

# AWS

# Architecting and SysOps

**Databases in AWS**  
June-July 2019



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

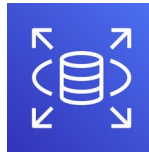
# Contents

Amazon RDS y Aurora  
Amazon DynamoDB  
More NoSQL Databases

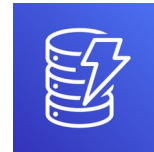
# Amazon RDS y Aurora

Managed relational databases over AWS

# Database



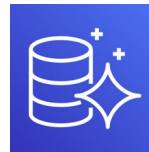
Amazon RDS



Amazon DynamoDB



Amazon Neptune



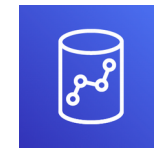
Amazon Aurora



Elasticache



Amazon DocumentDB



Amazon Redshift

# Relational Database Service (RDS)

- RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud
- It provides cost-efficient, resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups
- Amazon RDS is available on several database instance types and provides you with six familiar database engines to choose from
- You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to RDS
- RDS provides high availability and failover support for instances using multi-AZ deployments where a synchronous standby replica of the primary instance is kept in a different AZ



# RDS terms

- **DB Instance**
  - The basic building block of RDS is the DB instance
  - An isolated database environment in the cloud
  - It can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance
- **DB instance class**
  - Determines the computation and memory capacity of an RDS DB instance
  - The DB instance class you need depends on your processing power and memory requirements
  - RDS supports three types of instance classes: Standard, Memory Optimized, and Burstable Performance
- **DB instance storage**
  - RDS uses EBS volumes for database and log storage
  - Depending on the amount of storage requested, RDS automatically stripes across multiple EBS volumes to enhance performance

# RDS terms

- DB Engine

- Each DB instance runs a DB engine
- RDS currently supports MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines
- Each DB engine has its own supported features, and each version of a DB engine may include specific features
- Additionally, each DB engine has a set of parameters in a DB parameter group that control the behavior of the databases that it manages

- DB Instance Status

- Indicates the health of the DB instance
- Depending on the status, billing changes
- The status can be available, backing-up, failed, stopped, and so on

# RDS automated backups

- RDS creates and saves automated backups of your DB instance
  - Creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases
- Automated backups are performed during the preferred backup window of your DB instance:
  - ✓ According to the backup retention period that you specify. By default, is 7 days
  - ✓ If necessary, you can recover your database to any point in time during the backup retention period
  - ✓ Your DB instance must be in the ACTIVE state
  - ✓ The 30-minute backup window is selected at random from an 8-hour block of time for each AWS Region unless a specific time is specified
- You can also back up your DB instance manually, by manually creating a DB snapshot



# RDS Snapshots

- RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases
- The first snapshot of a DB instance contains the data for the full DB instance. Subsequent snapshots of the same DB instance are incremental, which means that only the data that has changed after your most recent snapshot is saved.
- Creating this DB snapshot on a Single-AZ DB instance results in a brief I/O suspension that can last from a few seconds to a few minutes, depending on the size and class of your DB instance. Multi-AZ DB instances are not affected by this I/O suspension since the backup is taken on the standby
- The amount of time it takes to create a snapshot varies with the size your databases. Since the snapshot includes the entire storage volume, the size of files, such as temporary files, also affects the amount of time it takes to create the snapshot

# Aurora

- A fully managed relational database engine that's compatible with MySQL and PostgreSQL
- It is one of the DB engines available for Amazon RDS web service
- The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora
- With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL without requiring changes to most of your existing applications
- Aurora includes a high-performance storage subsystem. The underlying storage grows automatically as needed, up to 64 terabytes
- Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration



# Features

- Provisioning, patching, backup, recovery, and other tasks is common to other RDS databases
- Offers 10 more read replicas with a maximum of 15 read replicas
  - In the event of the primary database failing, any of the read replicas can become the primary database, with no data loss
- Stores data in 10 GB chunks and scales automatically up to 64 TB
  - Each chunk of data is replicated six-ways across 3 AZs and encrypted using a KMS key.
- 5X the throughput of standard MySQL and 3X the throughput of standard PostgreSQL
  - Performance is on par with commercial databases, at 1/10th the cost
- Cost is 20% and 16% higher compared to similar sized RDS MySQL and PostgreSQL standard engines
  - Aurora Serverless can reduce that difference

# Aurora pricing comparison

- On-Demand MySQL-Compatible edition

Standard Instances	Aurora	MySQL
db.t2.micro	---	\$0.017
db.t3.small	\$0.041	\$0.038
db.t3.medium	\$0.082	\$0.068
db.t2.small	\$0.041	\$0.034
db.t2.medium	\$0.082	\$0.068

- On-Demand PostgreSQL-Compatible edition

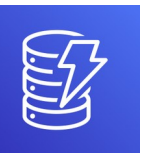
Memory Instances	Aurora	PostgreSQL
db.t2.micro	---	\$0.018
db.r4.large / db.r5.large	\$0.029	\$0.025
db.r4.xlarge / db.r5.xlarge	\$0.058	\$0.050

# Amazon DynamoDB

Traditional NoSQL solution as a service

# DynamoDB

- A NoSQL database that supports key-value and document data models and delivers single-digit millisecond performance at any scale
- Enables developers to build modern, serverless applications that can start small and scale globally to support petabytes of data and tens of millions of read and write requests per second
- A fully managed, multi-region database with built-in security, backup and restore, and in-memory caching for internet-scale applications
- Designed to run high-performance, internet-scale applications that would overburden traditional relational databases
  - DynamoDB can handle more than 10 trillion requests per day and support peaks of more than 20 million requests per second



# Performance at scale

- Support for both key-value and document data models that allows to have a flexible schema
  - Each row can have any number of columns at any point in time, so you can adapt the tables as your business requirements change, without having to redefine the table schema
- Microsecond latency with DynamoDB Accelerator
  - DynamoDB Accelerator (DAX) is an in-memory cache that delivers an improved read performance of your tables by up to 10 times—taking the time required for reads from milliseconds to microseconds, even at millions of requests per second
- Automated global replication with global tables across AWS regions
  - It automatically scales capacity to accommodate your workloads
  - Your globally distributed applications can access data locally in the selected regions to get single-digit millisecond read and write performance
- Real-time data processing with DynamoDB streams
  - A capture of item-level modifications in a time-ordered sequence logged for 24h

