



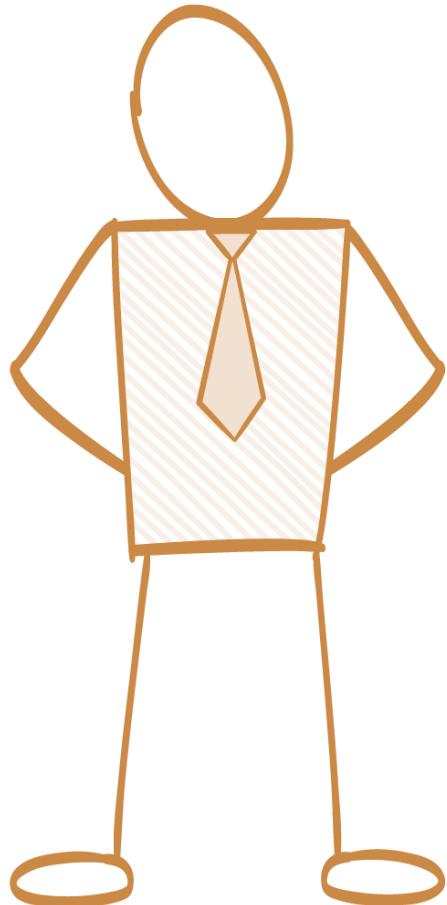
Training and  
Certification

# Architecting on AWS Student Guide

Version 3.1

100-ARC-31-EN-SG





## Module 10 : Overview of Database Services



# AWS Solutions Architect Associate

RDS



## Relational Database Service Introduction

# *Relational Database Service (RDS)*



A managed **relational database** service. Support multiple **SQL** engines, easy to scale, backup and secure.



# Introduction to RDS

Relational Database Service (RDS) is the AWS Solution for **relational** databases.  
There are **6 relational database** options currently available on AWS

Engine type [Info](#)

<input checked="" type="radio"/> Amazon Aurora 	<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 	<input type="radio"/> Microsoft SQL Server 



# AWS Solutions Architect Associate

RDS



## RDS Encryption



## RDS - Encryption

You can turn on encryption at-rest for all RDS engines

You may not be able to turn encryption on for older versions of some engines.

It will also encrypt the automated backups, snapshots, and read replicas.

Encryption is handled using the AWS Key Management Service (KMS)

### Encryption

#### Enable Encryption

Choose to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. [Info](#)

Master key [Info](#)

(default) aws/rds





# AWS Solutions Architect Associate

RDS



## RDS Backups



# RDS - Backup

There are 2 backup solutions available for RDS

The screenshot shows two panels. The top panel is titled 'Backup' and contains fields for 'Backup retention period' (set to 7 days), 'Backup window' (set to 06:00 UTC to 06:30 UTC), and a checkbox for 'Copy tags to snapshots'. The bottom panel shows an 'Actions' dropdown menu with options: Stop, Reboot, Delete, Create read replica, Create Aurora read replica, Promote, and Take snapshot, with 'Take snapshot' being the selected option.

## Automated Backups

Choose a Retention Period between 1 and 35 days

Stores transaction logs throughout the day

Automated backups are enabled by default

All data is stored inside S3

There is no additional charge for backup storage

You defined your backup window

Storage I/O may be suspended during backup

## Manual Snapshots

Taken manually by the user

Backups persist even if you delete the original RDS instance



# AWS Solutions Architect Associate

RDS



## RDS Restoring Backups



## RDS - Restoring Backup

When recovering AWS will take the most recent daily backup, and apply transaction log data relevant to that day. This allows point-in-time recovery down to a second inside the retention period.

Screenshot of the AWS RDS Actions menu:

- Actions ▾
- Restore from
- Stop
- Reboot
- Delete
- Create read replica
- Create Aurora read replica
- Promote
- Take snapshot
- Restore to point in time
- Migrate snapshot

Backup data is **never restored overtop** of an existing instance.

When you restore an RDS instance from Automated Backup or a Manual Snapshot a new instance is created for the restored database.

Restored RDS instances will have a new DNS endpoint.



