

## A tale of two databases with DynamoDB and RDS:

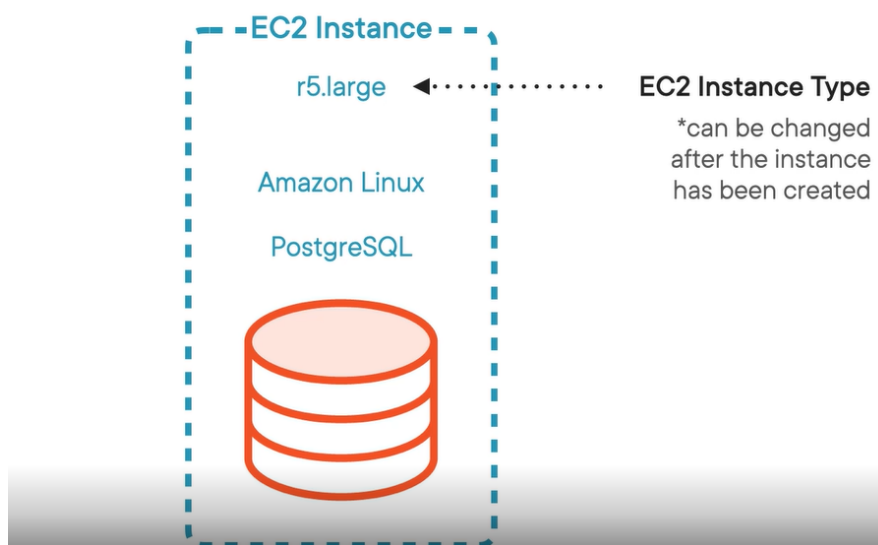
Relational database service:

- Managed database instances in AWS running on EC2.

RDS managed task examples:

- Software upgrades
- Nightly database backups
- Monitoring

## RDS Instance Architecture



RDS backups:

- Occurs daily
- Configurable backup window
- Backup stored 1 to 35 days
- Restore database from backup

Multi-AZ deployment:

- Database replication to different availability zone
- Automatic failover in case of catastrophic event

Database read replica:

- Non-production copy of database
- Eventual consistency with source
- Useful for running queries on data
- Will not be used as failover

RDS database options:

- Amazon aurora
- PostgreSQL
- Maria DB
- MySQL
- SQL server
- Oracle

Creating a database in RDS:

The screenshot shows the AWS RDS 'Create database' console. The 'Engine options' section is expanded, showing three engine types: Amazon Aurora, MySQL, and MariaDB. PostgreSQL is selected as the engine type. The 'Version' dropdown is set to 'PostgreSQL 12.7-R1'. The 'Templates' section shows three options: Production, Dev/Test, and Free tier. The 'Free tier' option is selected. The 'Availability and durability' section is partially visible at the bottom.

**Create database**

**Choose a database creation method** [Info](#)

☒ **Standard create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

☐ **Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

Engine type [Info](#)

☒ Amazon Aurora

☐ MySQL

☐ MariaDB

**Version**

PostgreSQL 12.7-R1

**Templates**  
Choose a sample template to meet your use case.

☐ **Production**  
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**  
This instance is intended for development use outside of a production environment.

☒ **Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

**Availability and durability**



DynamoDB Overview:

## Database Services in AWS

**RDS**

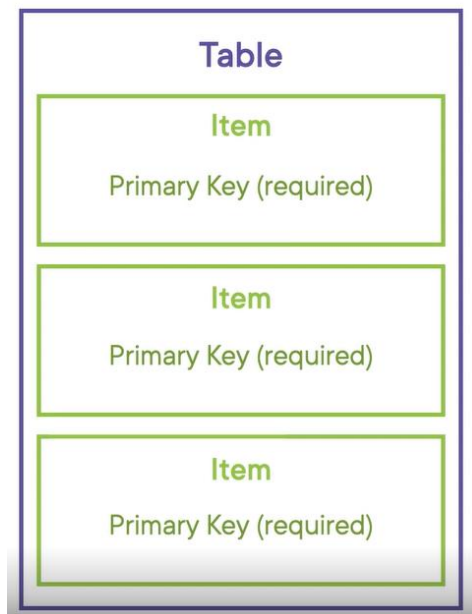
Relational

SQL

**DynamoDB**

Non-relational

NoSQL



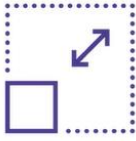
Provisioned Throughput capacity:

- Read/write operations per second provisioned for your DynamoDB table.
- DynamoDB will throttle or deny requests that exceed the table's provisioned throughput capacity.

DynamoDB capacity:

1. DynamoDB on demand capacity:

# DynamoDB On-demand Capacity



**Table capacity scales as needed**



**Pay per table or index request**



**More expensive than Provisioned Capacity mode**

2. DynamoDB provisioned capacity Auto scaling:

## DynamoDB Provisioned Capacity Auto Scaling



**Increases and decreases Provisioned Capacity based on rules**



**Works like EC2 Auto Scaling groups**



**Cheaper than On-demand Capacity mode**

Deciding between RDS and DynamoDB:

RDS:

- Efficient data transfer and storage
- Strict record schema
- Easy querying with SQL
- Strong query flexibility

## DynamoDB:

- No schema, only primary key restriction
- Strong storage flexibility
- Limited query properties

## Demo: create DynamoDB table:

The screenshot shows the 'Create table' wizard in the AWS Management Console. The 'Table details' section is active, showing the table name 'toppings' and the partition key 'id' with a 'String' data type. The wizard explains that DynamoDB is a schemaless database and provides instructions for naming and keying the table.

**Table details** [Info](#)  
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.  
  
Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.).

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.  
   
1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the

**Feedback** Looking for language selection? Find it in the new Unified Settings [↗](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Tables' page in the AWS Management Console. A green banner at the top indicates that the 'toppings' table was created successfully. The table is listed in the 'Tables (1)' section with the following details:

	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
<input type="checkbox"/>	toppings	Active	id (S)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)

**DynamoDB** [×](#) **The toppings table was created successfully.** [×](#) [i](#)

**Tables (1)** [Info](#) [Refresh](#) [Actions](#) [Delete](#) [Create table](#)

[<](#) [1](#) [>](#) [Settings](#)

**Feedback** Looking for language selection? Find it in the new Unified Settings [↗](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#create-table

aws Services Search for services, features, blogs, docs, and more [Alt+S]

DynamoDB > Tables > Create table

## Create table

**Table details** [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.

users

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.).

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

username String

1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

us-east-2.console.aws.amazon.com/dynamodbv2/home?region=us-east-2#tables

aws Services Search for services, features, blogs, docs, and more [Alt+S]

DynamoDB

Dashboard

**Tables**

Update settings

Explore items

PartiQL editor [New](#)

Backups

Exports to S3

Reserved capacity

Settings [New](#)

▼ DAX

Clusters

Subnet groups

Parameter groups

Events

**The users table was created successfully.**

DynamoDB > Tables

**Tables (2)** [Info](#)

Find tables by table name Any table tag

	Name ▲	Status	Partition key	Sort key	Indexes	Read capacity mode	Write ca
<input type="checkbox"/>	toppings	Active	id (S)	-	0	Provisioned with auto scaling (5)	Provision
<input type="checkbox"/>	users	Active	username (S)	-	0	Provisioned with auto scaling (5)	Provision

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences