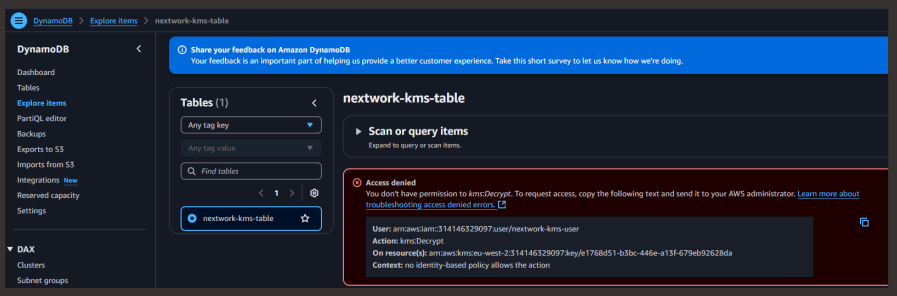
nextwork.org

Encrypt Data with AWS KMS

GA gazzok@gmail.com



GA
gazzok@gmail.com
NextWork Student
NextWork.org

Introducing Today's Project!

In this project, I will demonstrate the use of KMS keys to encrypt a database. The goal is to protect the data of the database through the use of granting permission to users for KMS keys.

Tools and concepts

Services I used include DynamoDB, IAM and AWS KMS. Key concepts I learnt include encryption, permission access and database creation.

Project reflection

This project took me approximately 1.5 hours. It was rewarding to see the power of AWS KMS keys on securing data in a database.

I did this project to inmprove my abilities on security in AWS. This was a great project that definitely met my expectations.

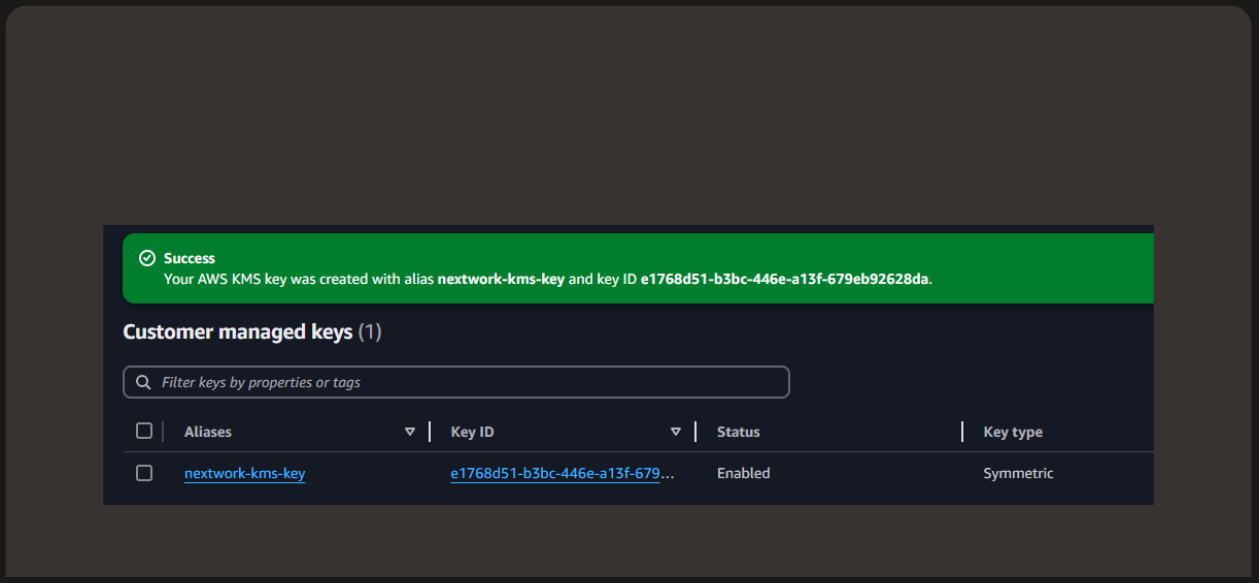
GA
gazzok@gmail.com
NextWork Student

Encryption and KMS

Encryption is the process of using algorithms to convert data into a secure format called ciphertext. Companies and developers do this to protect your data. Encryption keys are essentailly decoders to get the encrypted cyphertext back to normal.

AWS Key Management Service (KMS) is a secure vault for your encryption keys. Key management systems are important because it helps you maage all your keys in one place. KMS has the benefit of also giving you logs on every time a key is used.

Encryption keys are broadly categorized as the algorithm that will decrypt & encrypt your data. I set up a symmetric key because they are generally faster and more efficient for encrypting large amounts of data (and I'm not sharing the key).

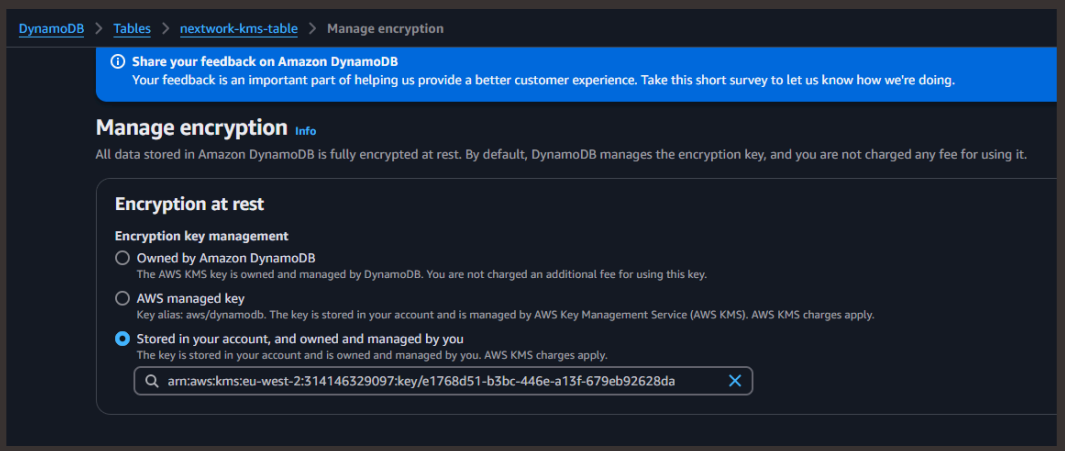


GA
gazzok@gmail.com
NextWork Student

Encrypting Data

My encryption key will safeguard data in DynamoDB, which is at rest

The different encryption options in DynamoDB include Owned by Amazon DynamoDB, AWS managed key and Stored in your account, and owned and managed by you. Their differences are based on ownership & management of the keys.

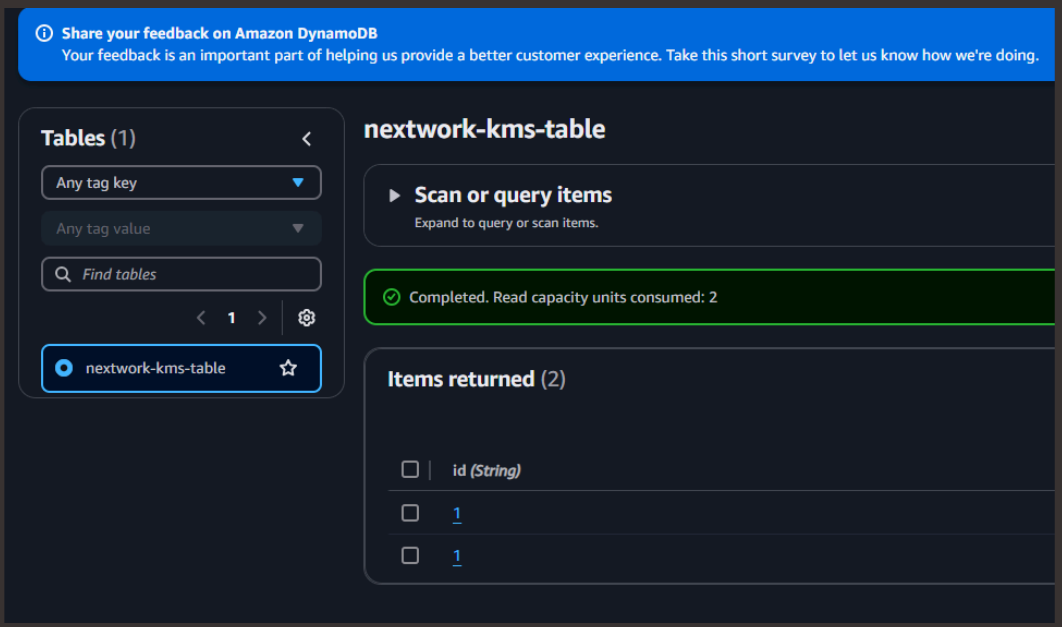


GA
gazzok@gmail.com
NextWork Student

Data Visibility

Rather than controlling who has access to the key, KMS manages user permissions by who has been granted the permission to use the key.

Despite encrypting my DynamoDB table, I could still see the table's items because my IAM user has permissions to use the encryption key in KMS. DynamoDB uses transparent data encryption, which means an authorized user will see it in a decryted format

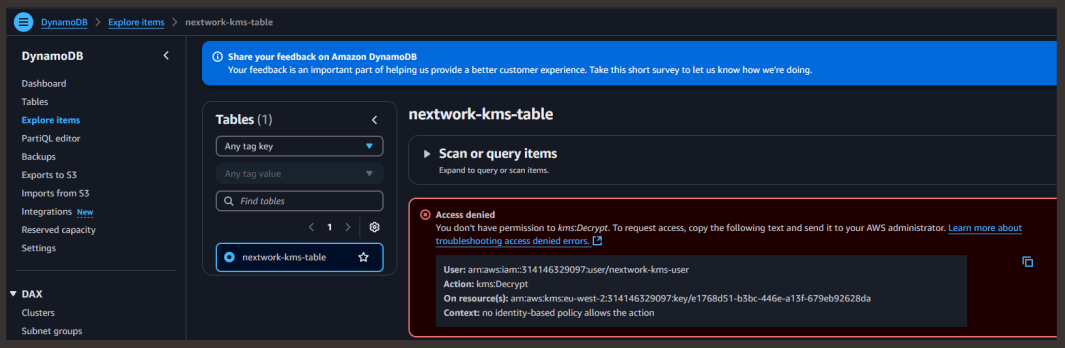


GA
gazzok@gmail.com
NextWork Student

Denying Access

I configured a new IAM user to verify our encryption using KMS is working correctly. The permission policies I granted this user are full access to DynamoDB, but not to my KMS key.

After accessing the DynamoDB table as the test user, I encountered an 'Access Denied' error because this user does not have KMS permissions and cannot decrypt the data in my DynamoDB table. This confirmed my KMS key is working how it should.

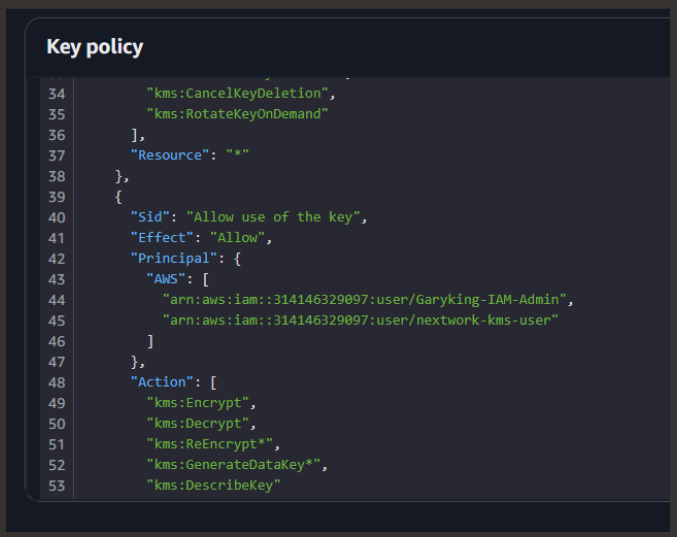


EXTRA: Granting Access

To let my test user use the encryption key, I granted the user permission to use the KMS key. My key's policy was updated to allow the new user permission to use it as seen in the code.

Using the test user, I retried accessing the DynamoDB table. I observed that the operation was completed this time which confirmed that adding the permission to use the KMS key has worked.

Encryption secures data instead of access to a resource (e.g. a DynamoDB table) or an entire service (e.g. DynamoDB). I could combine encryption with the afore mentioned measures to further limit access and keep our data & resources secure.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects