AWS Runbook   
RDS Encryption

Author: Cloudvisor

Date: 21/12/2023

[**1. Introduction 1**](#_heading=h.snns71p77pfe)

[**1.1 Target Audience 1**](#_heading=h.1r2fn7w5se2)

[**2. Scope 2**](#_heading=h.2dxmb87rv46m)

[**AWS resources covered 2**](#_heading=h.71lfkqlnzrrz)

[**Exclusions and Limitations 2**](#_heading=h.g4sp7pjyt8wm)

[**3. Prerequisites 2**](#_heading=h.5frs0vetan4)

[**4. Procedures 3**](#_heading=h.bj6b2m8nqebe)

[**5. Emergency Contacts 5**](#_heading=h.k70kjhkppdqc)

[**6. Troubleshooting 5**](#_heading=h.h98omjx0az1y)

[**7. Appendices 5**](#_heading=h.fqumo4j7jlp0)

[**Things to be aware of 5**](#_heading=h.32ak3oj0lrep)

[**Useful resources 5**](#_heading=h.248g1i6lyk3d)

## 1. Introduction

The primary purpose of this runbook is to provide a comprehensive, step-by-step guide on securely encrypting an existing unencrypted Amazon Relational Database Service (RDS) instance. Encrypting your RDS instance enhances data security by protecting your data at rest, a crucial requirement for compliance with various data protection regulations and standards. This process is vital for organizations that handle sensitive information and seek to bolster their data security measures.

Goals and objectives:  
 - secure data at rest

- compliance and standards

- best practices implementation

### 1.1 Target Audience

This runbook is intended for AWS administrators, database administrators, and IT professionals responsible for managing and securing database instances on AWS. It assumes a basic understanding of Amazon RDS and AWS services.

## 2. Scope

### AWS resources covered

* RDS
  + This is the primary focus of this runbook. It includes different database engines supported by RDS, such as MySQL, PostgreSQL, Oracle, SQL Server and MariaDB
  + it covers all instance types available in RDS
  + Addresses both single-AZ and multi-AZ deployment model
* KMS
  + This encryption process utilizes AWS KMS to create and manage the encryption keys used to encrypt the RDS instance.
* Other AWS Services related
  + EC2: while not directly involved in the encryption process, the runbook acknowledges the role of EC2 instances that may interact with the RDS instance
  + CloudWatch: Monitoring and logging is not affected by the encryption process

### Exclusions and Limitations

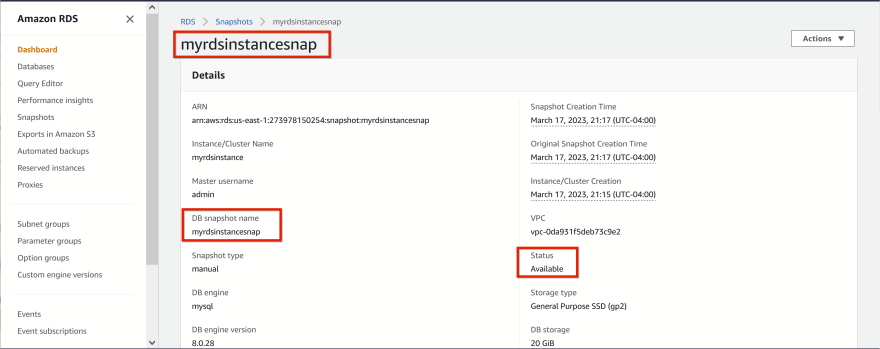
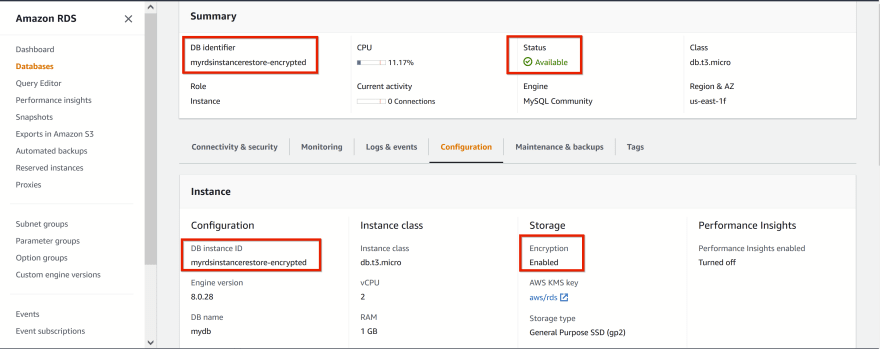
* Non-RDS Databases: this runbook does not cover encryption processes for databases not managed by RDS (like self-managed databases on EC2 or other cloud services)
* In-transit Encryption: the focus is on at-rest encryption thus encrypting data in transit is outside the scope of this document
* Cross-region encryption: the process described is limited to the encryption within the same AWS region and does not cover cross-region RDS instance encryption scenarios