# Serverless

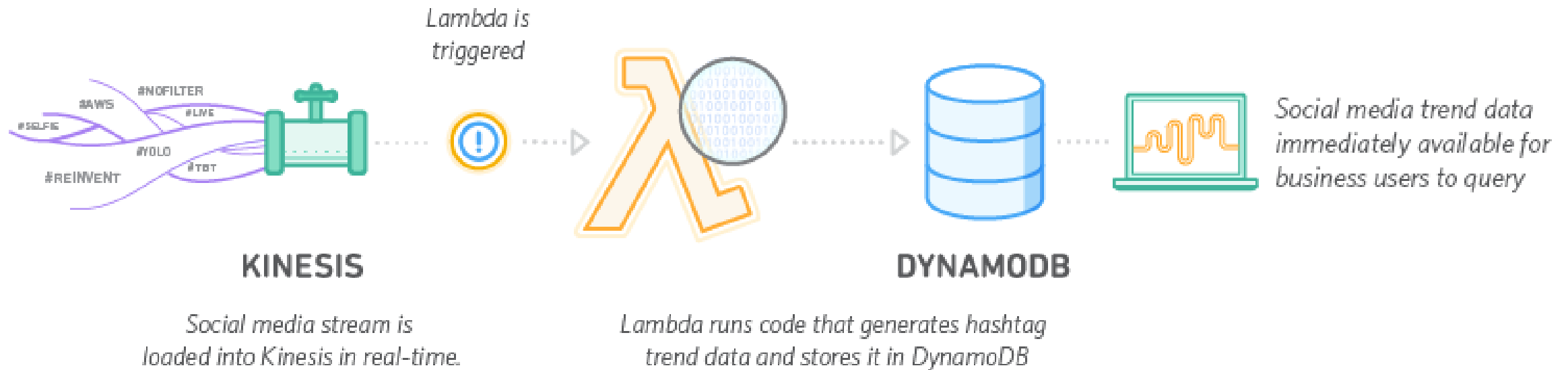
- Read/write capacity modes for each table: on-demand and provisioned
- On-demand capacity mode: for workload less predictable
  - Workloads are instantly accommodated as they ramp up or down to any previously reached traffic level. If a workload's traffic level hits a new peak, DynamoDB adapts rapidly to accommodate the workload and manages capacity for you
- Provisioned capacity mode: when you are confident about the utilization
  - First, you set read and write capacity, and then DynamoDB delivers automatic scaling up and down of throughput and storage based on that capacity by monitoring the performance usage of your application
- Change tracking with triggers
  - DynamoDB integrates with Lambda to provide triggers. Using triggers, you can automatically execute a custom function when item-level changes in a DynamoDB table are detected
  - With triggers, you can build applications that react to data modifications in your tables. The Lambda function can perform any actions you specify



# Use case: microservices data store

- Build flexible and reusable microservices using DynamoDB as a serverless data store for consistent and fast performance

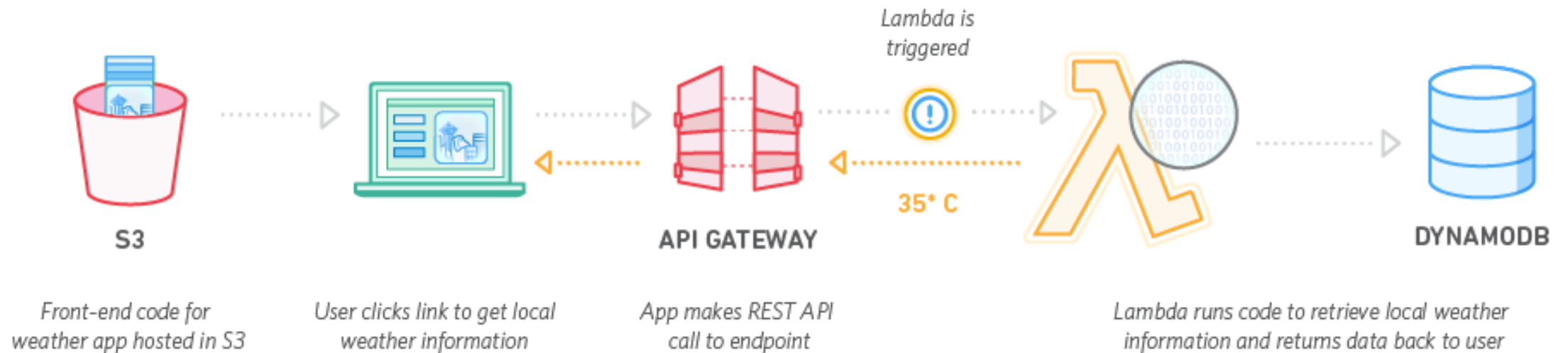
## Example: Analysis of Streaming Social Media Data



# Use case: serverless web applications

- Build powerful web applications that automatically scale up and down. You don't need to maintain servers, and your applications have automated high availability.

Example: Weather Application

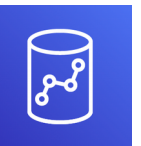


# More NoSQL Databases

More NoSQL solutions available on AWS

# Redshift

- Redshift powers mission critical analytical workloads for companies of all size around the world
- Data lakes are the future and Amazon Redshift allows you to query data in your data lake without moving it or transforming it into a set schema
- You can run analytic queries against petabytes of data stored locally in Redshift, and directly against exabytes of data stored in Amazon S3
- Redshift is natively integrated with the AWS analytics ecosystem.
  - AWS Glue can extract, transform, and load (ETL) data into Redshift
  - Amazon Kinesis Data Firehose is the easiest way to capture, transform, and load streaming data into Redshift for near real-time analytics.
  - Amazon QuickSight to create reports, visualizations, and dashboards
- More than 15k customers use Amazon Redshift



# Redshift



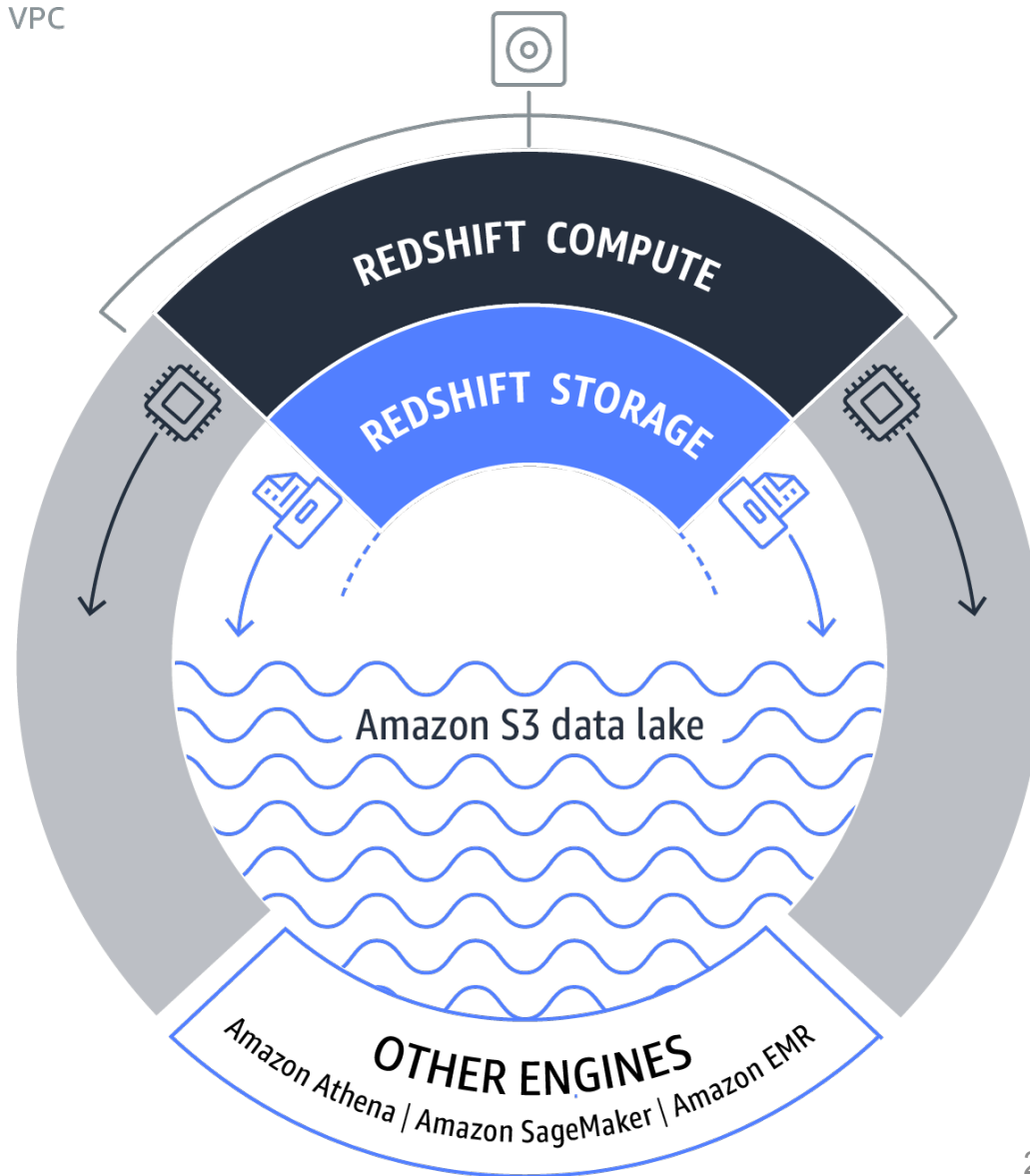
Leader node routes and speeds up queries using machine learning and results caching



