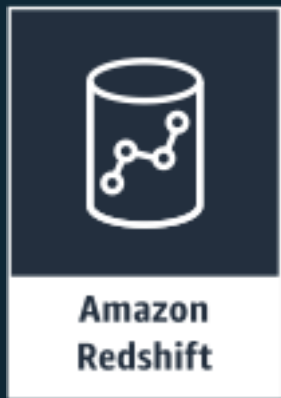
On-demand, auto-scaling database for applications with variable workloads



Starts up on demand, shuts down when not in use

Automatically scales with no instances to manage

Pay per second for the database capacity you use



Petabyte scale

*for as low as
\$934/TB per year*

Massively parallel

Columnar Store

Relational data warehouse

Fully managed = no admin

Amazon Redshift

Fast and powerful, petabyte-scale data warehouse

- Fully managed Relational Database
- Highly-parallel
- Columnar Data Store

Data warehouse-type queries

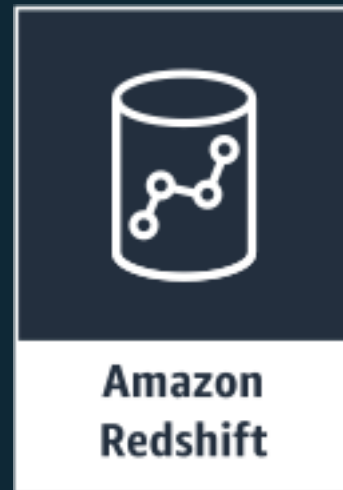
- Aggregations, historical analysis
- BI Tool integration

Grow with your data

- 160 GB → 1.6 PB

SSD and SAS Options

- SSD provides 10-15x perf @ 5.5x the cost per TB/year



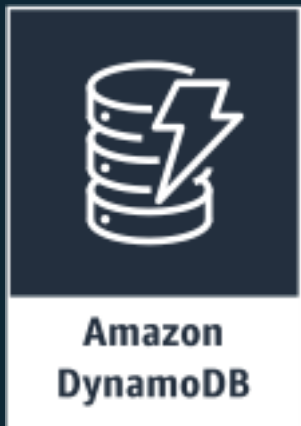
AWS DMS & AWS SCT

AWS Database Migration Service (DMS) easily and securely migrates and/or replicates your databases *and* data warehouses to AWS



AWS Schema Conversion Tool (SCT) converts your commercial database and data warehouse schemas to open-source engines, Amazon Aurora and Redshift. Converts and loads data warehouse data into Amazon Redshift

We have migrated over 103,000 unique databases. And counting...



NoSQL Database

Seamless scalability

Zero admin

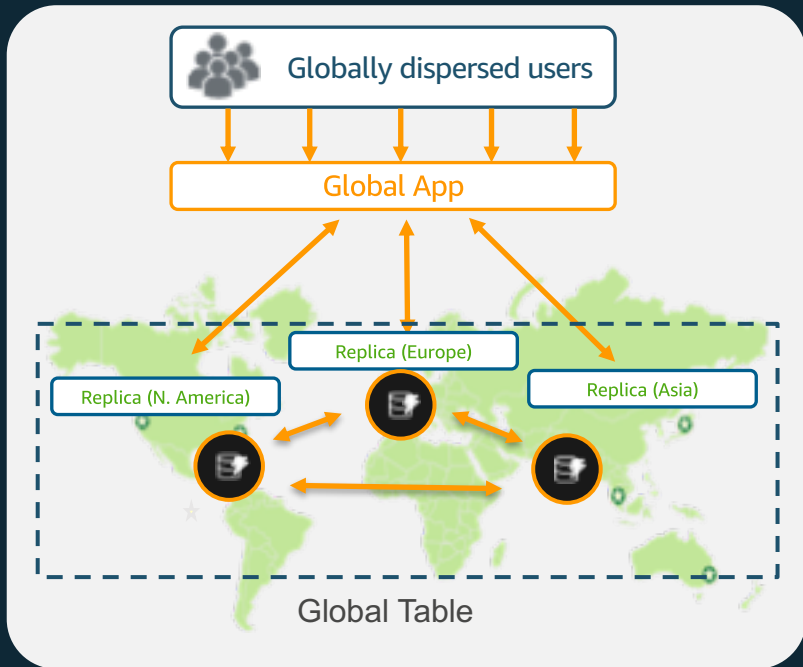
Single digit millisecond latency

Multi-Master

Multi-Region

DynamoDB Global Tables

First fully managed, multi-master, multi-region database



Build high performance, globally distributed applications

Low latency reads & writes to locally available tables

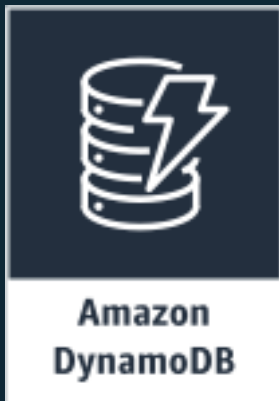
Disaster proof with multi-region redundancy

Easy to setup and no application re-writes required

NoSQL vs. SQL for a new app: how to choose?

Want simplest possible DB management?

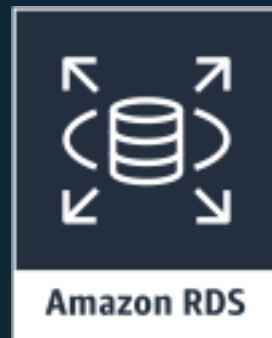
Want app to manage DB integrity?



Need joins, transactions, frequent table scans?

Want DB engine to manage DB integrity?

Team has SQL skills?



Amazon ElastiCache

In-memory cache in the cloud

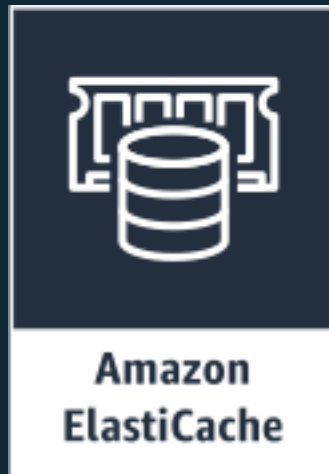
Improve latency and throughput for read-heavy workloads

Supports open-source caching engines

- Memcached
- Redis

Fully managed

Multi-AZ

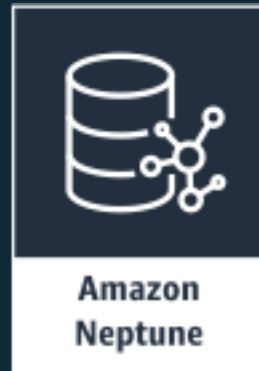


Examples

- Caching of MySQL database query results
- Caching of post-processing results
- Caching of user session & frequently accessed data

Amazon Neptune

Fully managed graph database for highly connected data



Open



Supports Apache
TinkerPop™ & W3C RDF
graph models

Fast & Scalable



Store billions of relationships;
query with millisecond latency

Reliable



6 replicas of your data
across 3 AZs with full
backup and restore

Easy



Build powerful queries
easily with Gremlin and
SPARQL
+
GRAPHQL with AppSync

It's all about choice

Performance-oriented
Cost-oriented

Any Questions?



