MyISAM Tables in MySQL RDS Instances

Before migrating a MySQL database to AWS Relational Database Service, be sure to understand the impact of that move if any tables in your MySQL database utilize the MyISAM engine. AWS repeats this warning in many places:

> *Amazon RDS automated backups and DB snapshots are currently supported for all DB engines. For the MySQL DB engine, only the InnoDB storage engine is supported; use of these features with other MySQL storage engines, including MyISAM, may lead to unreliable behavior while restoring from backups. Specifically, since storage engines like MyISAM do not support reliable crash recovery, your tables can be corrupted in the event of a crash. For this reason, we encourage you to use the InnoDB storage engine.*
>
> *[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.BackingUpAndRestoringAmazonRDSInstances.html#Overview.BackupDeviceRestrictions]*

Determine if your database contains MyISAM tables

Use this query to list the MyISAM tables in your database. As the root user, run the following query:

```
SELECT TABLE_SCHEMA, TABLE_NAME, ENGINE FROM INFORMATION_SCHEMA.TABLES WHERE ENGINE = 'MyISAM';
```

Solution Options

1. Convert MyISAM tables to InnoDB tables.

- This solution is best if you can do it, but there may be reasons you cannot. E.g., vendor support requires that you don't meddle with the database configuration used by their software.
- Understand the impact of making the change; see http://stackoverflow.com/questions/3818759/what-is-innodb-and-myisam-in-mysql.
- If you can swing it, the conversion is easy:

```
ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;
```

2. Be ready to repair MyISAM tables in the event of a crash. Be ready for the potential of data corruption or loss.

> *If you choose to use MyISAM, you can attempt to manually repair tables that become damaged after a crash by using the REPAIR command (see: http://dev.mysql.com/doc/refman/5.5/en/repair-table.html). However, as noted in the MySQL documentation, there is a good chance that you will not be able to recover all your data.*
>
> *[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.BackingUpAndRestoringAmazonRDSInstances.html#Overview.BackupDeviceRestrictions]*

3. Try to ensure that you get good snapshots.

AWS outlines the following process to endeavour to get good snapshots, if your MySQL RDS DB has MyISAM tables.

1. Stop all activity to your MyISAM tables (that is, close all sessions). You can close all sessions by calling the mysql.rds_kill command for each process that is returned from a MySQL SHOW FULL PROCESSLIST command.
2. Lock and flush each of your MyISAM tables. For example, the following command locks and flush two tables named myisam_table1 and myisam_table2:
    - `FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;`
3. Create a snapshot of the RDS instance.
4. Release the locks on the tables.
    - `UNLOCK TABLES;`

The lock-and-snap.rb script in https://github.com/CU-CommunityApps/docker-mysql-util attempts to follow this procedure to lock-and-flush MyISAM tables before creating and RDS snapshot.

We are in the process of understanding this process in greater detail.

Be aware that your database may have MyISAM tables in the internal schemas MySQL uses for operations. Initial trials suggest that flushing-and-locking those tables along with your own tables will block the snapshot from beginning (or at least completing). Snapshot creation times seem nominal if only your own MyISAM tables are flushed-and-locked, leaving out the internal MyISAM tables.

Read Replicas

- Making snapshots from Read Replicas may also be in the solution mix. However, be aware that creating a new read replica is based on a snapshot, so getting a good snapshot with stable/valid MyISAM table also may be problematic.
- RDS documentation warns to be sure to monitor the ReplicaLag metric if you have read replicas setup for MySQL RDS DBs containing MyISAM tables. In particular it warns that if the ReplicaLag metric returns -1, then replication may be having problems because of MyISAM tables.
    - From AWS: "Using a non-transactional storage engine such as MyISAM. Replication is only supported for the InnoDB storage engine."
    - However, AWS might really mean that "**Reliable** replication is only supported for the InnoDB storage engine." Empirical tests show that data in MyISAM tables is replicated though we haven't tested the reliability of that.

Resources

- Amazon Relational Database Service User Guide. This document has a fair amount to say about MyISAM tables–mostly warnings.
- AWS RDS FAQs. Search for MyISAM on the page)