Compute scales to handle high concurrency and queries to the data lake



Storage scales to exabytes of data in the Amazon S3 data lake in many open formats, including Parquet and ORC



# Features

- **Faster performance**
  - Redshift uses columnar storage, data compression, and zone maps to reduce the amount of I/O needed to perform queries
  - It uses machine learning to deliver high throughput, irrespective of your workloads or concurrent usage
  - Redshift uses result caching to deliver sub-second response times for repeat queries
- **Easy to setup, deploy, and manage**
  - You can deploy a new data warehouse with just a few clicks in the AWS console
  - Redshift automatically and continuously backs up your data to Amazon S3
  - Fault tolerant: has multiple features that enhance the reliability of your data warehouse cluster
  - It gives you the flexibility to execute queries within the console or connect SQL client tools, libraries, or Business Intelligence tools you want
- **Cost-effective: no upfront costs, pay as you go**
- **Scale quickly to meet your needs**
  - Petabyte-scale data warehousing
  - Provides consistently fast performance, even with thousands of concurrent queries

# Elasticache

- Almost any large-scale web application in the world runs fully or partially beside some kind of in-memory storage for performance purpose
- ElastiCache offers you the ability to deploy, run, and scale popular open source compatible in-memory data stores
- It encourages you to build data-intensive apps or improve the performance of your existing apps by retrieving data from high throughput and low latency in-memory data stores
- Popular use cases for Amazon ElastiCache are Gaming, Ad-Tech, Financial Services, Healthcare, and IoT apps
- ElastiCache comes in two flavours: Redis and Memcached



# Features of ElastiCache for Redis

- **Manageability**
  - Event Notifications via email or SMS
  - Monitoring and metrics
  - Fully Managed Redis, automating time-consuming management tasks
  - Easy Engine Upgrade to latest available engine version
- **Availability and Reliability**
  - Multi A-Z for the Redis Cluster with automatic failover functionality
  - Instance Monitoring and Repair, maintaining the nodes healthy
  - Backup, Restore, and Export of RDB files
- **Scalability**
  - For non-cluster mode, you can scale up to a larger node type, and go up to a maximum of 635.61 GiB
  - Supporting up to 250 nodes and shards, you can scale up to 155.17 TiB (170.6 TB) of in-memory data with 48.6 million reads and 9.7 million writes per second



# Features of ElastiCache for Memcached

- **Extreme Performance**
  - Works as an in-memory data store and cache to support the most demanding applications requiring sub-millisecond response times
- **Secure and Hardened**
  - Enables you to isolate your cluster to the IP ranges you choose for your nodes, and use them to connect to your application.
  - Continuously monitors your nodes and applies the necessary patches to keep your environment secure
- **Auto Discovery**
  - ElastiCache for Memcached Cluster Client with Auto Discovery saves you time by simplifying the way an application connects to a Memcached cluster
  - The client connects to a single endpoint and gets configuration information about other nodes
  - When nodes are added or removed, the client reconfigures itself to use the most current set of nodes
- **Easily Scalable**
  - Includes sharding to scale in-memory cache with up to 20 nodes and 12.7 TiB per cluster

# DocumentDB

- Amazon DocumentDB (with MongoDB compatibility) is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads
- It is designed from the ground-up to give you the performance, scalability, and availability you need when operating mission-critical MongoDB workloads at scale
- Offers easy interoperability with the Amazon DocumentDB console, CLI, or SDK
- Implements the Apache 2.0 open source MongoDB 3.6 API by emulating the responses that a MongoDB client expects from a MongoDB server, allowing you to use your existing MongoDB drivers and tools
- Decouples the storage and compute, allowing each to scale independently, and you can increase the read capacity to millions of requests per second



# Features

- **MongoDB-compatible**
  - Amazon DocumentDB is compatible with MongoDB 3.6 drivers and tools
  - A vast majority of the applications, drivers, and tools that customers already use can be used with Amazon DocumentDB with little or no change
  - Customers can easily migrate their MongoDB databases for free (6 months per instance) with virtually no downtime using the AWS Database Migration Service (DMS)
- **Fully Managed**
  - Automatic Provisioning and setup
  - Monitoring and Metrics for compute, memory, storage, query throughput, and active connections
  - Automatic Software Patching: you can control if and when your cluster is patched
- **Performance at scale**
- **Highly Secure and Compliant**
- **Highly Available**
  - Instance Monitoring and Repair
  - Multi-AZ Deployments with up to 15 Read Replicas

# Neptune

- A graph database is a store of vertices (nodes) and edges (relationships or connections) which can both have properties stored as key-value pairs
- Neptune is a fast and reliable graph database service that makes it easy to gain insights from relationships among your highly connected datasets
- It supports popular graph models Property Graph and W3C's RDF, and their respective query languages Apache TinkerPop Gremlin and SPARQL
- Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security, allowing you to easily build queries that efficiently navigate between nodes and relationships



# Features

- **High Performance and Scalability**
  - High Throughput, Low Latency for Graph Queries
  - Easy Scaling of Database Compute Resources
  - Storage that Automatically Scales up to a maximum of 64 TB
  - Low Latency Read Replicas to support high volume application requests by creating up to 15 database read replicas
  - You can scale query throughput to 100s of thousands of queries per second
- **High Availability and Durability**
  - Instance Monitoring and Repair
  - Multi-AZ Deployments with Read Replicas for high availability
  - Fault-tolerant and Self-healing Storage, the service supports fast-failover
  - Automatic, Continuous, Incremental Backups and Point-in-time Restore
  - Database Snapshots
  - Is designed to offer greater than 99.99% availability
- **Open Graph APIs**

# Features

- **Fully Managed**
  - Easy to Use: database instances are pre-configured with parameters and settings
  - Easy to Operate: you do not need to create custom indexes over your graph data
  - Monitoring and Metrics to view over 20 key operational metrics for your database instances
  - Automatic Software Patching
  - Database Event Notifications via email or SMS
  - Fast Database Cloning
- **Fast Parallel Bulk Data Loading**
  - Property Graph Bulk Loading for data that is stored in S3
  - RDF Bulk Loading: the N-Triples (NT), N-Quads (NQ), RDF/XML, and Turtle RDF 1.1 serializations are supported
- **Network Isolation and Resource-Level Permissions**
- **Cost-effectiveness:** you simply pay an hourly charge for each instance that you launch

# Database documentation

- Amazon RDS documentation
  - <https://docs.aws.amazon.com/rds/>
- Amazon DynamoDB documentation
  - <https://docs.aws.amazon.com/dynamodb/>
- Amazon Aurora documentation
  - <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/>
- ElastiCache documentation
  - <https://docs.aws.amazon.com/elasticache/index.html>
- Amazon DocumentDB documentation
  - [\\_x0001\\_https://docs.aws.amazon.com/documentdb/](https://docs.aws.amazon.com/documentdb/)
- Amazon Neptune documentation
  - <https://docs.aws.amazon.com/neptune/>
- Amazon Redshift documentation
  - <https://docs.aws.amazon.com/redshift/>