# AWS Solutions Architect Associate

RDS



## RDS Multi-AZ



## RDS - Multi AZ

Ensures database remains available if another AZ becomes unavailable

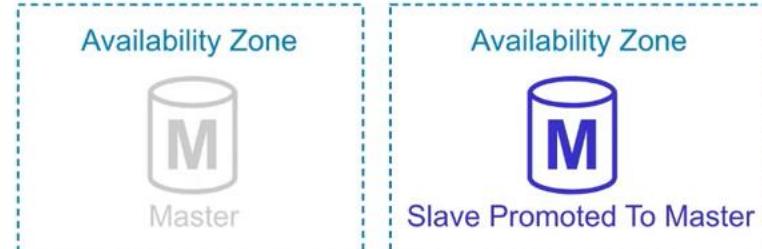
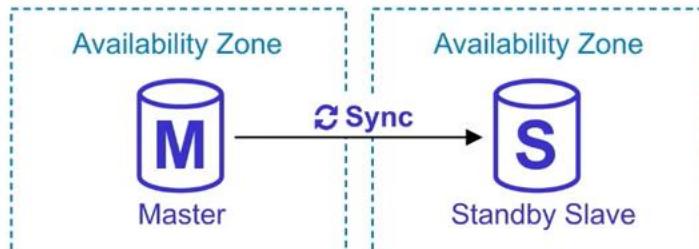
### Multi-AZ deployment

Specifies if the DB instance should have a standby deployed in another availability zone.

- Yes
- No

Makes an exact copy of your database in another AZ. AWS automatically **synchronizes** changes in the database over to the standby copy

**Automatic Failover protection** if one AZ goes down failover will occur and the standby slave will be promoted to master





# AWS Solutions Architect Associate

RDS

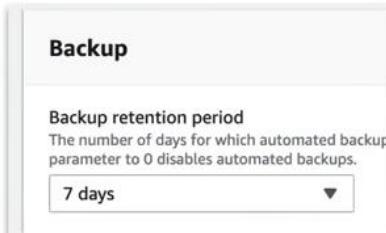


## RDS Read Replicas

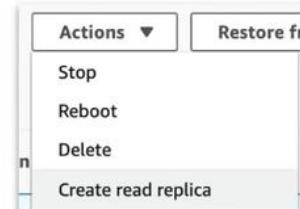


# RDS - Read Replicas

Read-Replicas allow you to run **multiple copies** of your database, these copies only allows **reads** (no writes) and is intended to alleviate the workload of your primary database to improve performance

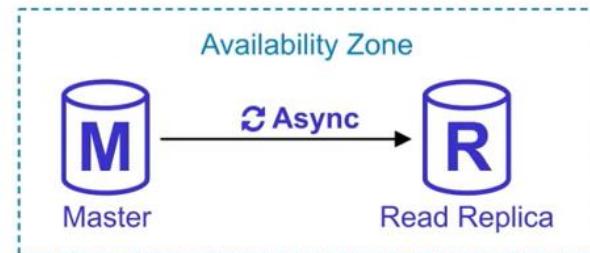


You must have automatic backups enabled to use Read Replicas



How to create a read replica:

**Asynchronous** replication happens between the primary RDS instance and the replicas.





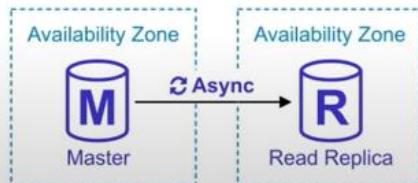
# RDS - Read Replicas

You can have up to  **5 replicas** of a database

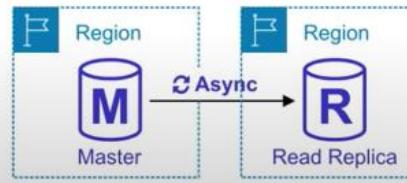
Each Read Replica will have its own DNS Endpoint

You can have Multi-AZ replicas, replicas in another region, or even replicas of other read replicas

**Multi-AZ Replicas**



**Cross-Region Replicas**



Replicas can be promoted to their own database, but this breaks replication

No automatic failover, if primary copy fails you must manually update urls to point at copy.



# AWS Solutions Architect Associate

RDS



## RDS Multi-AZ vs Read Replicas



# RDS - Multi-AZ vs Read Replicas

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance



# AWS Solutions Architect Associate

RDS



## RDS Cheat Sheet



# RDS *CheatSheet*

- Relational Database Service (RDS) is the AWS Solution for relational databases.
- RDS instances are managed by AWS, You cannot SSH into the VM running the database.
- There are 6 relational database options currently available on AWS, Aurora, MySQL, MariaDB, Postgres, Oracle, Microsoft SQL Server
- Multi-AZ is an option you can turn on which makes an exact copy of your database in another AZ that is only standby
- For Multi-AZ AWS automatically synchronizes changes in the database over to the standby copy
- Multi-AZ has Automatic Failover protection if one AZ goes down failover will occur and the standby slave will be promoted to master
- Read-Replicas allow you to run **multiples copies** of your database, these copies only allows **reads** (no writes) and is intended to alleviate the workload of your primary database to improve performance
- Read-Replicas use Asynchronous replication
- You must have automatic backups enabled to use Read Replicas



# RDS *CheatSheet*

- You can have upto 5 read replicas
- You can combine Read Replicas with Multi-AZ
- You can have Read Replicas in another Region (Cross-Region Read Replicas)
- Replicas can be promoted to their own database, but this breaks replication
- You can have Replicas of Read Replicas
- RDS has 2 backup solutions: Automated Backups and Database Snapshots
- Automated Backups, you choose a retention period between 1 and 35 days, There is no additional cost for backup storage, you define your backup window
- Manual Snapshots, you manually create backups, if you delete your primary the manual snapshots will still exist and can be restored
- When you restore an instance it will create a new database. You just need to delete your old database and point traffic to new restored database
- You can turn on encryption at-rest for RDS via KMS



# AWS Solutions Architect Associate

Aurora



## Introduction to Aurora

# Aurora



**Fully Managed** Postgres or MySQL compatible database  
**designed by default to scale** and **fine-tuned to be really fast**.



# Introduction to Aurora

Combines the **speed** and **availability** of **high-end databases** with the **simplicity** and **cost-effectiveness** of **open source databases**.

Aurora can run either MySQL or Postgres compatible engines



Aurora MySQL is **5x better performance** than traditional MySQL



Aurora Postgres is **3x better performance** than traditional Postgres

**1/10th** the costs of other solutions offering similar performance and availability.



# AWS Solutions Architect Associate

Aurora



## Scaling with Aurora



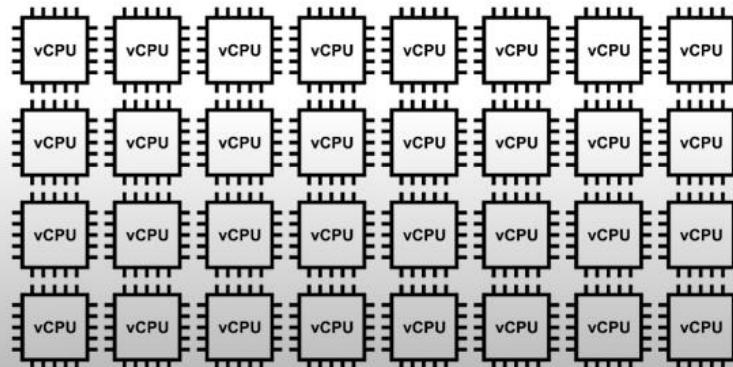
## Aurora - Scaling

Start with 10GB of storage, and scale in **10GB** increments up to 64TB.



Storage is autoscaling.

Computing resources can scale all the way up to **32 vCPUs** and **244GB** of memory.





# AWS Solutions Architect Associate

Aurora



## Availability with Aurora



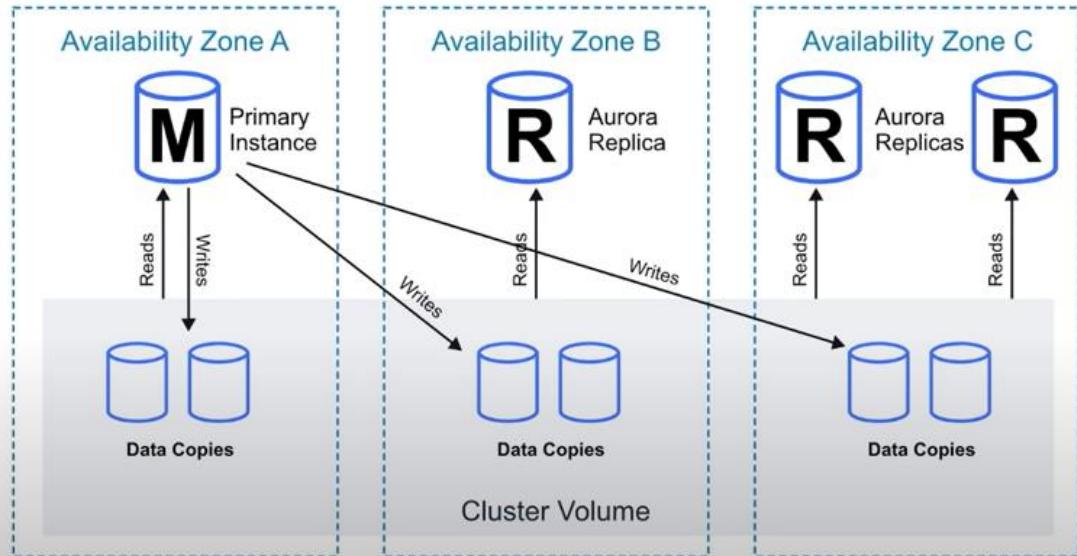
# Aurora - Availability

A minimum of **3 availability zones** each contain **2 copies** of your data at all times.

That means there are **6 copies**

Lose up to 2 copies of your data without affecting **write** availability.

Lose up to 3 copies of your data without affecting **read** availability.





# AWS Solutions Architect Associate

Aurora

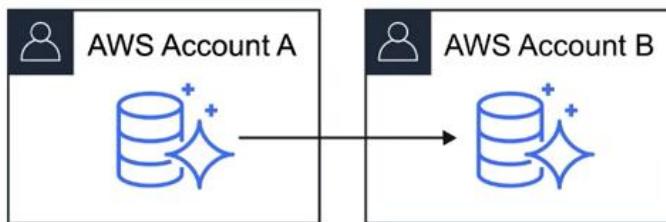


## Fault Tolerance & Durability



# Aurora - Fault Tolerance and Durability

Aurora Backup and Failover is handled **automatically**



Snapshots of data can be  
**shared** with other AWS  
accounts

Storage is **self-healing**, in that data blocks and  
disks are continuously scanned for errors and  
repaired automatically.



# AWS Solutions Architect Associate

Aurora



## Aurora Replicas



## Aurora - Replicas

There are  **2 types** of replicas available:

	<b>Amazon Aurora Replicas</b>	<b>MySQL Read Replicas</b>
Number of Replicas	Up to 15	Up to 5
Replication Type	Asynchronous (ms)	Asynchronous (s)
Performance impact on primary	Low	High
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs primary	No	Yes



# AWS Solutions Architect Associate

Aurora



## Aurora Serverless



# Aurora - Serverless

Engine type [Info](#)



Aurora except the database will automatically **start up, shut down, and scale capacity up or down** based on your application's needs.

Apps used a few minutes several times per day or week, eg. low-volume blog site

pay for database **storage** and the **database capacity** and **I/O** your database consumes while it is active

## Database features



### One writer and multiple readers

Supports multiple reader instances connected to the same storage volume as a single writer instance. This is a good general-purpose option for most workloads.



### Serverless

You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.

