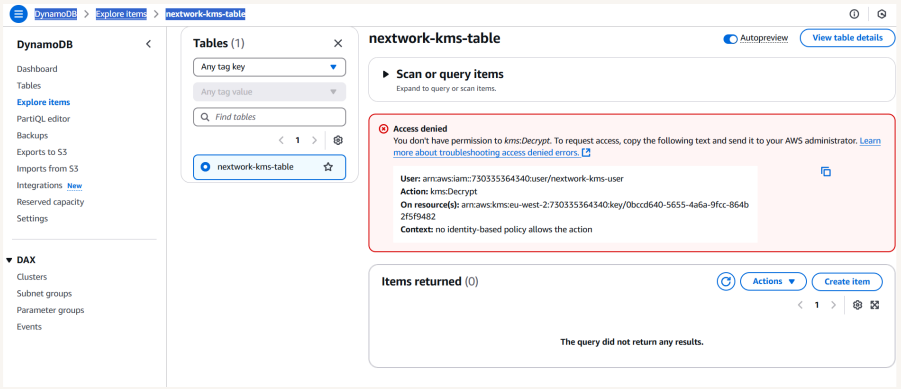
nextwork.org

Encrypt Data with AWS KMS

MA Marzena Pugo



Introducing Today's Project!

In this project, I will demonstrate using encryption to secure data. The goal is to create encryption keys with AWS KMS (Key Management System), encrypt a DynamoDB Table's data with that key, then test access using IAM users.

Tools and concepts

Services I used include AWS KMS (Key Management Service), DynamoDB, AWS IAM. Key concepts I learnt include encryption, database tables, KM using permissions to actions rather than just access to the key itself; creating a user to test access.

Project reflection

This project took me approximately 1.5 hours. The most challenging part was understanding how encryption works differently from other access controls. It was most rewarding to see my test user get access to encryption.

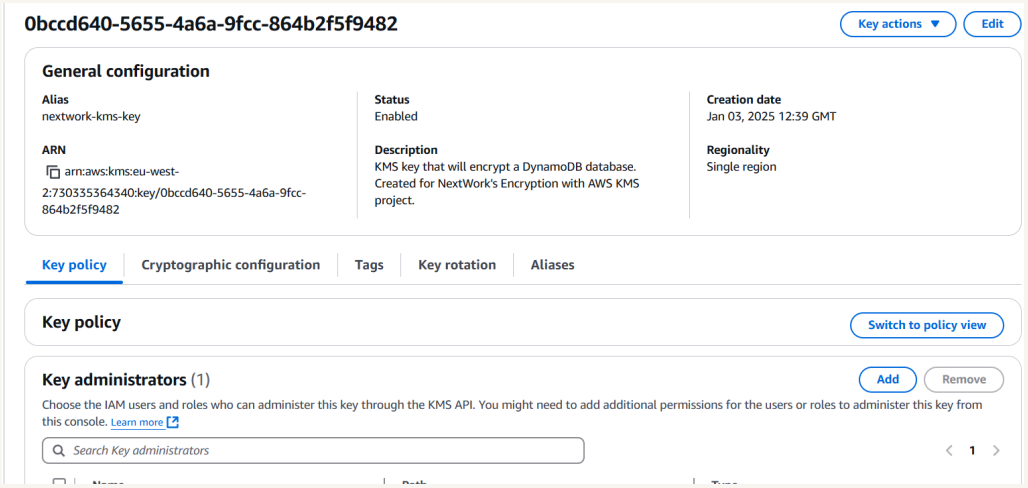
I did this project to learn all about encryption, securing data and how it actually works. This project showed me the foundations of encryption keys and managing access as an admin.

Encryption and KMS

Encryption is the process of turning original/plaintext data into a secure format. Companies and developers do this to secure their data from unauthorised users. Encryption keys are the secure code that informs an algorithm how it should encrypt.

AWS KMS is a vault for encryption keys. Key management services (KMS) are important because they help me secure and manage the keys I use to encrypt data. Unauthorised access to the key = exposing my encrypted data whch puts my secrity at risk.

