



COSC 416: Topics in Databases (DBaaS)

TOPIC 4: AMAZON RDS FEATURES

SCHEDULE

1. Amazon RDS Features

- 1. Encryption, at-rest and in-transit*
- 2. Read replicas*
- 3. Option groups*

RDS FEATURES: ENCRYPTION

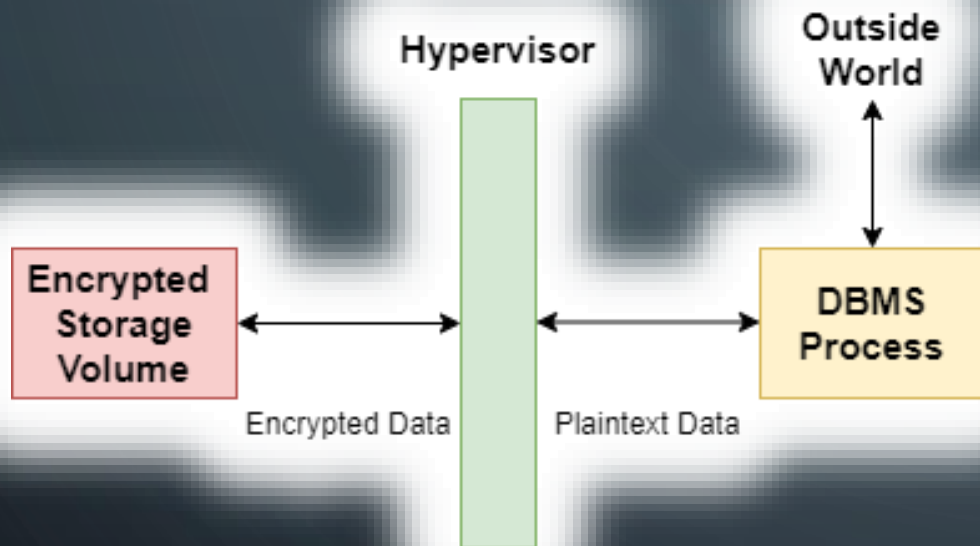
ENCRYPTION OPTIONS WITH RDS

- Amazon RDS offers a couple different options for encrypting your database, depending on your requirements:
 - Encryption of data at rest – database is stored encrypted and decrypted on the fly
 - Encryption of data in transit – securing the connection between the database and the client/application

ENCRYPTION AT REST

- Amazon offers AES-256 encryption for RDS instances
- This encryption is transparent to the database engine and pretty much invisible to the outside user
- Encryption at the whole database instance level (not encryption of individual databases, tables or columns)

HOW RDS ENCRYPTION-AT-REST WORKS



- Data in the storage volume is encrypted
- A hypervisor process performs encryption/decryption on the fly
- The DBMS process doesn't even know the disk is encrypted, so no extra configuration is required

WHY USE THIS TYPE OF ENCRYPTION?