## Capacity settings

This billing estimate is based on published prices. [Learn more](#)

Minimum Aurora capacity unit [Info](#)

2  
4GB RAM

Maximum Aurora capacity unit [Info](#)

384  
768GB RAM

### Additional scaling configuration

Force scaling the capacity to the specified values when the timeout is reached [Info](#)

Enable to force capacity scaling as soon as possible. Disable to cancel the capacity changes when a timeout is reached

Pause compute capacity after consecutive minutes of inactivity [Info](#)

You are only charged for database storage while the compute capacity is paused





# AWS Solutions Architect Associate

Aurora



## Aurora Cheat Sheet



# Aurora CheatSheet

- When you need a **fully-managed** Postgres or MySQL database that needs to scale, automatic backups, high availability and fault tolerance *think* Aurora
- Aurora can run MySQL or Postgres database engines
- Aurora MySQL is 5x faster over regular MySQL
- Aurora Postgres is 3x faster over regular Postgres
- Aurora is 1/10 the cost over its competitors with similar performance and availability options.
- Aurora replicates **6 copies** for your database across **3 availability zones**.
- Aurora is allowed up to **15 Aurora Replicas**
- An Aurora database can span multiple regions via **Aurora Global Database**
- **Aurora Serverless** allows you to stop and start Aurora and scale automatically while keeping costs low
- Aurora Serverless is ideal for new projects or projects with infrequent database usage



# AWS Solutions Architect Associate

Redshift



## Redshift Overview

# *Amazon Redshift*



**Fully Managed Petabyte-size Data Warehouse.**  
**Analyze (Run complex SQL queries) on massive amounts of data**  
**Columnar Store database.**



# What is a Data Warehouse?

## What is a Database Transaction?

A transaction symbolizes a unit of work performed within a database management system  
eg. reads and writes

## Database

Online **Transaction** Processing (OLTP)

A database was built to store current transactions and enable **fast access to specific transactions** for ongoing business processes

VS

## Data Warehouse

Online **Analytical** Processing (OLAP)

A data warehouse is built to store large quantities of historical data and **enable fast, complex queries across all the data**

### Adding Items To Your Shopping List

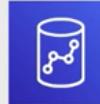
Single Source

short transactions (small and simple queries) with an emphasis on writes.

### Generating Reports

Multiple Sources

Long transactions (long and complex queries) with an emphasis on reads.



## Introduction of Redshift

AWS Redshift is the AWS managed, petabyte-scale solution for **Data Warehousing**.

Pricing starts at just \$0.25 per hour with no upfront costs or commitments.

Scale up to petabytes for \$1000 per terabyte, per year.

Redshift price is less than 1/10 cost of most similar services.

Redshift is used for Business Intelligence.

Redshift uses OLAP (Online Analytics Processing System)

Redshift is **Columnar Storage** Database

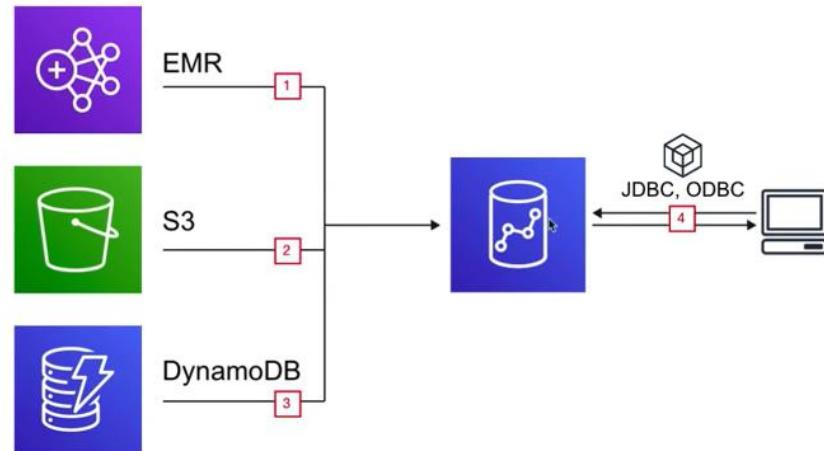
**Columnar storage** for database tables is an important factor in optimizing analytic query performance because it drastically reduces the overall disk I/O requirements and reduces the amount of data you need to load from disk.



## Redshift - Use Case

We want to continuously COPY data from  
1. EMR,  
2. S3 and  
3. DynamoDB  
to power a custom Business Intelligence tool.

Using a third-party library we can connect and query Redshift for data.





# Redshift - Columnar Storage

**Columnar Storage** stores data together as columns instead of rows.

Name	Rank	Species
John-Luc Picard	Captain	Human
Worf	Lieutenant	Klingon
Data	Lieutenant commander	Android

John-Luc Picard | Worf | Data

BLOCK 1

**OLAP** applications look at multiple records at the same time. You save memory because you fetch just the columns of data you need instead of whole rows

Since data is stored via column, that means all data is of the same data-type allowing for easy compression.



# Redshift - Configurations

## Single Node

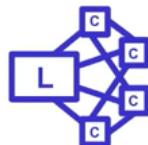
Nodes come in sizes of **160 GB**. You can launch a single node to get started with Redshift.

Cluster type **Single Node**

Number of compute nodes\* **1**

Maximum **1**

Minimum **1**



## Multi-Node

You can launch a cluster of nodes with Multi Node mode

**Leader Node** - manages client connections and receiving queries

**Compute Node** stores data and performs queries up to **128 compute nodes**

**Ask for AWS service limit increase**

Cluster type **Multi Node**

Number of compute nodes\*

Maximum **32**

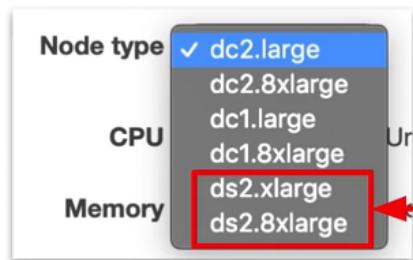
Minimum **2**



# Redshift - Node Types and Sizes

There are 🤝 two types of Nodes

The smallest node you can select is **dc2.large**



## Dense Compute (dc)

best for high performance, but they have less storage

## Dense Storage (ds)

clusters in which you have a lot of data



## Redshift - Compression

Redshift uses **multiple compression techniques** to achieve significant compression relative to traditional relational data stores.

**Similar data is stored sequentially** on disk.

Does **not** require **indexes** or **materialized views**, which saves a lot of space **compared to traditional systems**.

When loading data to an **empty table**, data is **sampled** and the most appropriate compression **scheme is selected automatically**.

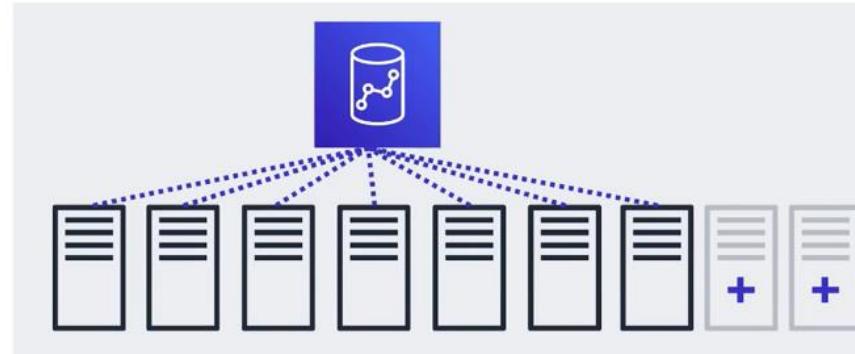


## Redshift – Processing

Redshift uses **Massively Parallel Processing (MPP)**.

**Automatically distributes** data and query loads **across all nodes**.

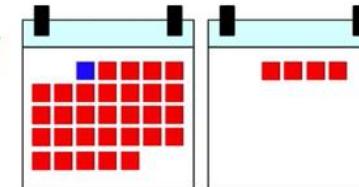
Lets you easily **add new nodes** to your data warehouse while still **maintaining fast query performance**.





## Redshift - Backups

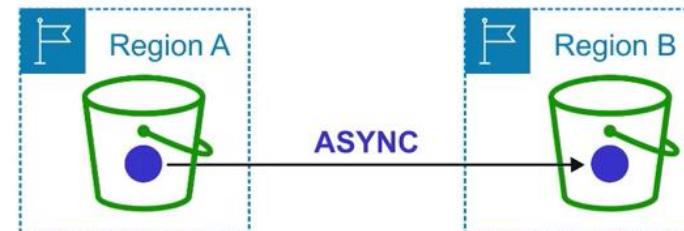
Backups are **enabled by default** with a **1 day** retention period. Retention period can be modified **up to 35 days**.



