

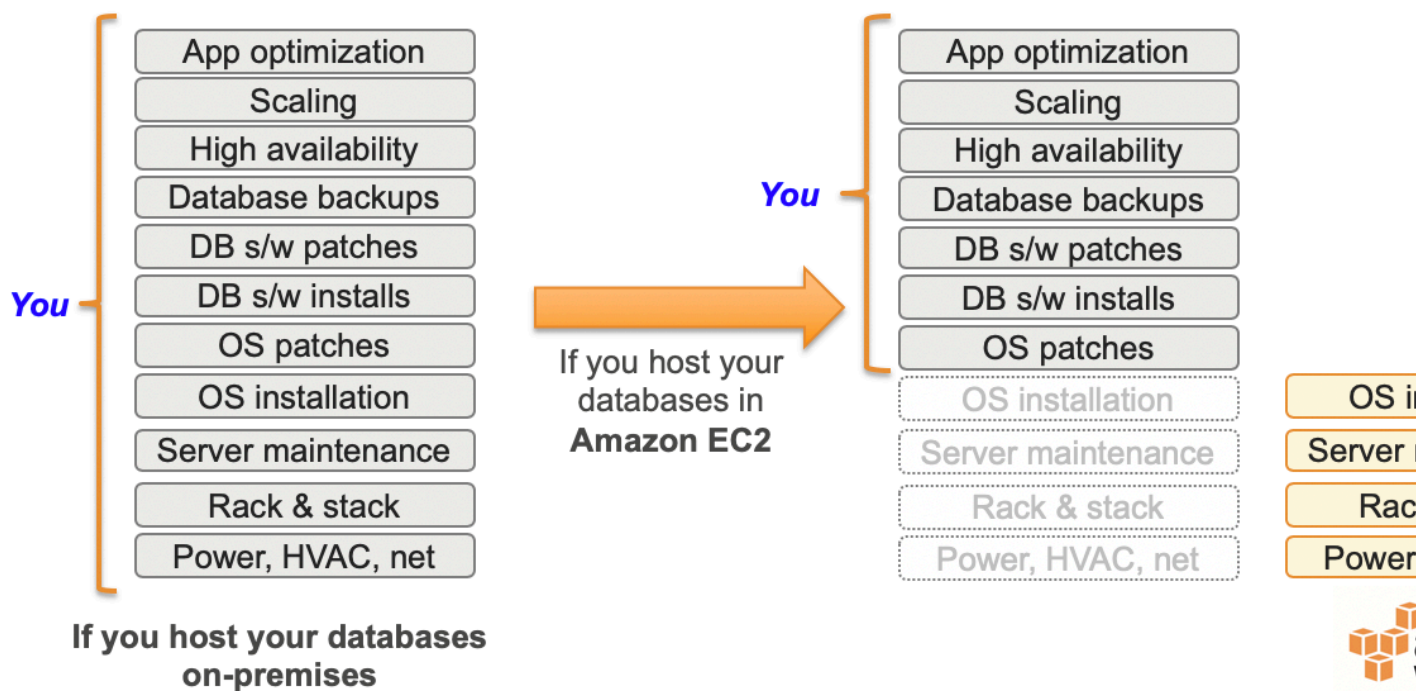
5 Databases

Sunday, 15 September 2019

2:53 PM

RDS

- **Managed Service**
 - DB Setup, HA Replication, Automated Backups, Patching, Scaling.
- **Resizable Capacity**
 - Compute, Database Storage, Storage IO.
- Deploy **MySQL, MariaDB, Microsoft SQL Server, Oracle, and PostgreSQL** databases
- **Amazon Aurora – MySQL, PostgreSQL**



kup,

S

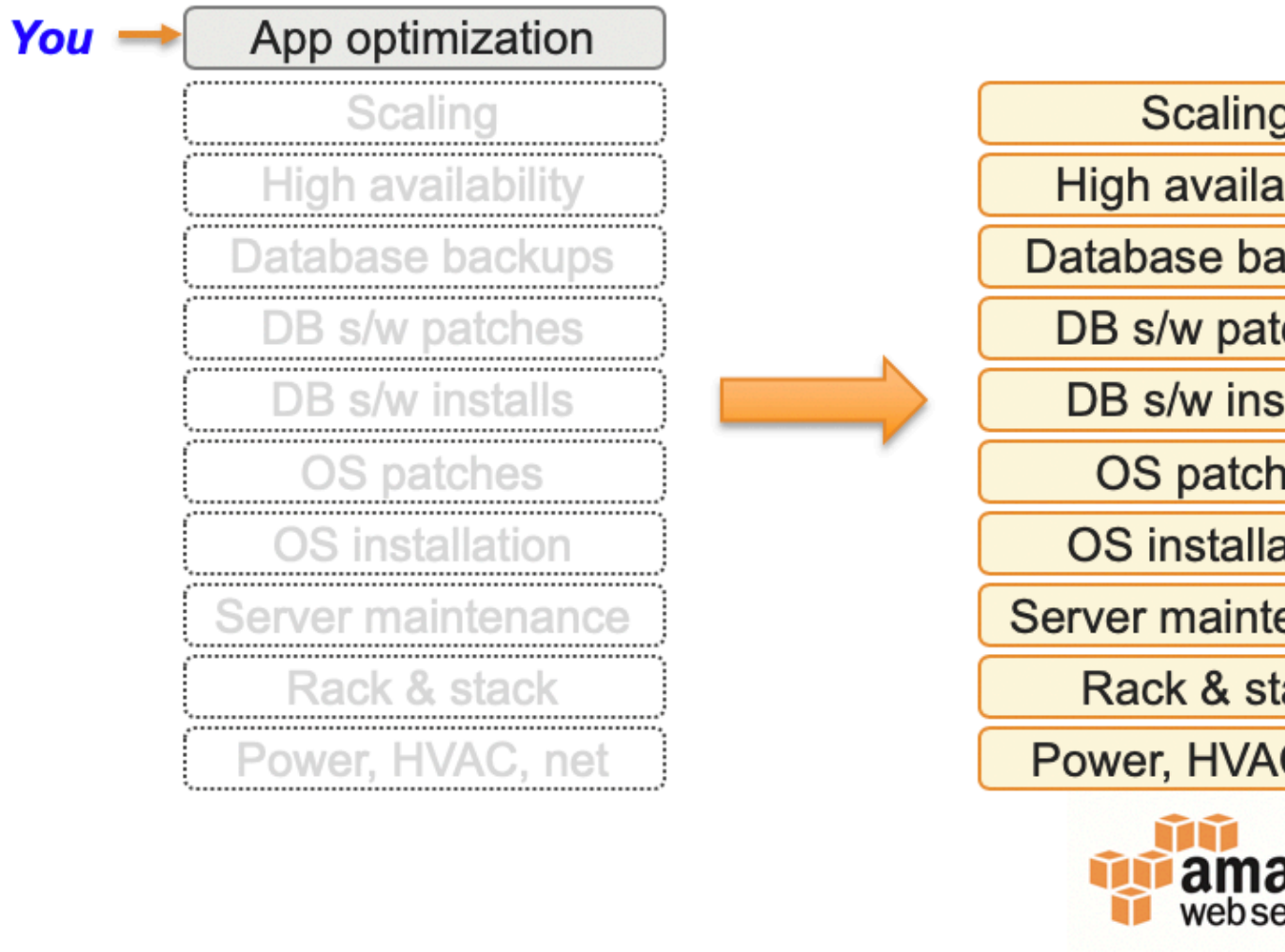
installation

maintenance

k & stack

, HVAC, net





- DB Options group
- DB instance class
- Read Replica
- High Availability
- Backup and restore
- Maintenance and patching

Tutorial: Create a Web Server and an Amazon RDS Database

- l
- bility
- ckups
- ches
- talls
- es
- tion
- enance
- ack
- C, net

amazon
services

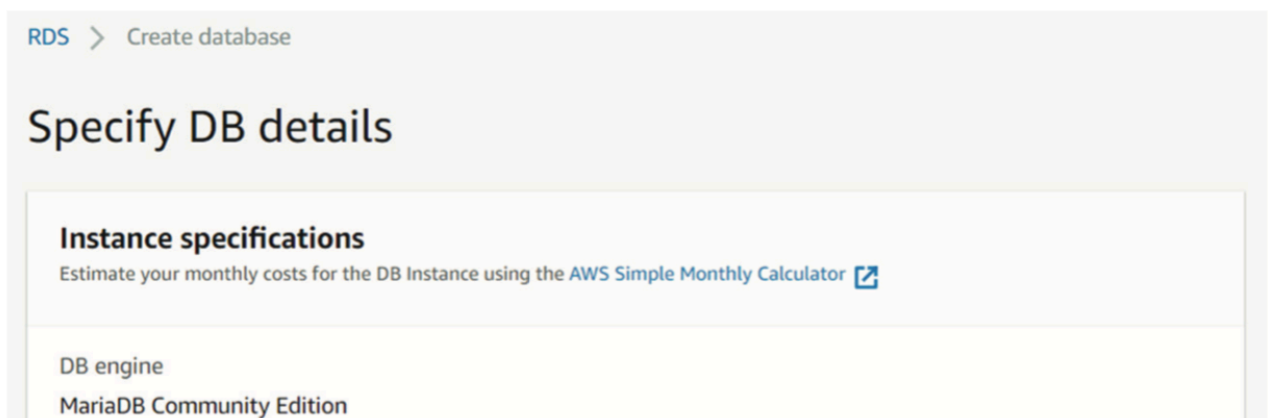
database

Ex 5.1: Create an RDS Database instance

In this exercise, you'll create an RDS database instance using MariaDB database engine. Free Tier accounts won't incur any charges.

1. **Go to the RDS Dashboard.**
2. **Click Create Database.**
3. **On the Select Engine page, select MariaDB and click Next.**
4. **For the use case, select Production - MariaDB. Then click Next.**
5. **For the DB Engine Version, stick with the default.**
6. **For DB Instance Class, select db.t2.micro - 1 vCPU, 1 GB RAM.**
7. **Choose No for Multi-AZ Deployment.**
8. **For Storage Type, select General Purpose (SSD).**
9. **Enter 20 GB as the Allocated Storage setting. Check your settings against the example in Figure 5.1 to make sure everything matches.**

Creating an RDS Instance



The screenshot shows the AWS RDS console interface for creating a new database. At the top, there is a breadcrumb trail: 'RDS > Create database'. Below this, the main heading is 'Specify DB details'. A section titled 'Instance specifications' is highlighted, containing a link to the 'AWS Simple Monthly Calculator' for cost estimation. Below this, the 'DB engine' is set to 'MariaDB Community Edition'.

RDS > Create database

Specify DB details

Instance specifications
Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#)

DB engine
MariaDB Community Edition

[\[1\]](#)

MariaDB as the

t.

Next.

RAM.

entries against the

The screenshot shows the configuration page for a new Amazon RDS database instance. The settings are as follows:

- License model:** [Info](#)
general-public-license
- DB engine version:** [Info](#)
mariadb 10.2.12
- DB instance class:** [Info](#)
db.t2.micro — 1 vCPU, 1 GiB RAM
- Multi-AZ deployment:** [Info](#)
 - ☐ Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
 - ☒ No
- Storage type:** [Info](#)
General Purpose (SSD)
- Allocated storage:**
20 GiB
(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage [may improve](#) IOPS performance.

10. In the Settings section, enter mydbinstance in the DB Instance Identifier field.
11. Enter a master username and master password of your choice and click Next.
12. In the Network & Security section, choose the Default VPC and Default Subnet Group settings.
13. For the Public Accessible setting, choose No.
14. For Availability Zone, select No Preference.
15. For VPC Security Group(s), select Create New VPC Security Group.
16. In the Database Options section, enter mydatabase in the Database Name field.
17. Leave Database Port set to the default of 3306.
18. Leave all the other settings at their defaults and click Create Instance.
19. Click View DB Instance Details to view the new database instance.
20. Keep the instance up and running, as you'll need it for Exercise 2.

Instance Identifier

choice and then click

PC and Default

ty Group.

ne Database Name

reate Database.

e instance.

xercise 5.2.

Ex 5.2:

Create a read replica

- 1. In the RDS Dashboard, click Instances.**
- 2. Select the instance you created earlier, click Instance Actions, then Create Read Replica.**
- 3. Under the Network & Security heading, stick with the default settings. No for Publicly Accessible.**
- 4. Under the Settings section, make sure the Read Replica Source Instance is mydbinstance.**
- 5. Enter mydbinstance-rr in the Database Instance Identifier field.**
- 6. Click the Create Read Replica button.**

Ex 5.3: Promote the read replica to a master

- 1. In the RDS console, click Instances.**
- 2. Select the read replica you created.**
- 3. Click Instance Actions and then select Promote Read Replica to Master.**
- 4. Click Continue.**
- 5. Click Promote Read Replica.**

Redshift

NoSQL Databases

tions, and then click

defaults, but choose

Source is

er field.

olica.

...NoSQL Databases

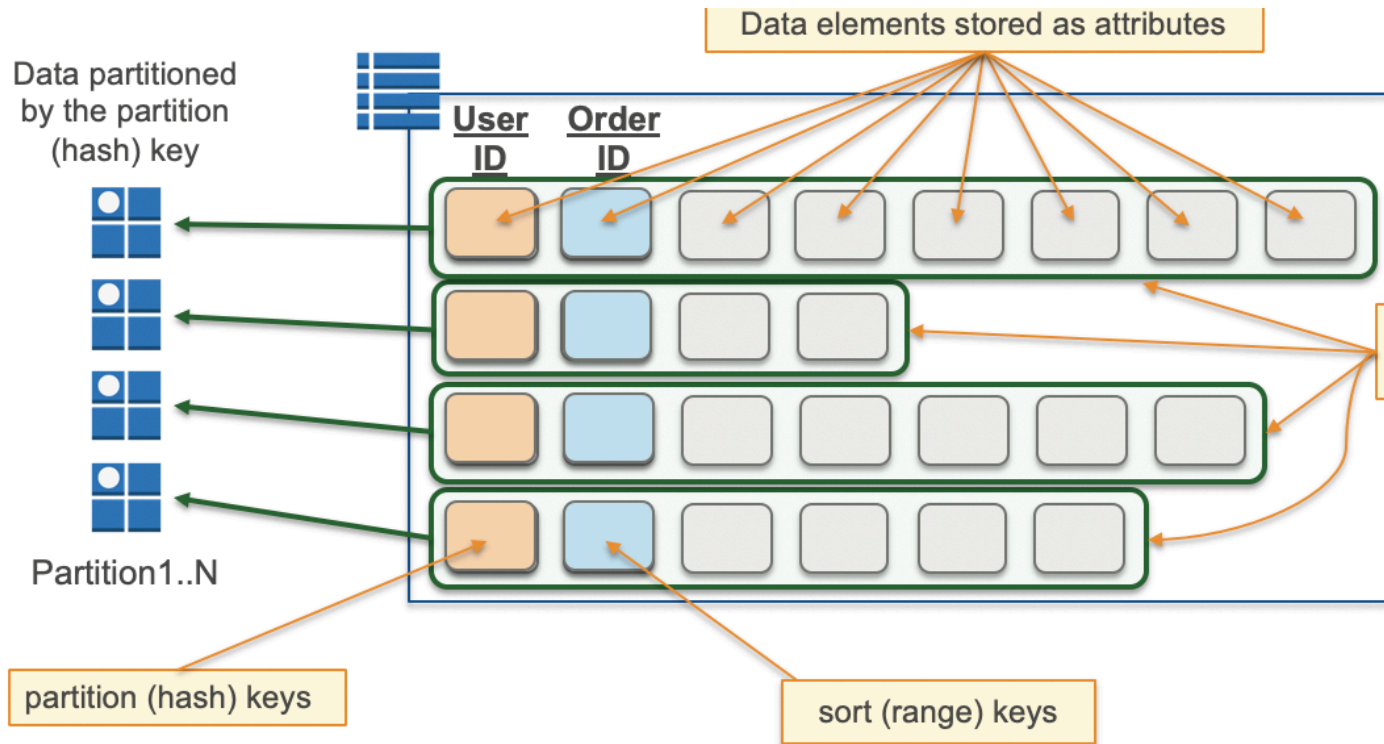
- Schemaless
- Primary Key (Partition Key)
- Denormalized tables
- Intelligent Keys
- Optimized for queries based on PK

Types of "non-relational databases"

DynamoDB

- Managed/Serverless
- Large reads/writes
- Single digit milliseconds latency
- Scales by using **partitions**
- SSD
- Partition Key/Hash Key
- Tables/Items/Attributes
- Consistency

DynamoDB Data Model



Tables Items and Attributes

People

<pre>{ "PersonID": 101, "LastName": "Smith", "FirstName": "Fred", "Phone": "555-4321" }</pre>
<pre>{ "PersonID": 102, "LastName": "Jones", "FirstName": "Mary", "Address": { "Street": "123 Main", "City": "Anytown", "State": "OH", "ZIPCode": 12345 } }</pre>

of data elements

Item = a collection
of attributes

```
graph TD; A[of data elements] --> B[Item = a collection of attributes];
```

The diagram consists of two yellow rectangular boxes with orange borders. The top box contains the text 'of data elements'. The bottom box contains the text 'Item = a collection of attributes'. An orange arrow points from the bottom-left corner of the top box to the top-left corner of the bottom box. A blue line is visible on the left side of the image, partially obscured by the boxes.


```
}
```

```
{
```

```
  "PersonID": 103,  
  "LastName": "Stephens",  
  "FirstName": "Howard",  
  "Address": {  
    "Street": "123 Main",  
    "City": "London",  
    "PostalCode": "ER3 5K8"  
  },  
  "FavoriteColor": "Blue"
```

```
}
```

Music

```
{
```

```
  "Artist": "No One You  
  "SongTitle": "My Dog  
  "AlbumTitle": "Hey N  
  "Price": 1.98,  
  "Genre": "Country",  
  "CriticRating": 8.4
```

```
}
```

```
{
```

```
  "Artist": "No One You  
  "SongTitle": "Somewhe  
  "AlbumTitle": "Somewh  
  "Genre": "Country",  
  "CriticRating": 8.4,  
  "Year": 1984
```

```
}
```

```
{
```

```
  "Artist": "The Acme B  
  "SongTitle": "Still i  
  "AlbumTitle": "The Bu  
  "Price": 2.47,  
  "Genre": "Rock",  
  "PromotionInfo": {  
    "RadioStationsPla  
      "KHCR",  
      "KQBX",  
      "WTNR",  
      "WJJH"  
    },  
    "TourDates": {  
      "Seattle": "2  
      "Cleveland":
```

```
  },
```

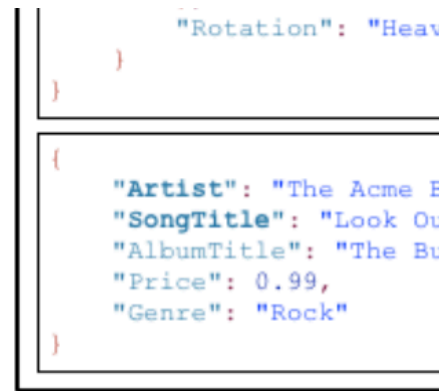
```
u Know",  
Spot",  
ow",
```

```
u Know",  
ere Down The Road",  
at Famous",
```

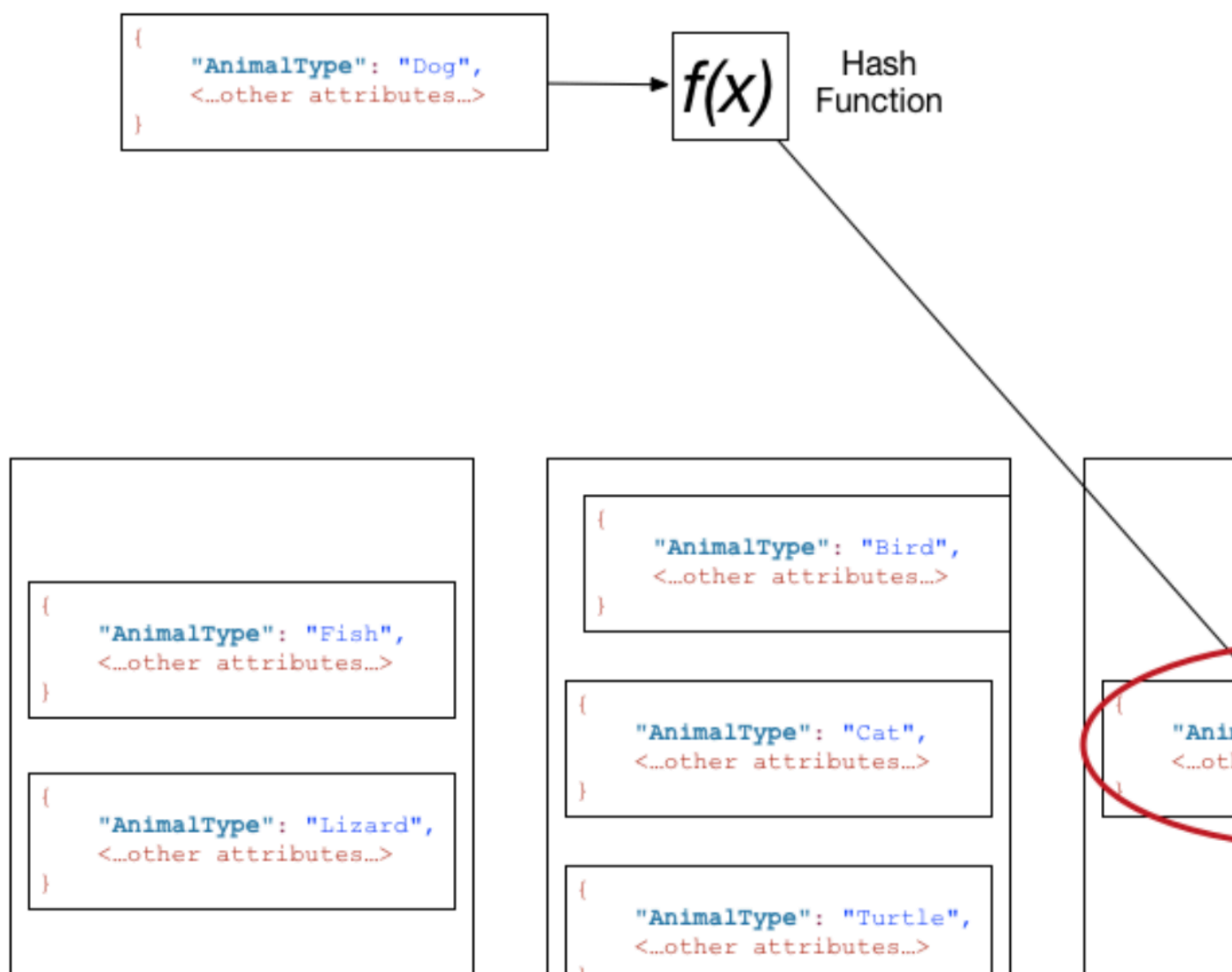
```
Band",  
in Love",  
ck Starts Here",
```

```
aying": [
```

```
20150625",  
"20150630"
```



Partition Keys



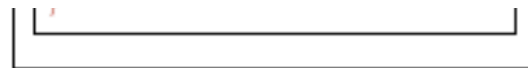
ry"

Band",
at, World",
uck Starts Here",

malType": "Dog",
her attributes...>

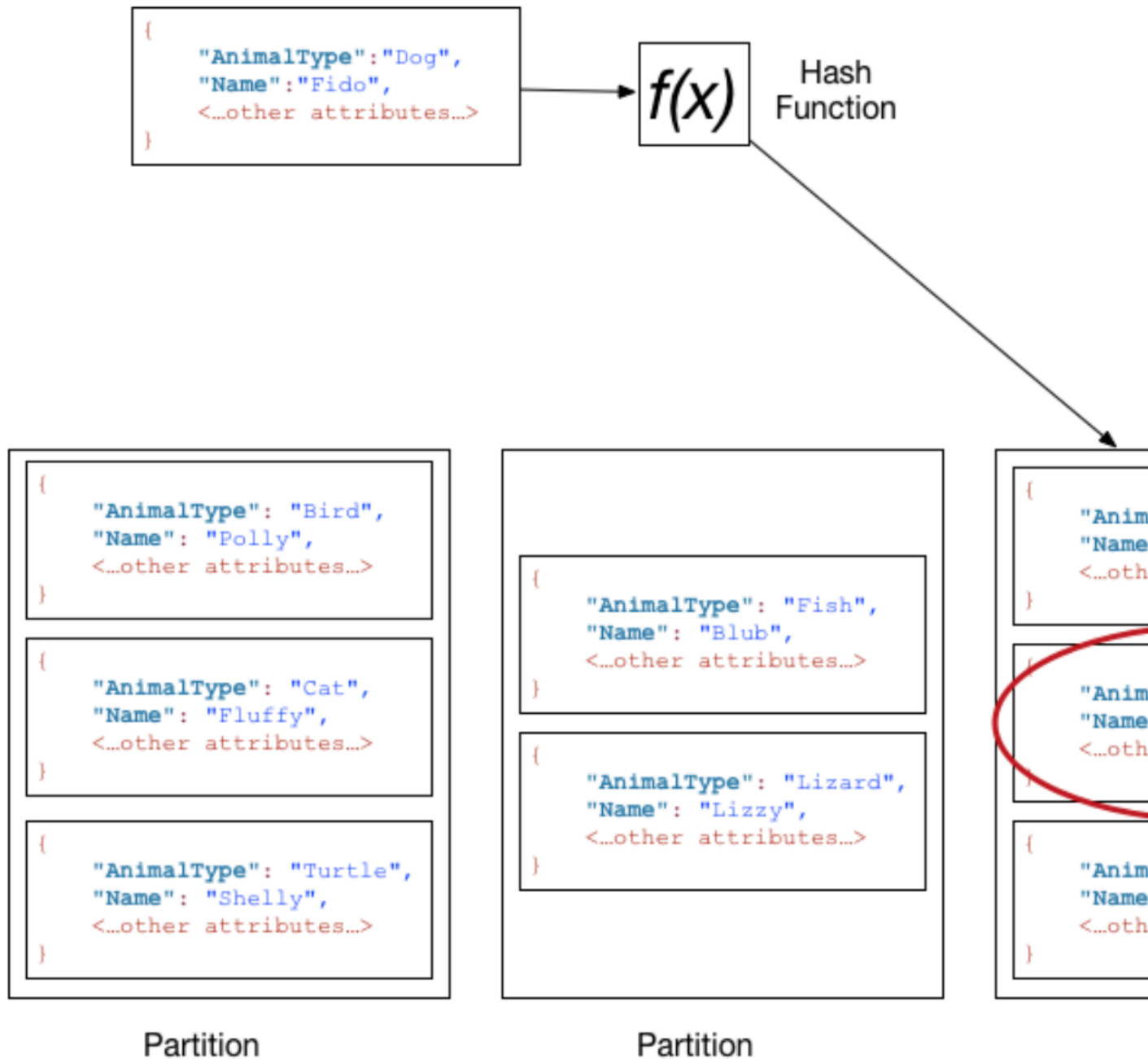


Partition



Partition

Partition and Sort Keys



Partition

```
alType": "Dog",  
": "Bowser",  
er attributes...>
```

```
alType": "Dog",  
": "Fido",  
er attributes...>
```

```
alType": "Dog",  
": "Rover",  
er attributes...>
```

Partition

Secondary Index



GenreAlbumTitle

```
Country",
e": "Hey Now",
"No One You Know",
": "My Dog Spot"
```

```
Country",
e": "Somewhat Famous",
"No One You Know",
': "Somewhere Down The Road"
```

```
Rock",
e": "The Buck Starts Here",
"The Acme Band",
': "Still in Love"
```

```
Rock"
e": "The Buck Starts Here",
```



```

{
  "AlbumTitle": "The Buck Starts Here",
  "Price": 0.99,
  "Genre": "Rock"
}

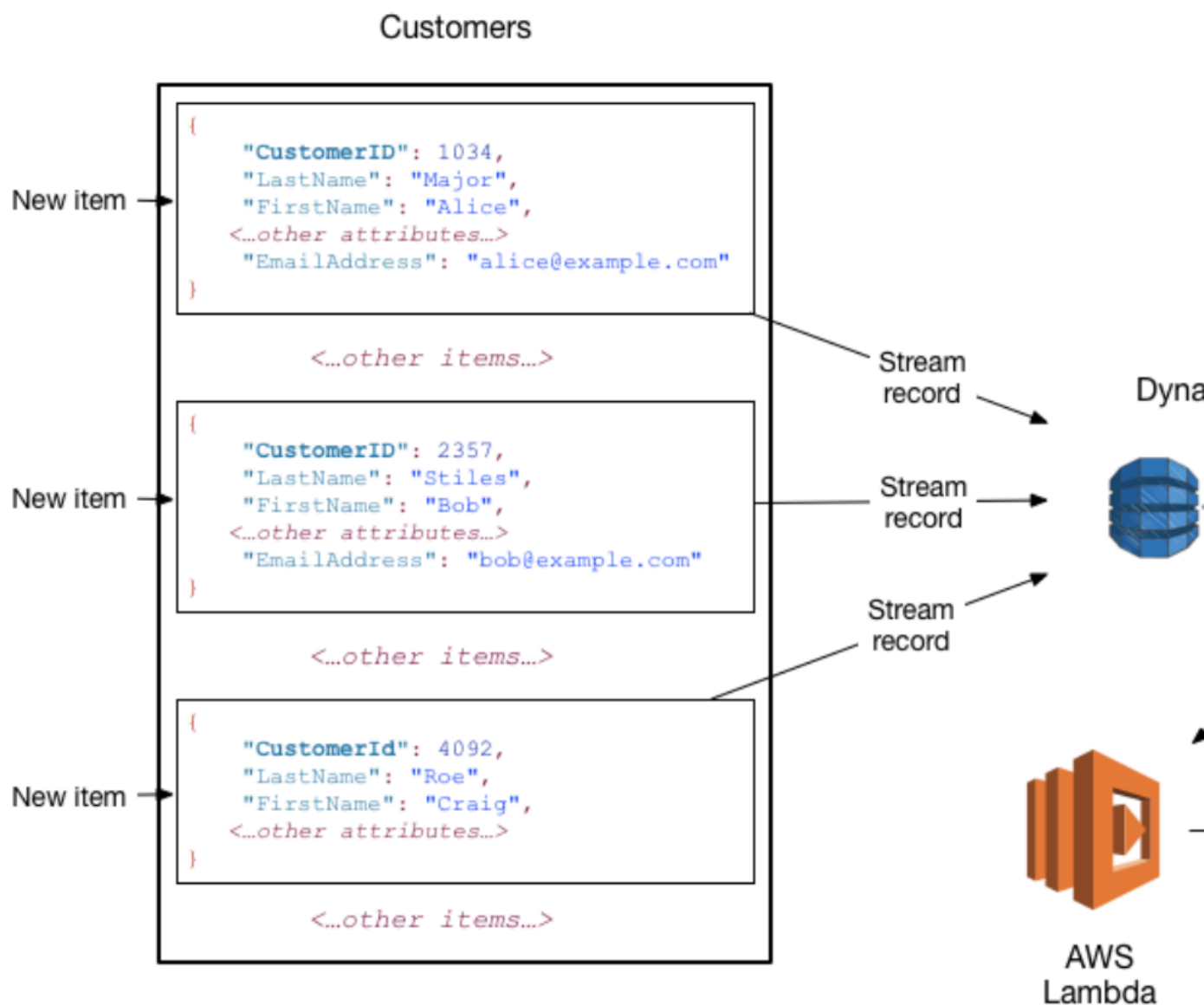
```

```

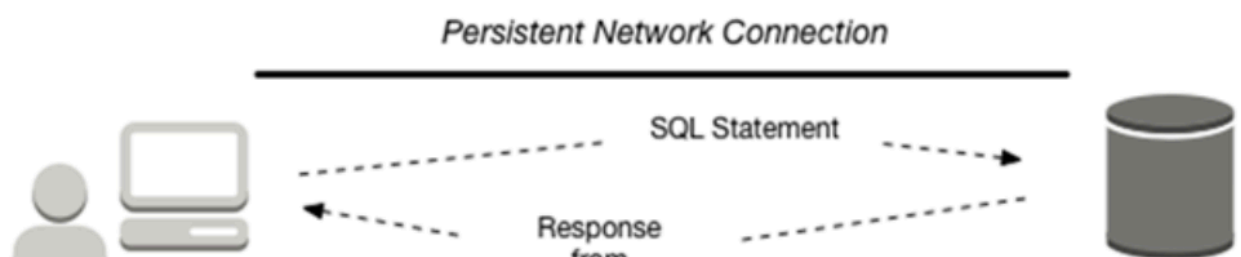
{
  "Artist": "
  "SongTitle":
}

```

DDB Streams



The following diagram shows a client's interaction with a relational database, and with Dynam



```
"The Acme Band",  
": "Look Out, World"
```

AmazonDB Streams



Amazon
SES

AmazonDB.



non-
Database

Relational
Database



<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStartedDDB.html>

RCU/WCU

DDB Best Practices

- Keep item size small.
- Store metadata in DynamoDB and large items in Amazon S3.



[DynamoDB.htm](#)

BLOBs

- Use table per day, week, month, etc., for time series data.
- Avoid hot keys and hot partitions.

Understand the differences between relational and nonrelational

Know the different database engines RDS supports.

Be able to select the right instance class and storage type given storage requirements.

Understand the differences between multi-AZ and read replicas

Be able to determine the appropriate primary key type for a table

Know how DynamoDB throughput capacity works

storing

ional databases

ven specific

icas

DynamoDB