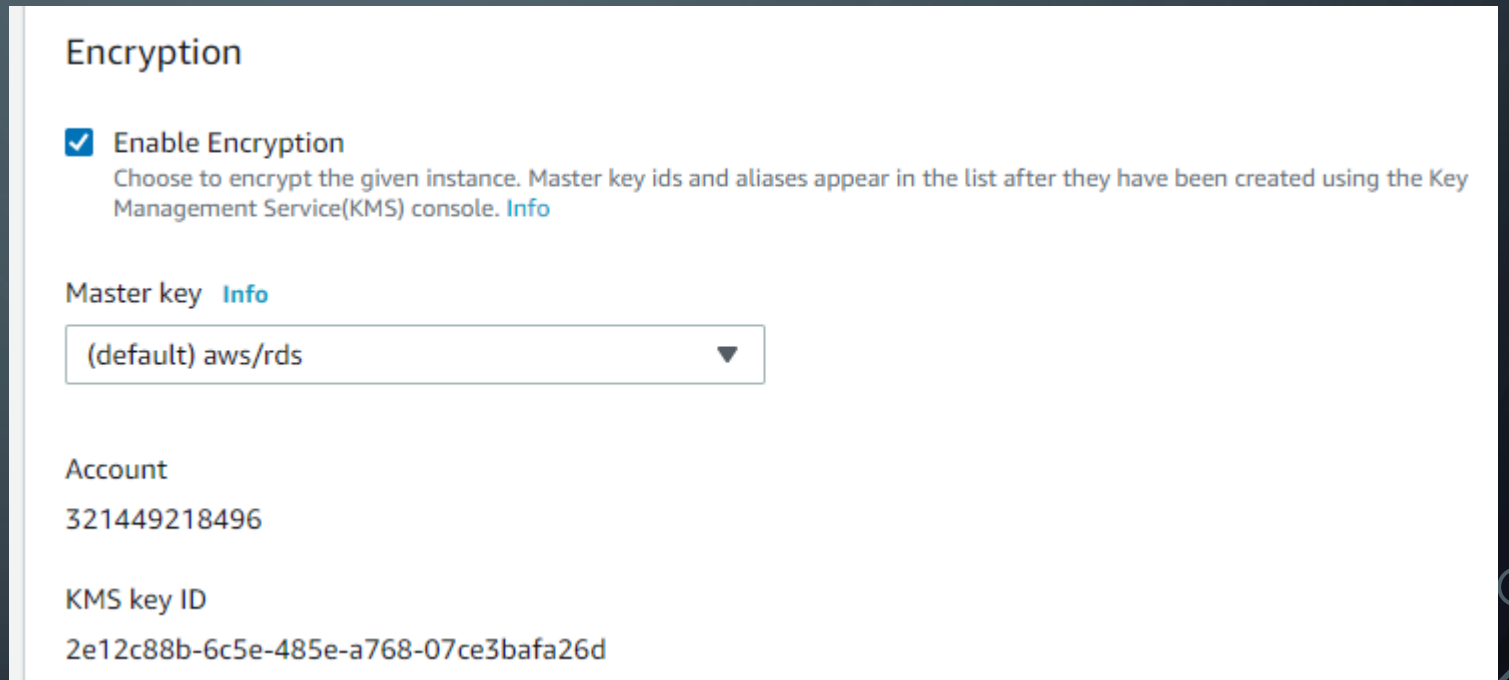
- This type of encryption primarily protects you against someone managing to access your VPC or EC2 instances and forcibly dumping storage volume contents (Or someone physically stealing Amazon's storage disks)
- It won't protect you against someone accessing your AWS account, or using your application to pull data from the database, or using the SQL terminal remotely if not secured

SOME MINOR CONCERNS

- Remember that this only encrypts data as it is read or written on the disk. Data in the memory (RAM) will be unencrypted
 - An unlikely attack, but violating memory access isn't unheard of
- Doesn't work on the db.t2.micro instance size – minimum size is the t2.small instance

CREATING AN ENCRYPTED INSTANCE

- Creating an encrypted RDS instance is as simple as picking a db.t2.small size instance in the creation menu, and then enabling encryption under the “Additional Configuration” menu



The screenshot shows the 'Encryption' configuration page in the AWS RDS console. It includes a checkbox for 'Enable Encryption' which is checked, a dropdown menu for 'Master key' set to '(default) aws/rds', and fields for 'Account' (321449218496) and 'KMS key ID' (2e12c88b-6c5e-485e-a768-07ce3bafa26d).

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 ▼

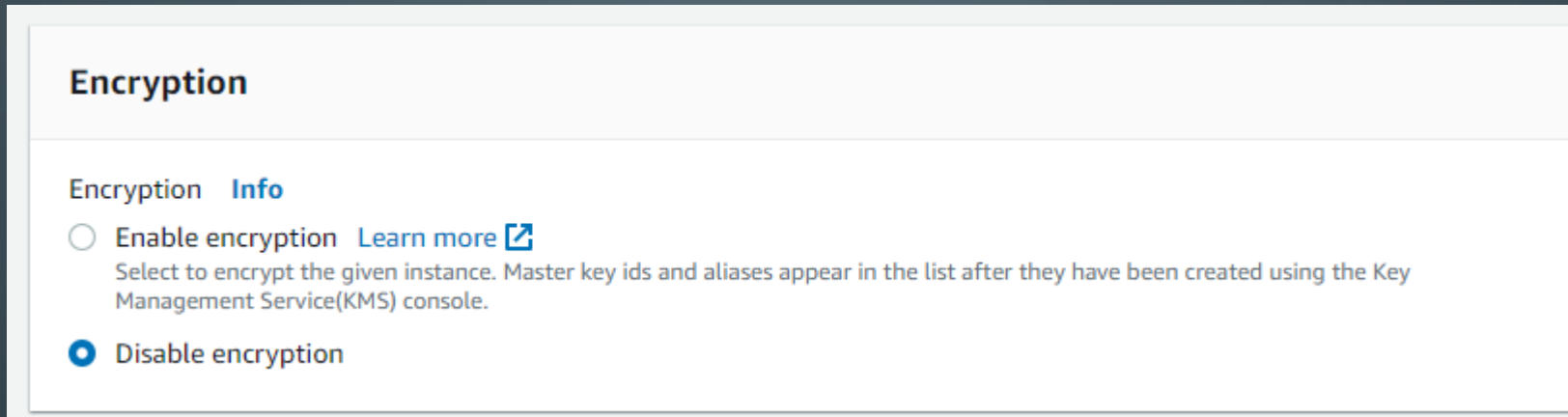
Account
321449218496

KMS key ID
2e12c88b-6c5e-485e-a768-07ce3bafa26d

WHAT ABOUT ENCRYPTING AN UNENCRYPTED DATABASE?


- Unfortunately, you can't directly encrypt a database instance that you've already created
- You can however create an encrypted snapshot copy of the database, and then restore the snapshot to create an encrypted clone of the original instance

CREATING THE ENCRYPTED COPY



Encryption

Encryption [Info](#)

☐ Enable encryption [Learn more](#) 

Select 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.

☒ Disable encryption

- First, create a snapshot of your database instance.
- Then, select it in the snapshots menu, and under the action button, click “Copy Snapshot”
- You will have the choice at the bottom of the menu to enable encryption.

CREATING THE ENCRYPTED INSTANCE

- Once the encrypted copy of the snapshot has been made, simply restore the encrypted copy
- By default, encryption will be enabled on the restored instance
- Note: It's actually not possible to create an unencrypted instance from an encrypted snapshot. So keep an unencrypted snapshot if you ever want to be able to restore an unencrypted version of the instance

KEY MANAGEMENT IN AMAZON RDS

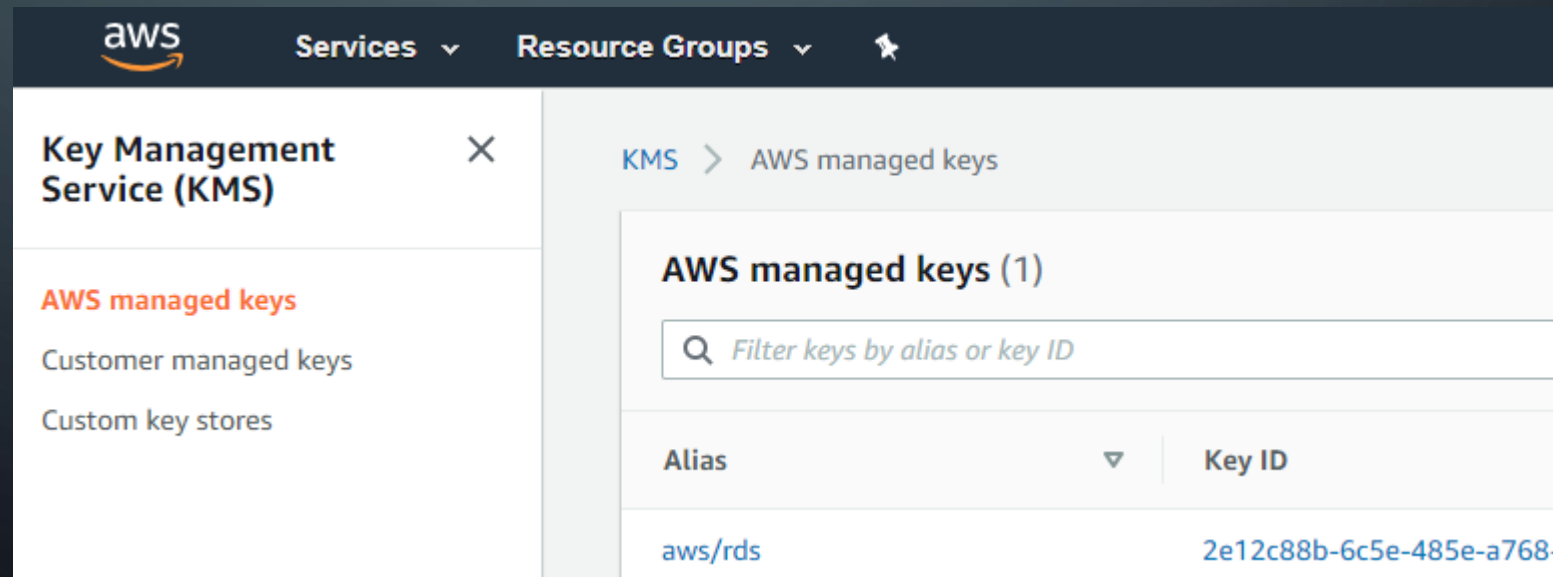
- By default, Amazon will use an automatically generated, AWS-managed key for the encryption (free)
- An alternative is to use Customer Managed Keys (CMKs) which allow you a greater level of control (\$1 /month per key)

CUSTOMER MANAGED KEYS

- The AWS Key Management Service allows you store and use Customer Managed Keys
- You can use your own key generation inputs and storage, handle access control and key privileges, enable/disable keys, and audit key usage, among other things
- In comparison, we can't really do that with the free AWS-managed key, but it's easy and doesn't cost us anything

VIEWING YOUR KEYS

- You can view the keys (both AWS managed and CMK) by navigating to the Key Management Service (KMS)



ALRIGHT, WHAT ABOUT IN TRANSIT?

- If we're dealing with any kind of sensitive data, we'd probably want to make sure it is encrypted in transit, so that even if it is intercepted, it can't be read
- We can encrypt our database connections to the RDS instance using SSL/TLS, the same protocols used for HTTPS (secure WWW access)

USING SSL/TLS FOR IN-TRANSIT ENCRYPTION

- When the db instance is provisioned, Amazon will automatically generate an SSL/TLS certificate for it
- This SSL/TLS certificate can be used to open a secure connection to the MySQL shell via the command line argument `–ssl-ca=[certificate bundle here] –ssl-mode=VERIFY_IDENTITY`

WHAT DOES THIS DO?

- By using an SSL connection, data will be encrypted via a symmetric key before transmission and decrypted on arrival, in both directions
- Using an SSL secured connection prevents someone from intercepting your traffic and reading it
- Using the identity verification option also protects against someone trying to spoof the RDS service

OTHER OPTIONS

- Since we don't have access to the underlying OS on our RDS instance, some of our security options are limited
- In addition, RDS doesn't support the InnoDB engine tablespace encryption for MySQL
- Can always encrypt data at the client/application level and store the encrypted data in the database instead (Even if someone gains access to your RDS instance, unless they have the key the data is useless)