Redshift always attempts to maintain at least **3 copies of your data**.

1. The original copy
2. Replica on the compute nodes
3. Backup copy in S3

Can asynchronously replicate your snapshots to S3 in a **different region**





## Redshift - Billing

### Compute Node Hours

- The total number of hours ran across all nodes in the billing period
- Billed for 1 unit per node, per hour.
- **Not charged for leader node hours**, only compute notes incur charges

### Backup

- Backups are stored on S3 and you are billed the S3 storage fees

### Data Transfer

- Billed for Only transfers within a VPN, not outside of it



## Redshift - Security

**Data-in-transit** - Encrypted using SSL

**Data-at-rest** - Encrypted using AES-256 encryption

Database Encryption can be applied using

- Key Management Service (KMS) multi-tenant HSM
- CloudHSM single-tenant HSM

Database encryption  None  KMS  HSM [Learn more about database encryption](#)

Master key

Description Default master key that protects my Redshift clusters when no other key is defined

Account This account (655604346524)

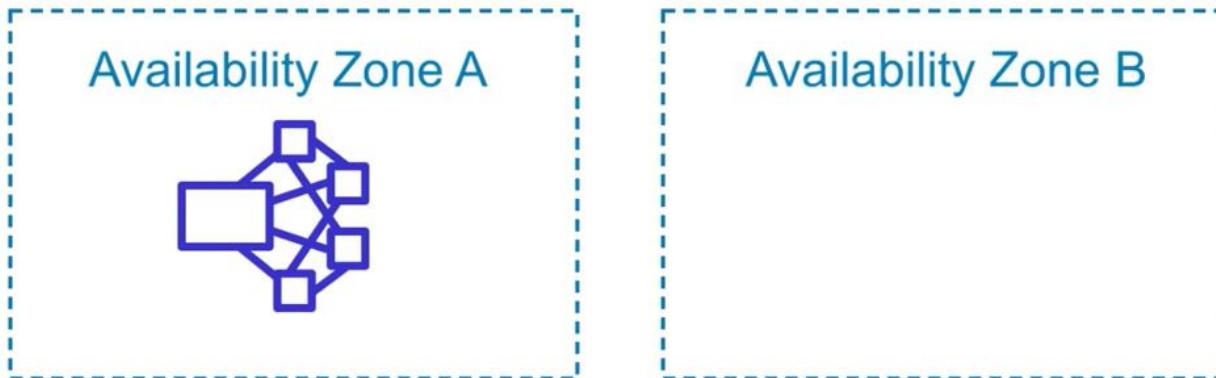
KMS key ID alias/aws/redshift



## Redshift - Availability

Redshift is **Single-AZ**. To run in Multi-AZ you would have to run multiple RedShift Clusters in different AZs with same inputs.

Snapshots **can be restored to a different AZ** in the event an outage occurs





# Redshift *CheatSheet*

---

- Data can be loaded from S3, EMR, DynamoDB, or multiple data sources on remote hosts.
- Redshift is Columnar Store database which can SQL-like queries and is an OLAP.
- Redshift can handle petabytes worth of data. Redshift is for Data Warehousing
- Redshift most common use case is Business Intelligence
- Redshift can only run in a 1 availability zone (**Single-AZ**)
- Reshift can run via a single node or multi-node (clusters)
- A single node is 160 GB in size
- A multi-node is comprised of a leader node and multiple compute nodes
- You are bill per hour for each node (excluding leader node in multi-node)
- You are not billed for the leader node
- You can have up to 128 compute nodes
- Redshift has two kinds of Node Type **Dense Compute** and **Dense Storage**
- Redshift attempts to backup 3 copies of your data, the original, on compute node and on S3
- Similar data is stored on disk sequentially for faster reads
- Redshift database can be encrypted via KMS or CloudHSM
- Backup Retention is default to 1 day and can be increase to maximum of 35 days
- Reshift can asynchronously back up your snapshot to Another Region delivered to S3
- Redshift uses Massively Parallel Processing (MPP) to distribute queries and data across all loads
- In the case of empty table, when importing Redshift will sample data to create a schema.



# AWS Solutions Architect Associate

DynamoDB



## Introduction to DynamoDB

# *DynamoDB*



A key-value and document database (NoSQL) which can guarantees **consistent reads and writes** at any scale.



# Introduction to DynamoDB

## What is NoSQL

NoSQL is database which is neither relational and does not use SQL to query the data for results

### What is a Key/Value Store?

A form of data storage which has a key which references a value and nothing more

```
{ Title: 'S01E019 DS9 Duet' }
```

### What is a Document Store?

A form of data storage which a nested data structure

```
{
  Series: 'DS9'
  Episodes: [
    {
      Season: 1,
      Episode: 19,
      Title: 'Duet'
    }
  ]
}
```



# Introduction to DynamoDB

DynamoDB is a NoSQL **key/value** and **document** database for internet-scale applications.

## Features

- Fully managed
- Multiregion
- Multimaster
- Durable database
- Built-in security
- Backup and restore
- In-memory caching

Specify your read and write capacity per second, it just works at **whatever capacity you need** without you tweaking anything.

Provisioned capacity

	Read capacity units	Write capacity units
Table	100	100

Estimated cost \$58.04 / month ([Capacity calculator](#))

## Provides

- Eventual Consistent Reads (default)
- Strongly Consistent Reads



All data is stored on **SSD storage** and is spread across **3 different regions**.



# AWS Solutions Architect Associate

DynamoDB



## Table Structure



# DynamoDB - Table Structure

Primary Key			
Partition Key	Sort Key		
IMDB ID	Year	Title	Box Office
tt0079945	1979	Star Trek: The Motion Picture	139000000
tt0084726	1982	Star Trek II: The Wrath of Khan	97000000
tt0088170	1984	Star Trek III: The Search for Spock	87000000
tt0092007	1986	Star Trek IV: The Voyage Home	133000000
tt0098382	1989	Star Trek V: The Final Frontier	63000000
tt0102975	1991	Star Trek VI: The Undiscovered Country	96900000
tt0111280	1994	Star Trek Generations	118000000
tt0117731	1996	Star Trek: First Contact	146000000
tt0120844	1998	Star Trek: Insurrection	117800000
tt0253754	2002	Star Trek: Nemesis	67300000



# AWS Solutions Architect Associate

DynamoDB



## Consistent Reads



# DynamoDB - Reads

When data needs to be updated it has to write updates to all copies. **It is possible for data to be inconsistent** if you are reading from a copy which has yet to be updated. You have the ability to choose the read consistency in DynamoDB to meet your needs.



## Eventual Consistent Reads (DEFAULT)

When copies are being updated it is possible for you to read and be returned an inconsistent copy

Reads are fast but there is no guarantee of consistent

All copies of data eventually become generally consistent within a second.

## Strongly Consistent Reads

When copies are being updated and you attempt to read, it will not return a result until all copies are consistent.

You have a guarantee of consistency but the trade off is higher latency (slower reads).

All copies of data will be consistent within a second



# AWS Solutions Architect Associate

DynamoDB



## DynamoDB Cheat Sheet



# DynamoDB *CheatSheet*

- **DynamoDB** is a fully managed **NoSQL** key/value and document database.
- Applications that contain large amounts of data but require predictable read and write performance while scaling is a good fit for DynamoDB
- DynamoDB scales with whatever **read and write capacity you specific** per second.
- DynamoDB can be set to have **Eventually Consistent Reads (default)** and **Strongly Consistent Reads**
- **Eventually consistent reads** data is returned immediately but data can be inconsistent. Copies of data will be generally consistent in 1 second.
- **Strongly Consistent Reads** will wait until data is consistent. Data will never be inconsistent but latency will be higher. Copies of data will be consistent with a guarantee of 1 second.
- DynamoDB stores 3 copies of data on SSD drives across 3 regions.