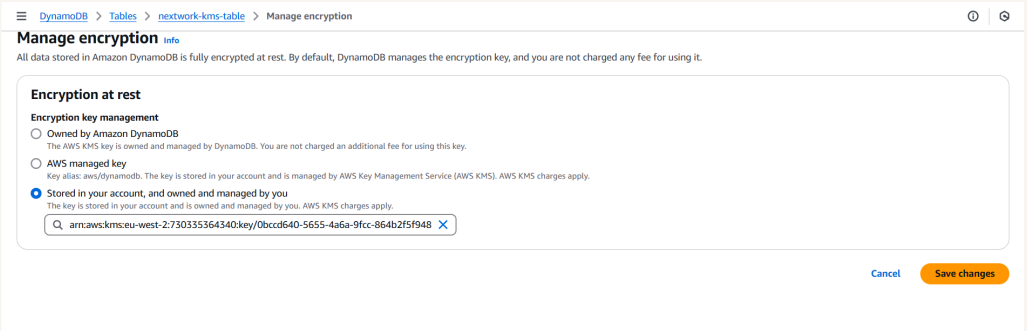
Encryption keys are broadly categorised as symmetric and asymmetric. I set up a symmetric key because I will be using the exact same key to encrypt and decrypt my data.


0bccd640-5655-4a6a-9fcc-864b2f5f9482
Key actions
Edit
General configuration
Alias
nextwork-kms-key
Status
Enabled
Creation date
Jan 03, 2025 12:39 GMT
ARN
arn:aws:kms:eu-west-2:730335364340:key/0bccd640-5655-4a6a-9fcc-864b2f5f9482
Description
KMS key that will encrypt a DynamoDB database. Created for NextWork's Encryption with AWS KMS project.
Regionality
Single region
Key policy
Cryptographic configuration
Tags
Key rotation
Aliases
Key policy
Switch to policy view
Key administrators (1)
Add
Remove
Choose the IAM users and roles who can administer this key through the KMS API. You might need to add additional permissions for the users or roles to administer this key from this console. Learn more
Search Key administrators
1

Encrypting Data

My encryption key will safeguard data in DynamoDB, which is a fast and flexible AWS Database service. DynamoDB is great for applications that need fast access to large amount of data e.g. gaming

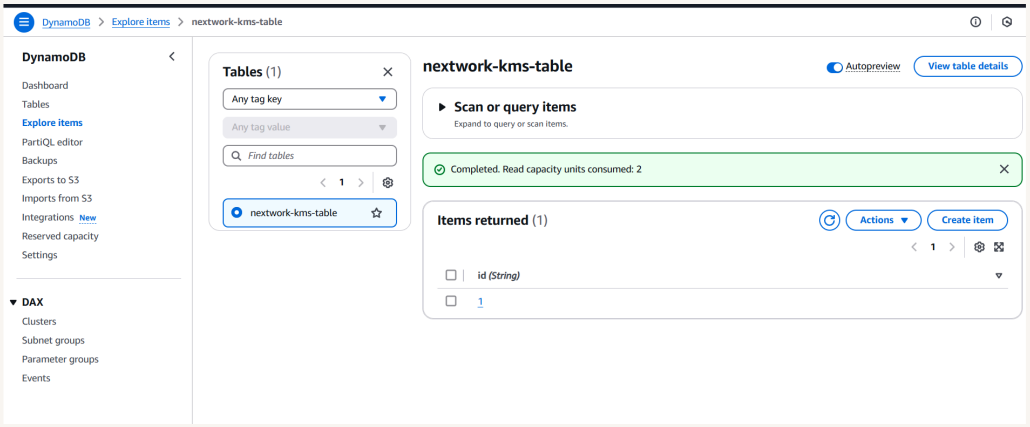
The different encryption options in DynamoDB include DynamoDB owned, AWS managed and customer managed (CMK). Their differences are based on who creates and manages the key; and whether I have visibility. I selected CMK to use my created KMS key.



Data Visibility

Rather than controlling who has access to the key, KMS manages user permissions by controlling the actions that people can do with that key. In my case, even if I gave my test user the permission to see the key, it would need permission to decrypt.