RDS ENCRYPTION IN A NUTSHELL

- Both at-rest and in-transit data can be encrypted free of charge using the KMS service and SSL/TLS connections
- For really secure data, encryption at the client/application level may be more appropriate
- Ultimately, it will most likely be the security of your application that causes problems, rather than direct access to the RDS instance

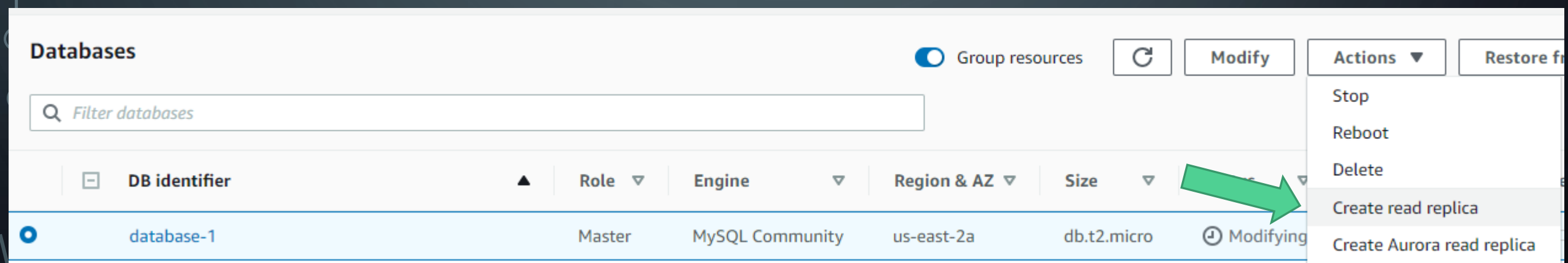
RDS FEATURES: READ REPLICAS

READ REPLICAS – EASY TRAFFIC MANAGEMENT

- RDS offers the ability to create *Read Replicas* for MariaDB, MySQL, Oracle, and PostgreSQL
- Read Replicas are read-only copies of a normal RDS instance
- When changes are made to the underlying RDS instance (like inserting a record), this change is asynchronously replicated over to the read replica

CREATING A READ REPLICA

- To create a read replica, simply select the database instance you wish to create a replica of, and under actions select “Create read replica”



The screenshot shows the AWS Management Console interface for the 'Databases' section. At the top, there is a search bar labeled 'Filter databases'. Below it, a table lists database instances. The first instance, 'database-1', is selected. The 'Actions' dropdown menu is open, showing options: 'Stop', 'Reboot', 'Delete', 'Create read replica', and 'Create Aurora read replica'. A green arrow points to the 'Create read replica' option.

Databases Group resources Refresh Modify Actions ▼ Restore from backup

DB identifier	Role	Engine	Region & AZ	Size	Instance state
database-1	Master	MySQL Community	us-east-2a	db.t2.micro	Modifying

- Stop
- Reboot
- Delete
- Create read replica
- Create Aurora read replica

USING READ REPLICAS

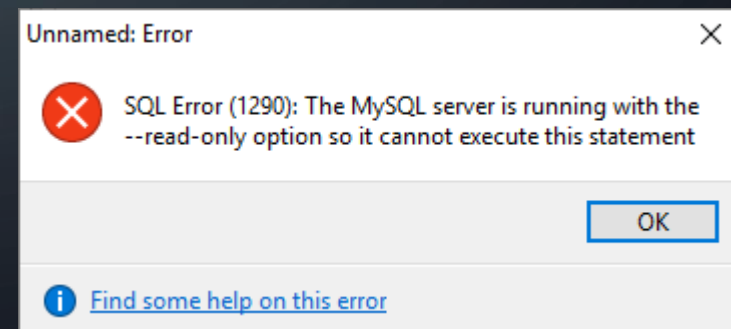
- Read replicas only support read-only connections, however this is fine if all we need to do is read from the database
- Idea: route read (query) traffic to the read replica, and route updates/deletions/insertions to the original database instance
- Reduce overall load on the main instance

ADDITIONAL USES FOR READ REPLICAS

- Read replicas can be created across multiple regions
- If you're trying to provide geographically close connections, you could use the read replicas to provide instances nearby
- Read replicas can also be sized individually based on standard RDS sizes – multiple micro read-replicas versus fewer larger read-replicas

INTERFACING WITH READ-REPLICAS

- The replicas behave exactly like a standard database instance, with the exception that all sessions have the `--read-only` option enabled
- Attempting to perform a write action when connected to one will return an SQL error 1290



A NOTE ABOUT REPLICAS AND ENCRYPTION

- Similar to how you cannot create an unencrypted instance from an encrypted snapshot, an encrypted instance will always create encrypted read replicas
- As with regular database instances, the encryption is transparent and won't interfere with the operation of the read replica

ASYNCHRONOUS REPLICATION

- There is one issue we need to consider with a read replica though: the replication lag between the database instance and the replica
- If we access the read replica, we aren't necessarily getting the most-up-to-date version of the database
- You can also purposefully set a minimum replication delay time, so that the replicas are always at least X period of time stale (potentially good for immediate backups?)

IN SHORT

- Read replicas are simple, easy to use, and fairly straightforward
- They require minimal intervention on your part – you can just direct read-only traffic to them like you would any other database server
- Some minor caveats – potentially stale data, long creation time, read-only obviously

The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

RDS FEATURES: OPTIONS GROUPS

OPTION GROUPS IN RDS

- Option groups provide a means of enabling or adjusting additional options for a database instance
- You can share an option group across multiple instances, allowing you to duplicate a particular settings profile easily

OPTIONS FOR MYSQL

- Each DBMS offered by RDS has different options available for it's option group
- For MySQL, the only options supported are:
 - MariaDB Audit Plugin (database activity logging)
 - MySQL memcached (caching system for MySQL)
- New options groups can be created by navigating to the Options group section of the RDS dashboard

ADDING OPTIONS

- Once you have created a new option group, you can select it and click the “Add option” button on the Option groups page
- Selecting an option from the dropdown will reveal additional option settings
- You can then add the option to your option group

Option details

Option group name

memcached-enabled

Option

Name of Option you want to add to this group

Choose an option

Apply Immediately

[Info](#)

☐ Yes

☒ No

ENABLING OPTIONS ON AN INSTANCE

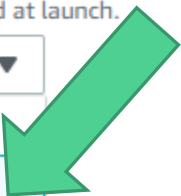
- Options can be enabled on a new instance, or on a previously created instance by modifying it
- In the Database options section, simply select your new Option group from the dropdown

Database options

Database port
Specify the TCP/IP port that the DB instance will use for application connections. The connection string of any application connecting to the DB instance must specify the port number of the DB instance. Both the security group applied to the DB instance and your company's firewalls must allow connections to the port. [Learn More](#)

DB parameter group
Database parameter group to associate with this DB instance

Option group
Name of an option group that contains options (e.g. Memcached, Oracle Enterprise Manager) you want attached to this DB instance. If there aren't any option groups compatible with the selected engine, a default option group will be created at launch.



Manage your database user credentials through AWS IAM users and roles.

☒ Disable

OPTION GROUPS – USEFUL?

- The option groups allow you to easily enable the MariaDB auditing plugin, or the memcached plugin, but not much else
- Very limited option selection for all databases except for Oracle (although MSSQL allows you to enable Transparent Data Encryption, an MS-specific encryption option separate from RDS encryption)

SCHEDULE

1. Amazon RDS Features

- 1. Encryption, at-rest and in-transit*
- 2. Read replicas*
- 3. Option groups*

The image features a dark blue background with a subtle radial gradient. In the four corners, there are decorative white line art elements resembling circuit traces or a stylized network. These lines connect to small white circles, some of which are arranged in a grid-like pattern. The central text is a large, white, sans-serif phrase.

SO LONG, FOLKS!