Designing Highly Available and Scalable Databases on AWS

Danny Jessee

AWS CERTIFIED DEVOPS ENGINEER - PROFESSIONAL

@dannyjessee

Overview

Amazon Relational Database Service (RDS)

Supported database engines

RDS Multi-AZ deployments

Vertical and horizontal scaling

Amazon DynamoDB

3-tier Architecture

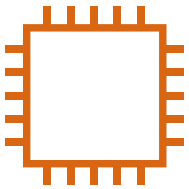
Front-end
web server

Application
server

Database
server

Requires ongoing patching and maintenance

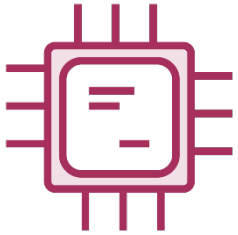
Requires DBA labor to configure replication and failover



EC2 instance

Amazon Relational Database Service (RDS)

Fully-managed



Leverages EC2 instances and EBS volumes

AWS handles patching and maintenance

Automated backups for point-in-time recovery

User-initiated snapshots

Multi-AZ deployments

Running Your Database on EC2

Shell access to server file system or other server-side dependencies

Willing to expend DBA labor in exchange for lower usage costs

Additional work required to support maintenance and failover scenarios

Hands-on lab: Build your own highly available database servers on EC2

Supported RDS Database Engines

MySQL, MariaDB, PostgreSQL

Oracle and Microsoft SQL Server

“Easy create” options

Oracle supports “bring your own license” and “license included” pricing

Microsoft SQL Server “license included”

