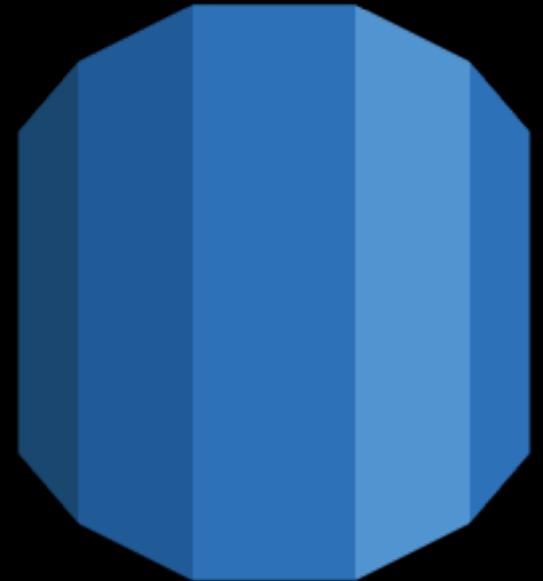
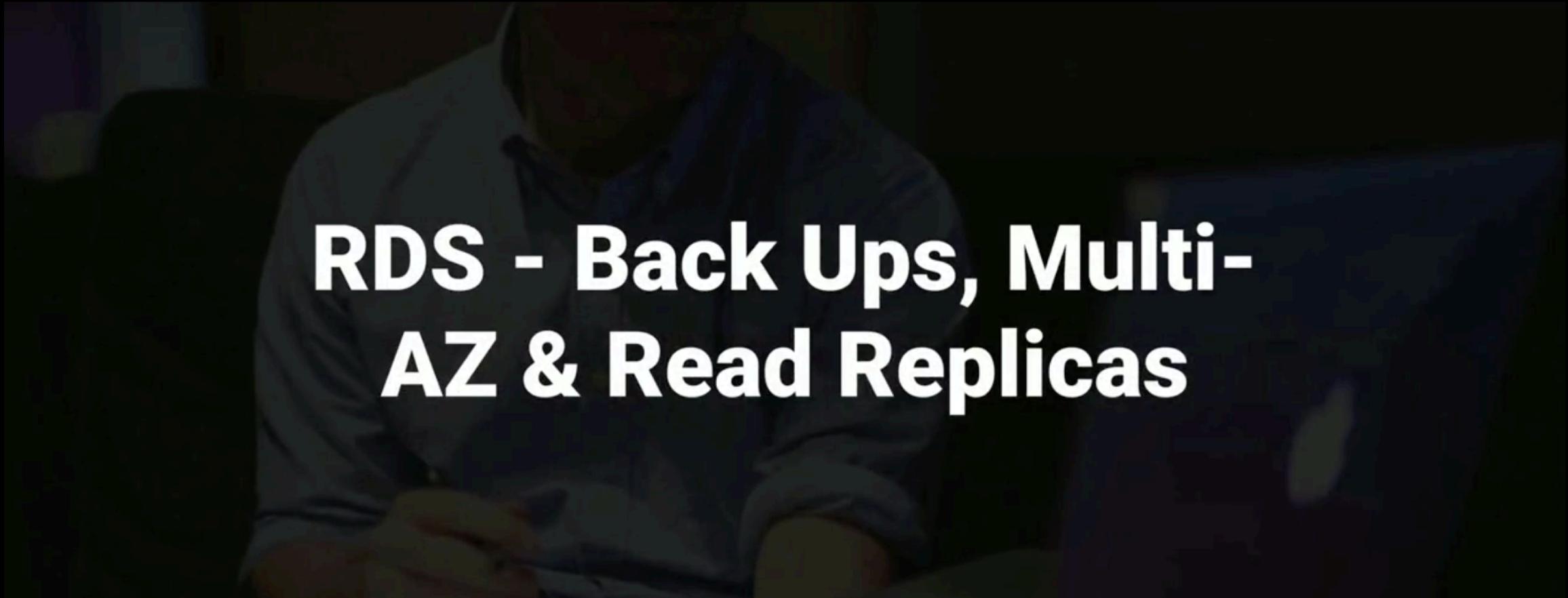


# RDS





RDS - Back Ups, Multi-AZ & Read Replicas

# Automated Backups

There are two different types of Backups for AWS. Automated Backups and Database Snapshots.

Automated Backups allow you to recover your database to any point in time within a “retention period”. Retention period can be between one and 35 days. Automated Backups will take a full daily snapshot and will also store transaction logs throughout the day. When you do a recovery, AWS will first choose the most recent daily back up, and then apply transaction logs relevant to that day. This allows you to do a point in time recovery down to a second, within the retention period.

## Automated Backups

Automated Backups are enabled by default. The backup data is stored in S3 and you get free storage space equal to the size of your database. So if you have an RDS instance of 10Gb you will get 10Gb worth of storage.

Backups are taken within a defined window. During the backup window, storage I/O may be suspended while your data is being backed up and you may experience elevated latency.

# Snapshots

DB Snapshots are done manually (ie they are user initiated). They are stored even after you delete the original RDS instance, unlike automated backups.

## Restoring Backups

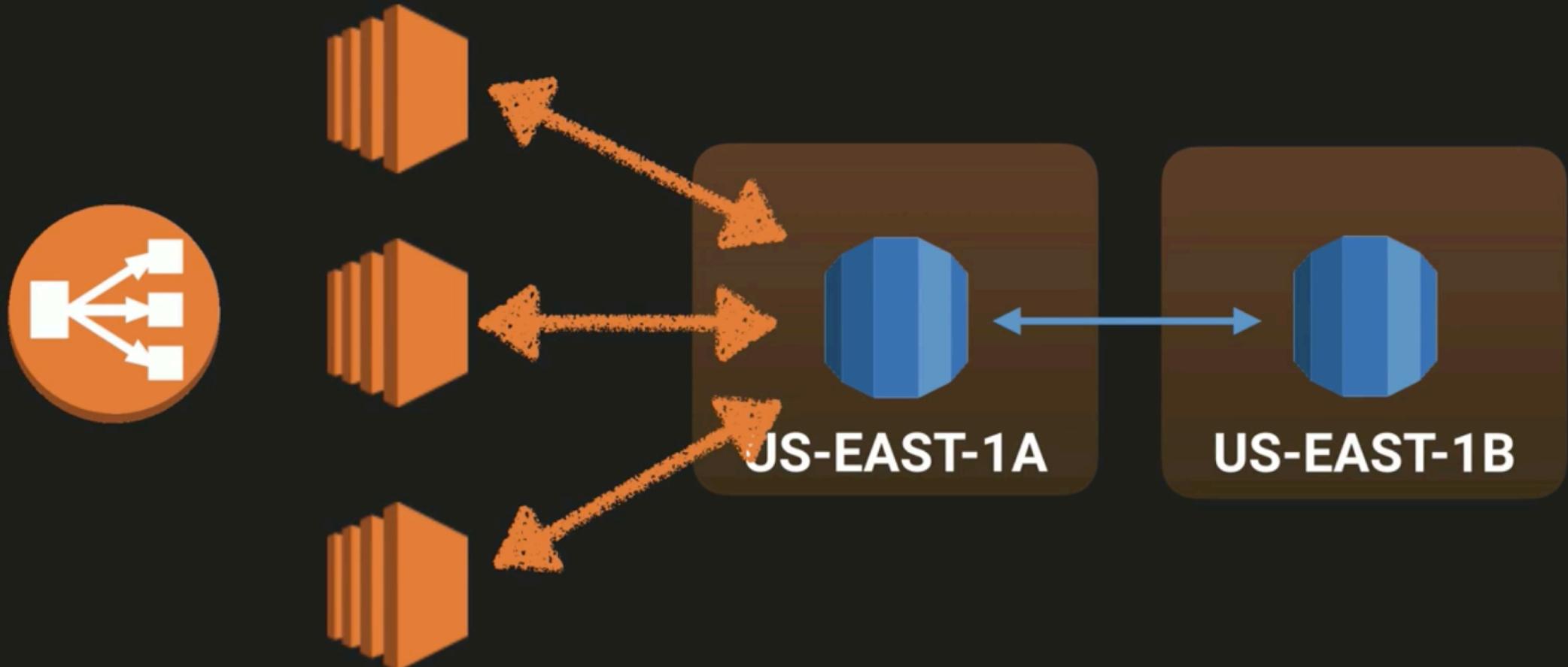
When ever you restore either an Automatic Backup or a manual Snapshot, the restored version of the database will be a new RDS instance with a new end point.

# Encryption

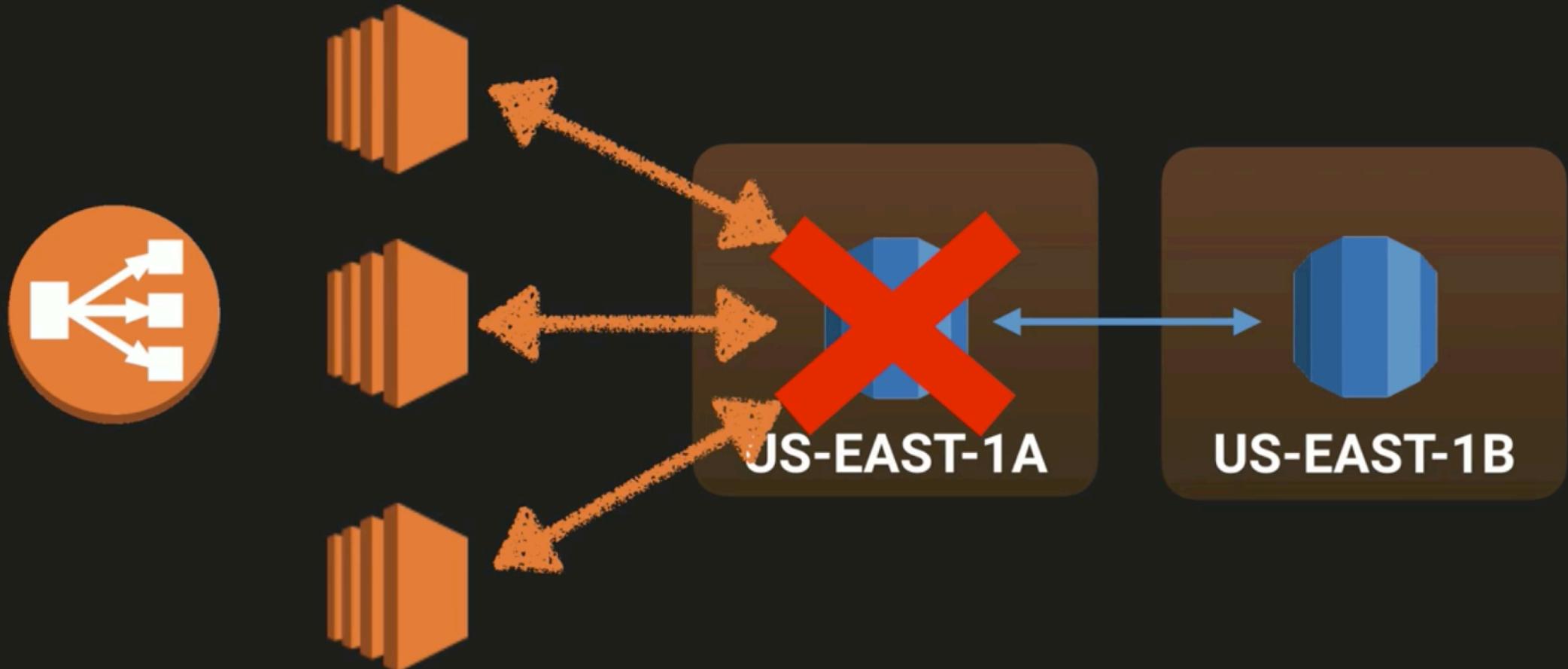
Encryption at rest is supported for MySQL, Oracle, SQL Server, PostgreSQL & MariaDB. Encryption is done using the AWS Key Management Service (KMS) service. Once your RDS instance is encrypted the data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas, and snapshots.

At the present time, encrypting an existing DB Instance is not supported. To use Amazon RDS encryption for an existing database, create a new DB Instance with encryption enabled and migrate your data into it.

# What is a Multi-AZ?



# What is a Multi-AZ?



## What is Multi-AZ RDS?

Multi-AZ allows you to have an exact copy of your production database in another Availability Zone. AWS handles the replication for you, so when your production database is written to, this write will automatically be synchronised to the stand by database.

In the event of planned database maintenance, DB Instance failure, or an Availability Zone failure, Amazon RDS will automatically failover to the standby so that database operations can resume quickly without administrative intervention.

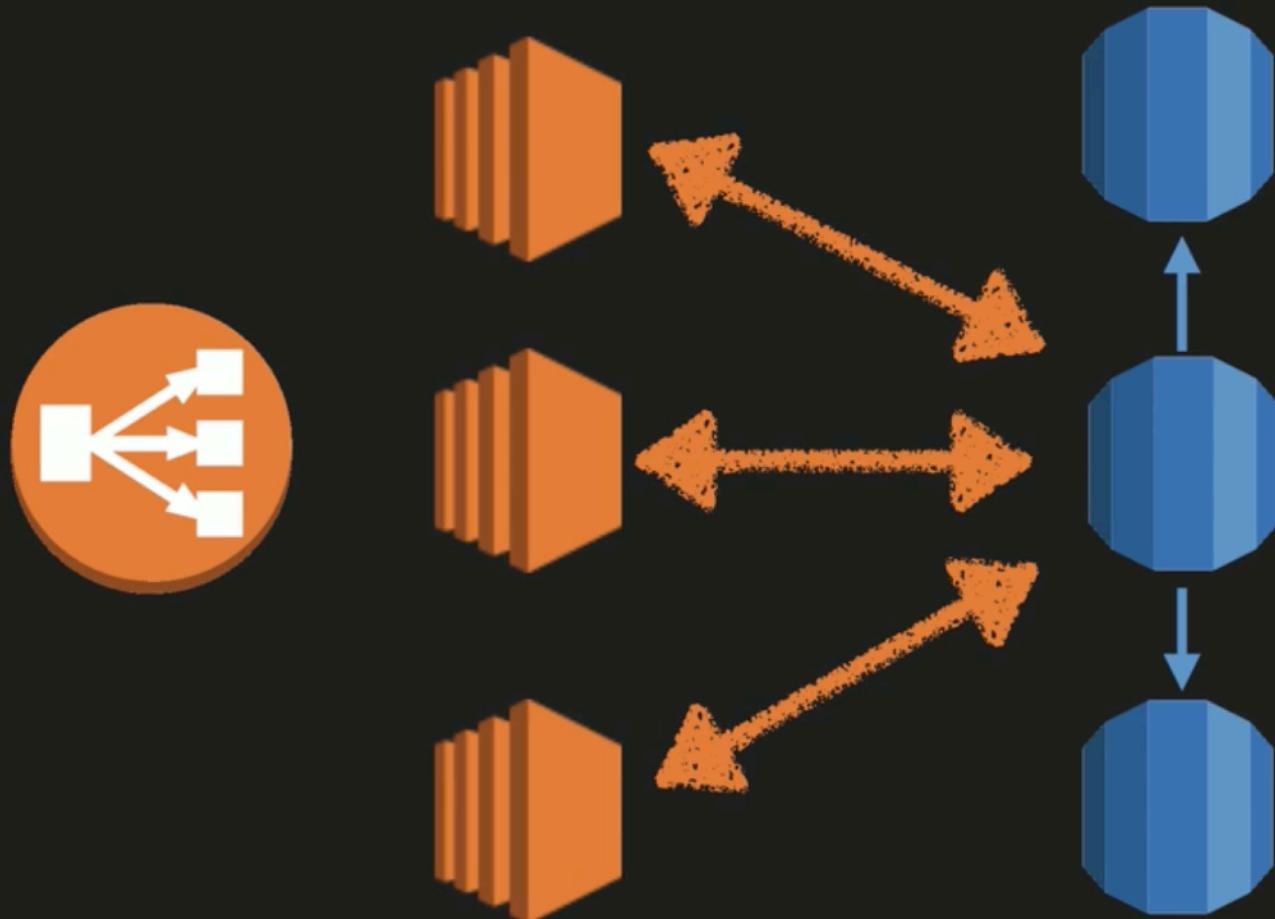
# What is Multi-AZ RDS?

Multi-AZ is for Disaster Recovery only. It is not primarily used for improving performance. For performance improvement you need Read Replicas.

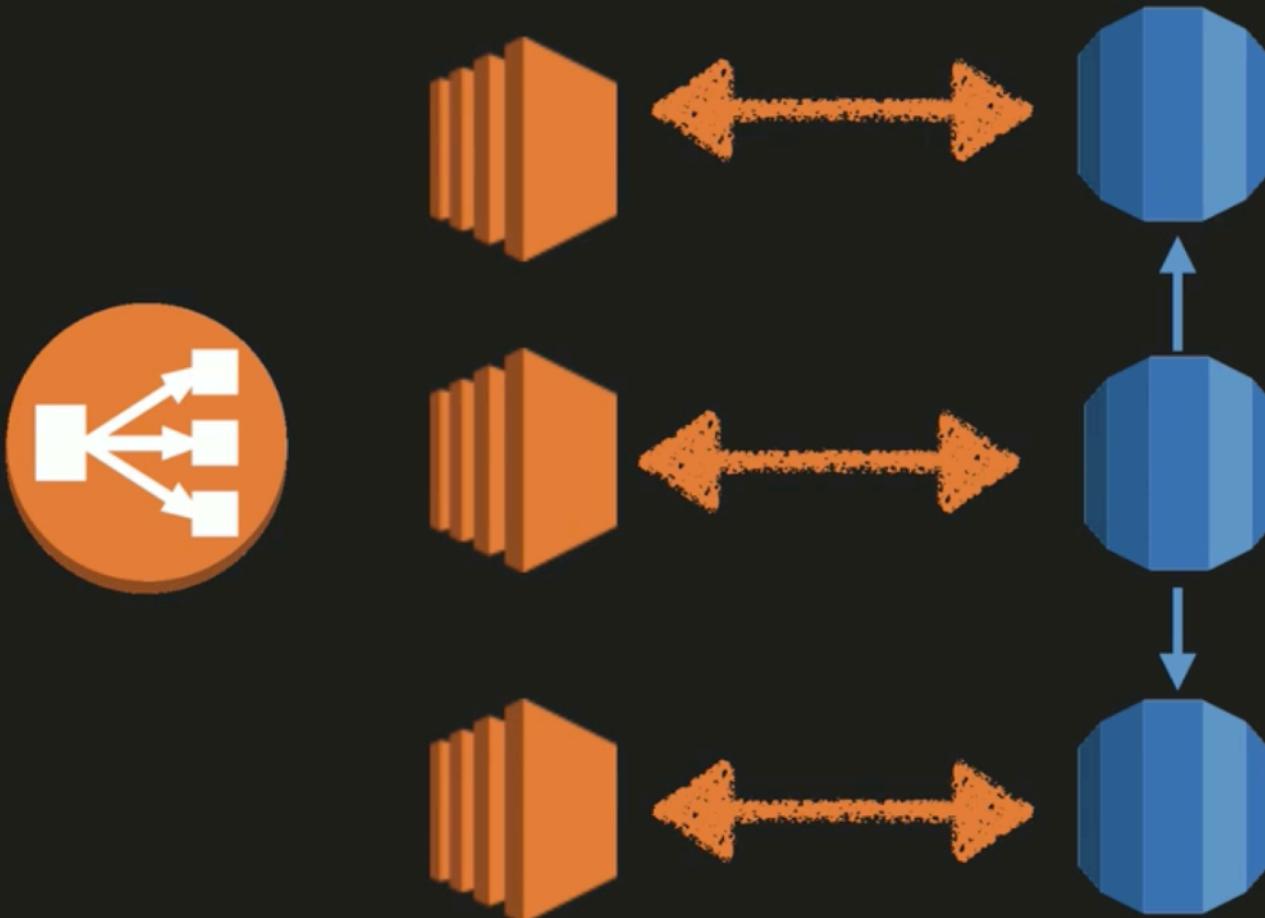
# Multi-AZ Databases

- SQL Server
- Oracle
- MySQL Server
- PostgreSQL
- MariaDB

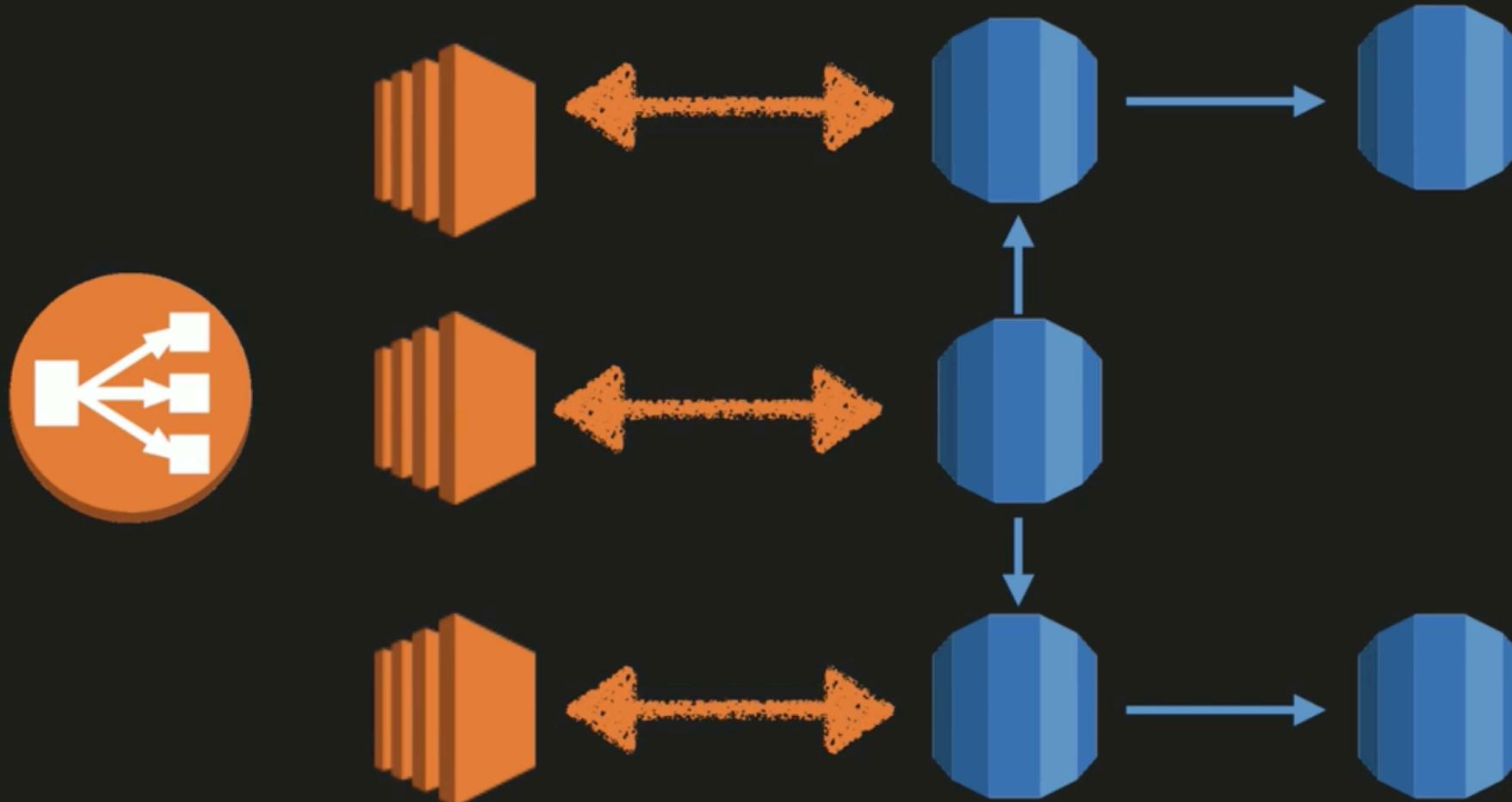
# What is a Read Replica?



# What is a Read Replica?



# What is a Read Replica?



# What is a Read Replica?

Read replica's allow you to have a read only copy of your production database. This is achieved by using Asynchronous replication from the primary RDS instance to the read replica. You use read replica's primarily for very read-heavy database workloads.

# Read Replica Databases

- MySQL Server
- PostgreSQL
- MariaDB

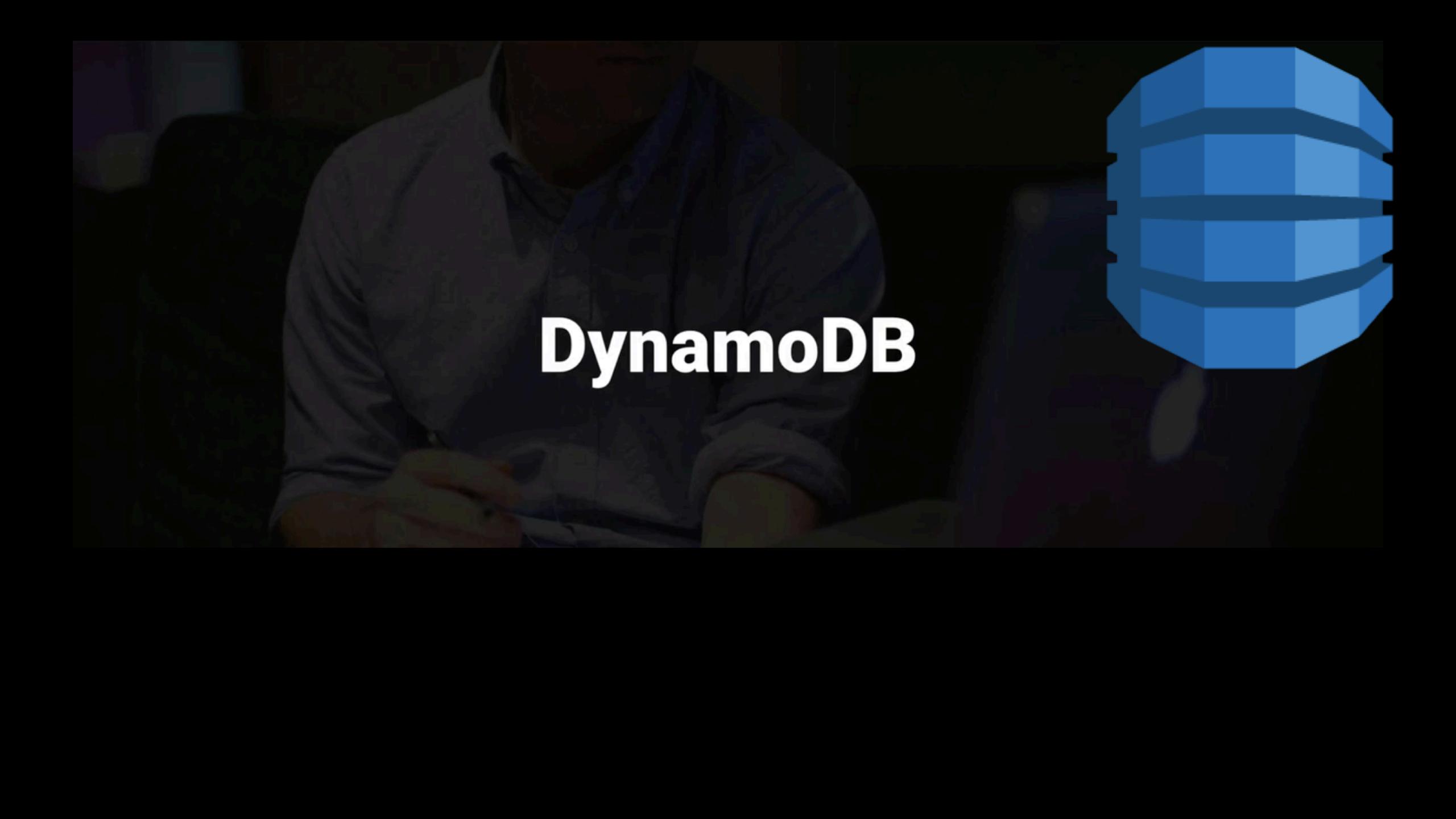
# Read Replica Databases

- Used for Scaling!!! Not for DR!
- Must have automatic backups turned on in order to deploy a read replica
- You can have up to 5 read replicas copies of any databases
- You can have read replicas of read replicas (but watch out for latency)
- Each read replica will have its own DNS end point
- You cannot have Read Replicas that have Multi-AZ
- You can create Read Replica's of Multi-AZ source databases however
- Read Replicas can be promoted to be their own databases. This breaks the replication.
- Read Replica in a second region for MySQL and MariaDB. Not for PostgreSQL

# DynamoDB vs RDS

DynamoDB offers “push button” scaling, meaning that you can scale your database on the fly, without any down time.

RDS is not so easy and you usually have to use a bigger instance size or to add a read replica.



# DynamoDB



# What is DynamoDB?

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed database and supports both document and key-value data models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

# DynamoDB

- Stored on SSD storage
- Spread Across 3 geographically distinct data centres
- Eventual Consistent Reads (Default)
- Strongly Consistent Reads

# DynamoDB

- Eventual Consistent Reads
  - Consistency across all copies of data is usually reached within a second. Repeating a read after a short time should return the updated data. (Best Read Performance)
- Strongly Consistent Reads
  - A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.

# DynamoDB Pricing

- Provisioned Throughput Capacity
  - Write Throughput \$0.0065 per hour for every 10 units
  - Read Throughput \$0.0065 per hour for every 50 units
- Storage costs of \$0.25Gb per month.

## Pricing Example

Let's assume that your application needs to perform 1 million writes and 1 million reads per day, while storing 3 GB of data.

First, you need to calculate how many writes and reads per second you need.

1 million evenly spread writes per day is equivalent to  $1,000,000 \text{ (writes)} / 24 \text{ (hours)} / 60 \text{ (minutes)} / 60 \text{ (seconds)} = 11.6 \text{ writes per second}$ .

## Pricing Example

A DynamoDB Write Capacity Unit can handle 1 write per second, so you need 12 Write Capacity Units.

Similarly, to handle 1 million strongly consistent reads per day, you need 12 Read Capacity Units.

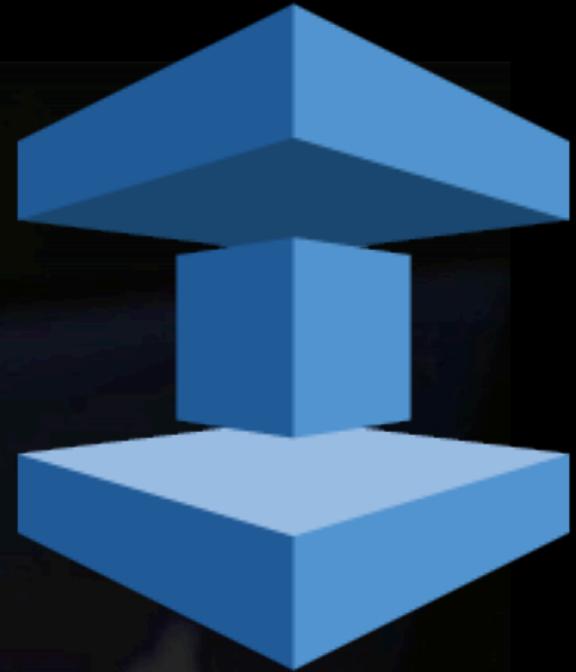
With Read Capacity Units, you are billed in blocks of 50, with Write Capacity Units you are billed in blocks of 10.

To calculate Write Capacity Units =  $(0.0065/10) \times 12 \times 24 = \$0.1872$ .

To calculate Read Capacity Units =  $(0.0065/50) \times 12 \times 24 = \$0.0374$



# Elasticache



# What is ElastiCache

ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

# What is ElastiCache

Amazon ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads (such as social networking, gaming, media sharing and Q&A portals) or compute-intensive workloads (such as a recommendation engine).

Caching improves application performance by storing critical pieces of data in memory for low-latency access. Cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations.

# Types of ElastiCache

- Memcached
  - A widely adopted memory object caching system. ElastiCache is protocol compliant with Memcached, so popular tools that you use today with existing Memcached environments will work seamlessly with the service.

# Types of ElastiCache

- Redis
  - A popular open-source in-memory key-value store that supports data structures such as sorted sets and lists. ElastiCache supports Master / Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy.

# Elasticache Exam Tips

Typically you will be given a scenario where a particular database is under a lot of stress/load. You may be asked which service you should use to alleviate this.

Elasticache is a good choice if your database is particularly read heavy and not prone to frequent changing.

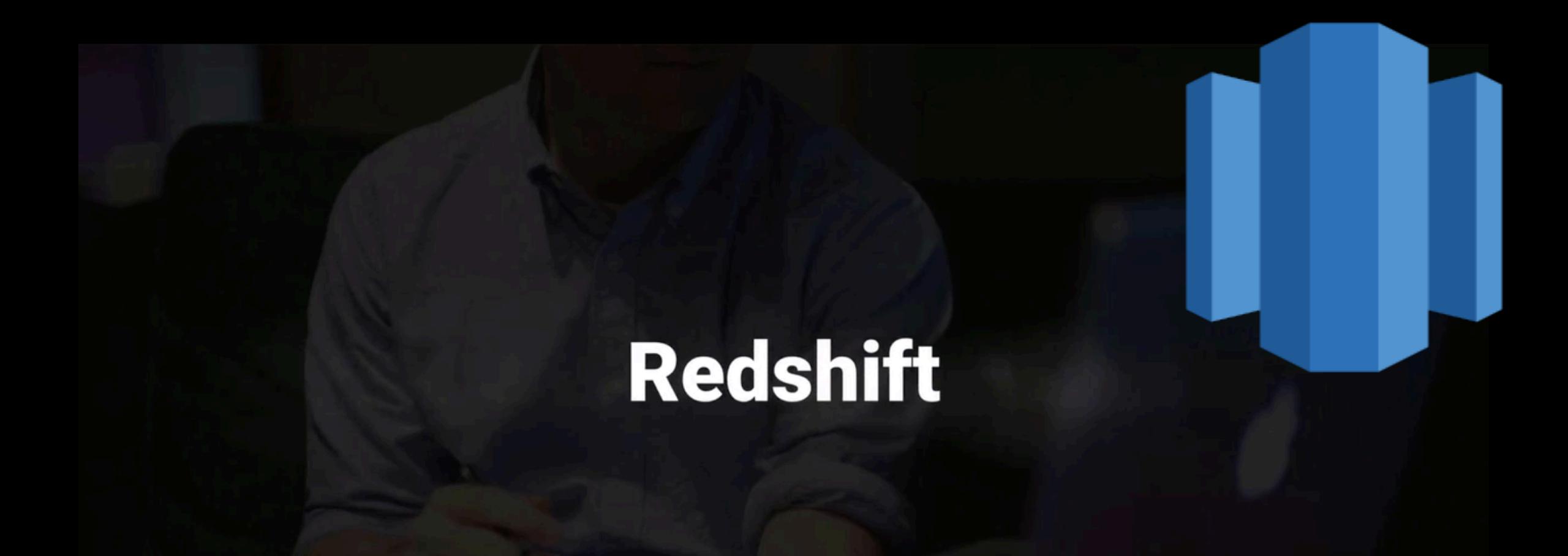
Redshift is a good answer if the reason your database is feeling stress is because management keep running OLAP transactions on it etc.

# Elasticache Exam Tips

Typically you will be given a scenario where a particular database is under a lot of stress/load. You may be asked which service you should use to alleviate this.

Elasticache is a good choice if your database is particularly read heavy and not prone to frequent changing.

Redshift is a good answer if the reason your database is feeling stress is because management keep running OLAP transactions on it etc.



A man in a grey suit jacket and patterned shirt is seated at a desk, looking down at a laptop screen. The screen displays a 3D bar chart consisting of four blue bars of increasing height. The background is dark.

# Redshift

# What is Redshift

Amazon Redshift is a fast and powerful, fully managed, petabyte-scale data warehouse service in the cloud. Customers can start small for just \$0.25 per hour with no commitments or upfront costs and scale to a petabyte or more for \$1,000 per terabyte per year, less than a tenth of most other data warehousing solutions.

# OLAP

**OLAP transaction Example:**

**Net Profit for EMEA and Pacific for the Digital Radio Product.**

**Pulls in large numbers of records**

**Sum of Radios Sold in EMEA**

**Sum of Radios Sold in Pacific**

**Unit Cost of Radio in each region**

**Sales price of each radio**

**Sales price - unit cost.**

**Data Warehousing databases use different type of architecture both from a database perspective and infrastructure layer.**

# Redshift Configuration

- Single Node (160Gb)
- Multi-Node
  - Leader Node (manages client connections and receives queries).
  - Compute Node (store data and perform queries and computations). Up to 128 Compute Nodes

## Redshift - 10 times faster

**Columnar Data Storage:** Instead of storing data as a series of rows, Amazon Redshift organizes the data by column. Unlike row-based systems, which are ideal for transaction processing, column-based systems are ideal for data warehousing and analytics, where queries often involve aggregates performed over large data sets. Since only the columns involved in the queries are processed and columnar data is stored sequentially on the storage media, column-based systems require far fewer I/Os, greatly improving query performance.

## Redshift - 10 times faster

**Advanced Compression:** Columnar data stores can be compressed much more than row-based data stores because similar data is stored sequentially on disk. Amazon Redshift employs multiple compression techniques and can often achieve significant compression relative to traditional relational data stores. In addition, Amazon Redshift doesn't require indexes or materialized views and so uses less space than traditional relational database systems.

When loading data into an empty table, Amazon Redshift automatically samples your data and selects the most appropriate compression scheme.

# Redshift - 10 times faster

**Massively Parallel Processing (MPP)**: Amazon Redshift automatically distributes data and query load across all nodes. Amazon Redshift makes it easy to add nodes to your data warehouse and enables you to maintain fast query performance as your data warehouse grows.

# Redshift Pricing

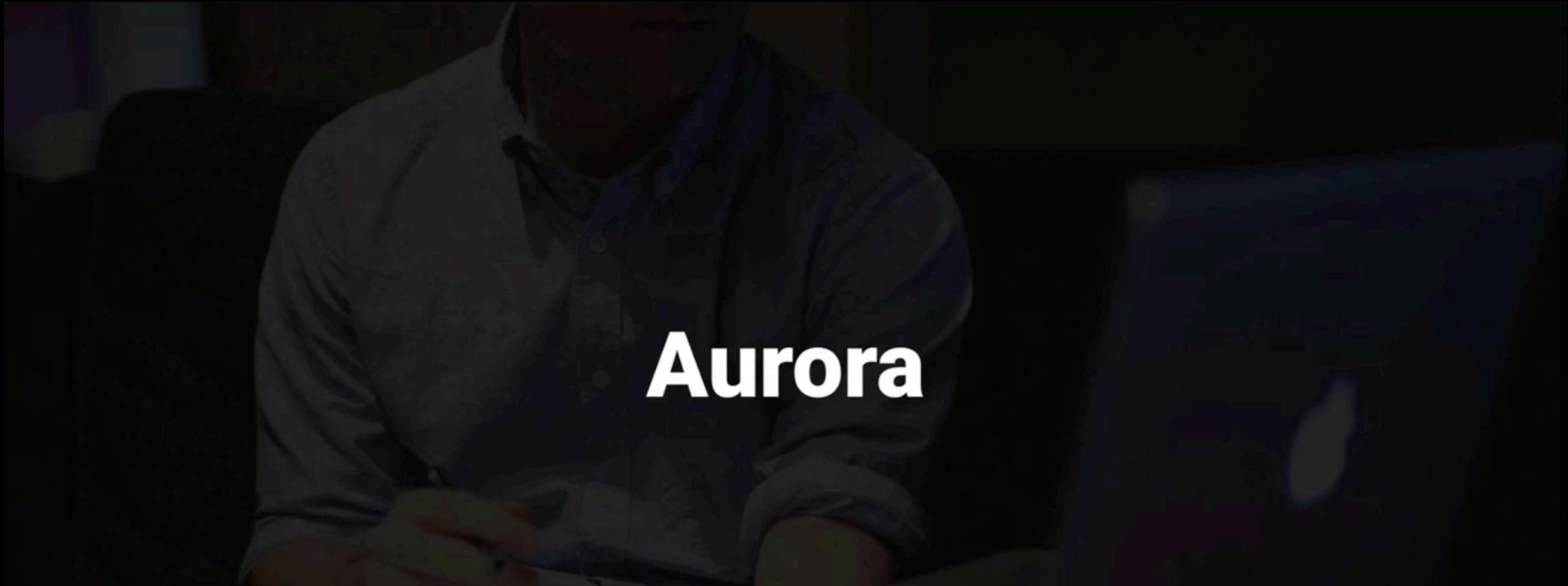
- Compute Node Hours (total number of hours you run across all your compute nodes for the billing period. You are billed for 1 unit per node per hour, so a 3-node data warehouse cluster running persistently for an entire month would incur 2,160 instance hours. You will not be charged for leader node hours; only compute nodes will incur charges.)
- Backup
- Data transfer (only within a VPC, not outside it)

# Redshift Security

- Encrypted in transit using SSL
- Encrypted at rest using AES-256 encryption
- By default RedShift takes care of key management.
  - Manage your own keys through HSM
  - AWS Key Management Service

## Redshift Availability

- Currently only available in 1 AZ
- Can restore snapshots to new AZ's in the event of an outage.



Aurora

## What is Aurora

Amazon Aurora is a MySQL-compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.

Amazon Aurora provides up to five times better performance than MySQL at a price point one tenth that of a commercial database while delivering similar performance and availability.

# Aurora Scaling

- Start with 10Gb, Scales in 10Gb increments to 64Tb (Storage Autoscaling)
- Compute resources can scale up to 32vCPUs and 244Gb of Memory.

# Aurora Scaling

- 2 copies of your data is contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.
- Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.

# Aurora Scaling

- 2 copies of your data is contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.
- Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.
- Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically.

# Aurora Replicas

- 2 Types of Replicas are available.
- Aurora Replicas (currently 15)
- MySQL Read Replicas (currently 5)