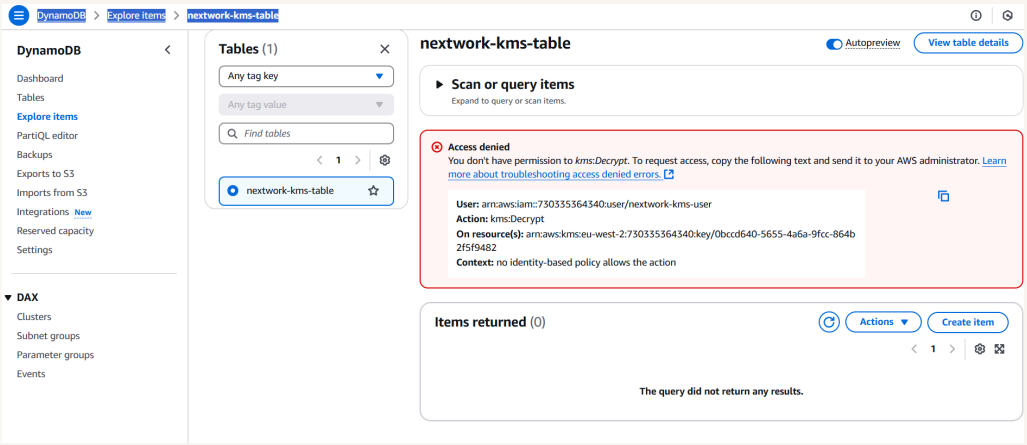
Despite encrypting my DynamoDB table, I could still see the table's items because I am user of the key. DynamoDB uses transparent data encryption, which means it does the encryption/decryption process for me because it knows I am authorised.



Denying Access

I configured a new IAM user to validate whether unauthorised users can access encrypted data. The permission policies I granted this user are DynamoDB full access but not encryption/decryption permission with AWS KMS.

After accessing the DynamoDB table as the test user, I encountered an accesss denied message because my test user has no access to decryption with the key. This confirmed that encryption keys can be used to secure data.

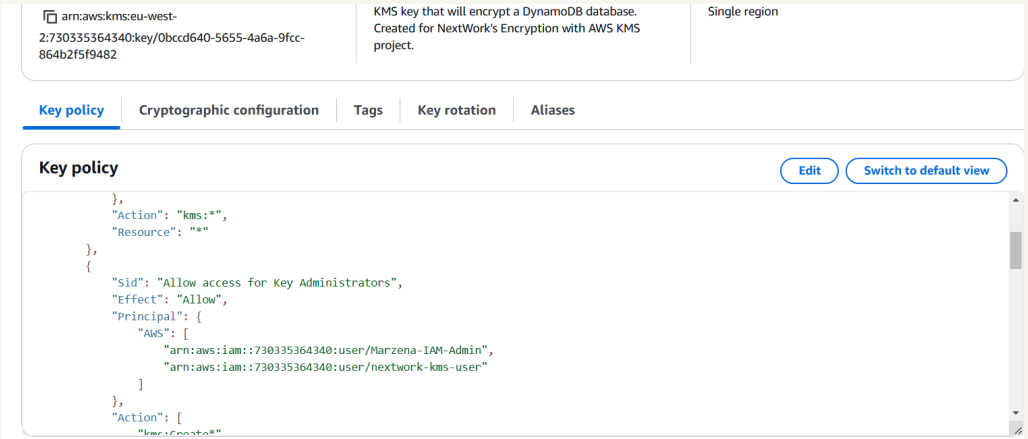


EXTRA: Granting Access

To let my test user use the encryption key, I made it a key user in the KMS console. My key's policy was updated to allow nextwork-kms-user to encrypt, decrypt, re-encrypt etc using the key.

Using the test user, I retried accessing the DynamoDB table. I observed which confirmed that the user can see the data inside again, which confirmed that making it a key user is an effective way to authorise someone to see encrypted data.

Encryption secures data instead of an entire resource or service. I could combine encryption with other access control tools like security groups and permission policies to have two layers of security -the resource level and then the data level.


arn:aws:kms:eu-west-2:730335364340:key/0bccd640-5655-4a6a-9fcc-864b2f5f9482
KMS key that will encrypt a DynamoDB database. Created for NextWork's Encryption with AWS KMS project.
Single region
Key policy
Cryptographic configuration
Tags
Key rotation
Aliases
Key policy
Edit
Switch to default view
},
"Action": "kms:*",
"Resource": "*"
},
{
"Sid": "Allow access for Key Administrators",
"Effect": "Allow",
"Principal": {
"AWS": [
"arn:aws:iam::730335364340:user/Marzena-IAM-Admin",
"arn:aws:iam::730335364340:user/nextwork-kms-user"
]
},
"Action": [

The place to learn & showcase your skills

Check out nextwork.org for more projects