Amazon Aurora

MySQL and PostgreSQL-compatible

5x faster than MySQL for 1/10 the cost

Up to 64 TB per database instance

Data written 6x across 3 availability zones

Up to 15 read replicas across 3 availability zones

Global databases can span multiple AWS regions

RDS Multi-AZ Deployments

Highly available by spanning two availability zones

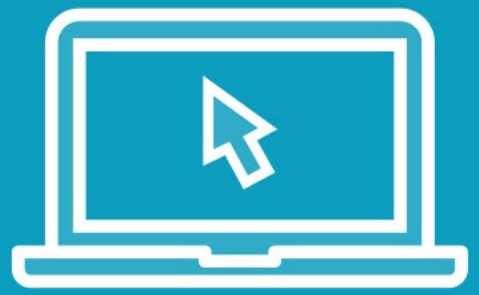
Can convert a non-Multi-AZ deployment

RDS maintains a synchronous standby replica

Automatic failover in the event of an outage or maintenance

Expose a single DNS endpoint

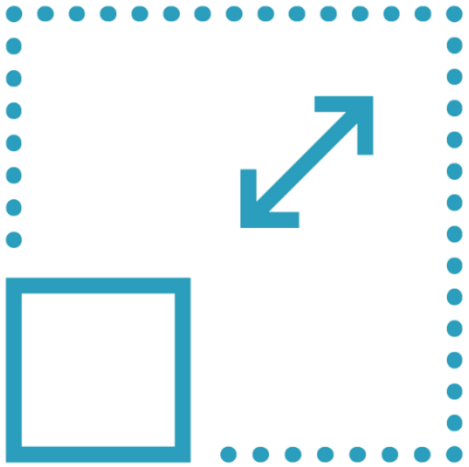
Demo



Create a new RDS Multi-AZ deployment

Failover scenarios

Scaling with RDS



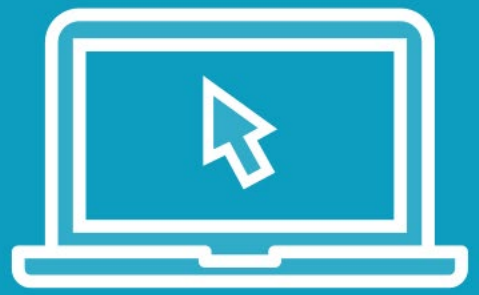
Scale up
by changing
instance class

Add storage
or provisioned
IOPS

Scale out
by adding
read replicas

Handle increased
demand for
read operations

Demo



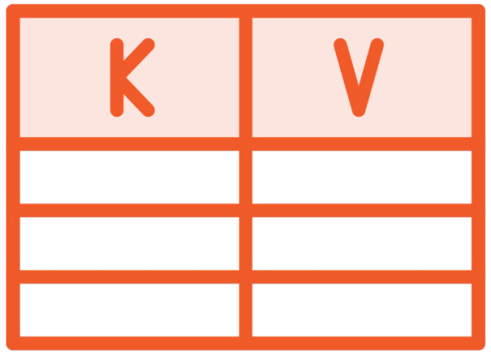
Change instance type and storage

Add an asynchronous read replica

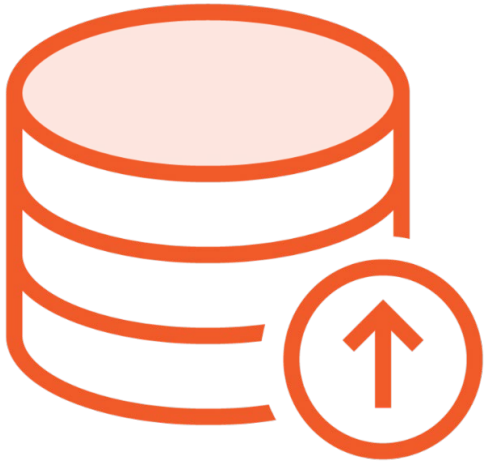
Introduction to DynamoDB

DynamoDB Overview

Serverless



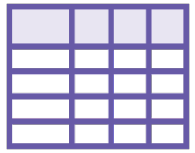
Schemaless (NoSQL)



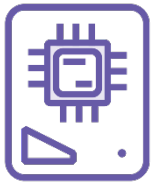
Not easy to convert to/from relational

Continuous or on-demand backups

DynamoDB Tables



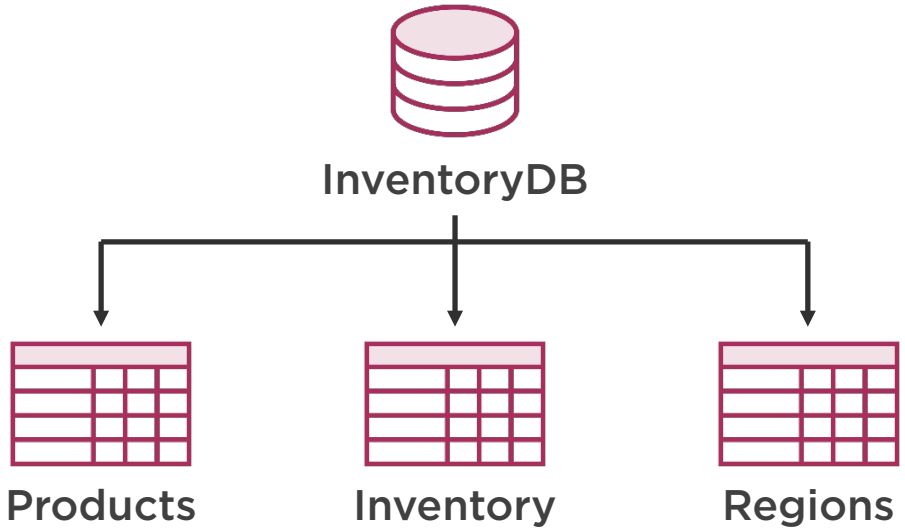
No database, table is highest-level entity

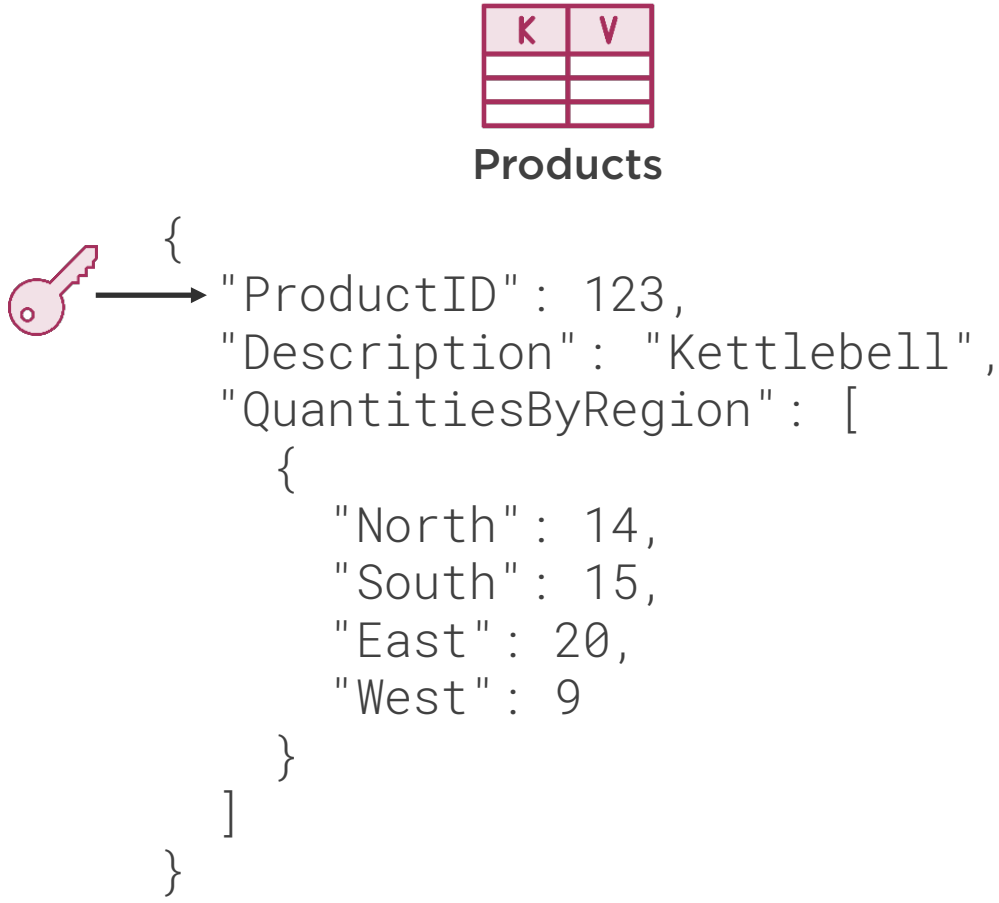


Essentially unlimited storage and scalability

Data automatically replicated across multiple availability zones

Relational Database vs. DynamoDB





DynamoDB Secondary Indexes

Local Secondary Indexes (LSI)

Only created when table is created

Same partition key, different sort key

Local to a partition

Can return non-projected attributes

Consume provisioned RCU/WCU from parent table

Global Secondary Indexes (GSI)

Can be created after table is created

Different partition and sort keys

Global to a table

Can not return non-projected attributes

Separately provisioned RCU/WCU

DynamoDB Provisioned Capacity

Read/Write capacity units (RCU/WCU)

Throttling when you exceed provisioned capacity

Can use auto scaling

Estimate average size of data read and write requests

Estimate number of reads and writes per second

Cost is based on provisioned RCU and WCU

DynamoDB Provisioned Capacity

Read Capacity Units (RCU)

For items up to 4 KB in size:

1 RCU = 2 eventually consistent read requests per second

1 RCU = 1 strongly consistent read request per second

2 RCU = 1 transactional read request per second

DynamoDB Provisioned Capacity

Write Capacity Units (WCU)

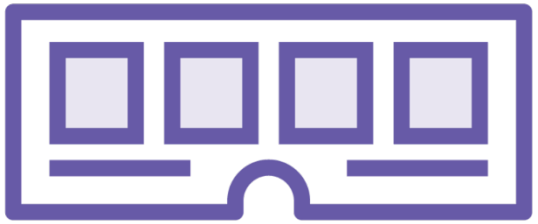
For items up to 1 KB in size:

1 WCU = 1 standard write request per second

2 WCU = 1 transactional write request per second

Amazon DynamoDB Accelerator (DAX)

Caching layer to prevent duplicate read requests



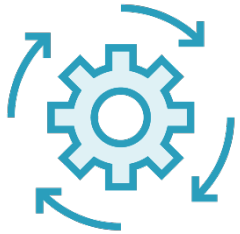
Fully managed, in-memory cache

Provision a DAX cluster in front of your DynamoDB table

DynamoDB Streams and Global Tables

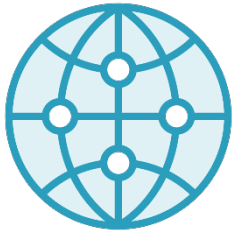
DynamoDB writes data across AZs by default

Streams can allow data to be written across regions

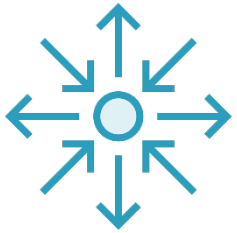


Streams capture all activity within a table

Can include before and after images of items

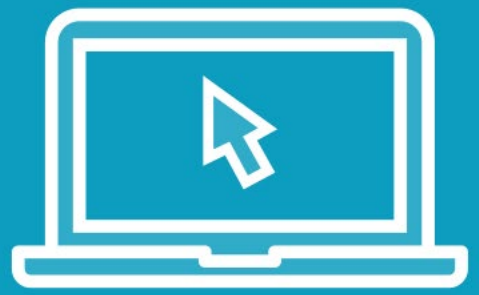


Global Tables automate replication across regions



Global, fault-tolerant applications

Demo



Create a new DynamoDB table

Partition and sort keys

Secondary indexes

Provision read and write capacity

DynamoDB Global Tables

Review

- Amazon RDS
- Amazon Aurora
- RDS Multi-AZ deployments
- Read replica instances
- Amazon DynamoDB
- DynamoDB Streams and Global Tables

Up Next:
Disaster Recovery Strategies on AWS