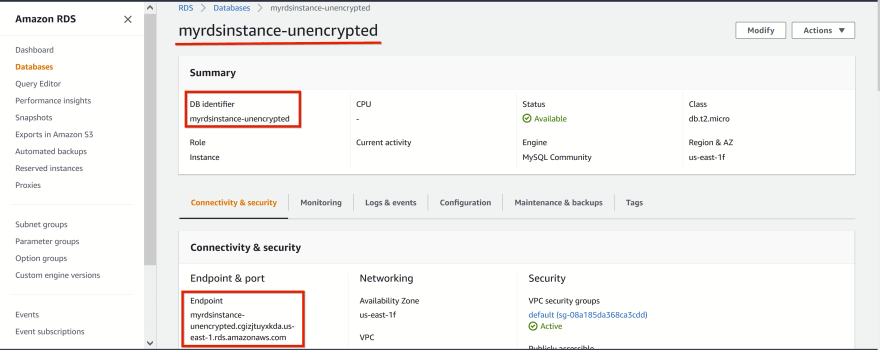
## 3. Prerequisites

Before proceeding with the encryption process, certain prerequisites must be met to ensure a smooth and successful implementation. This section outlines the necessary requirements to use this runbook effectively.

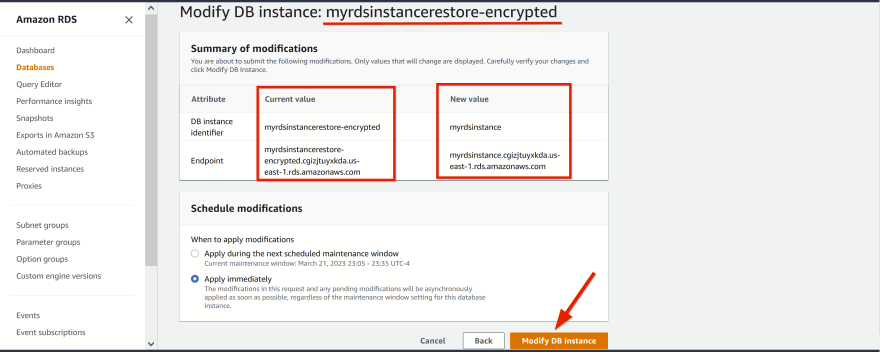
1. Account and RDS Instance details: make sure you have access to the account in which you want to perform this action. Also make sure you know which RDS instance you want to perform against the encryption process
2. Required Permissions: make sure you have the required IAM permissions to execute these actions, some permissions worth mentioning here are: rds:ModifyDBInstance, rds:RebootDBInstance, and kms:CreateKey but this is not the full list. Contact your AWS Administrator if you get any permissions errors
3. Backup and Recovery Plan: ensure that a recent backup of the RDS instance is available. This is crucial for data protection in case of any issues during the encryption process
4. Recovery Strategy: have a recovery strategy in place, including understanding how to restore from backups in case the encryption process encounters issues
5. Understanding of the RDS and KMS services

## 4. Procedures

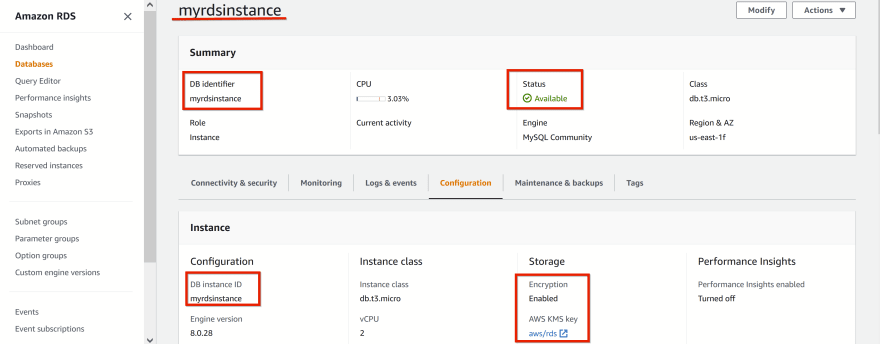
1. Signin to the AWS Management console
2. Create a snapshot of the unencrypted RDS instance and validate the creation of the snapshot  
   
3. Copy the snapshot and enable encryption. Name it something like mydbinstancesnap-encrypted and wait to become available.  
   
4. Restore the DB Instance from the encrypted snapshot, name it something like myrdsinstancerestore-encrypted with the same configuration like the original database (make sure it is in the same VPC, subnet group, have the same security group etc.)  
   
5. Change the name of the original RDS instance, something like myrdsinstance-unencrypted and “Apply immediately”



1. Change the name of the encrypted RDS instance, to the same name that the unencrypted instance had before (this will make sure you have the same endpoint for your application to connect to). Make sure you “Apply immediately” the changes.



1. Check everything is ok from the application point of view and that there is a connection to the database



1. Once everything is ok, you can remove the unencrypted original database

This process can take quite some time depending on the size of your database, so make sure you acknowledge the fact that there will be some downtime to your application while the endpoint is changing.

## 5. Emergency Contacts

Provide contact information for key personnel or teams in case of emergencies.

## 6. Troubleshooting

Offer common troubleshooting tips or links to relevant AWS documentation.

## 7. Appendices

### Things to be aware of

Some instance types does not support encryption-at-rest such as db.t2 instance family

### Useful resources

[What is Amazon Relational Database Service (Amazon RDS)?](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html)

[Encrypting Amazon RDS resources](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html)

[Backing up and restoring a DB instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BackupRestore_Instance.html)