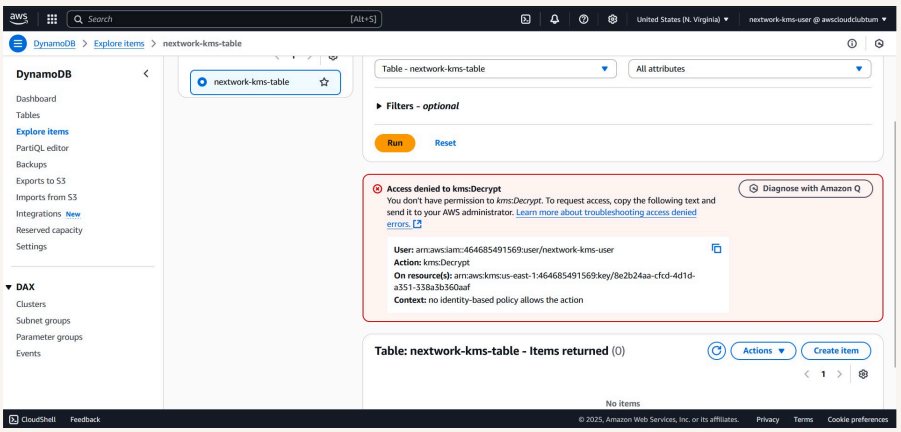
nextwork.org

Encrypt Data with AWS KMS

Chrispinus Jacob



Chrispinus Jacob
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use encryption to secure data in AWS. The goal is to create an encryption key with AWS Key Management Service (KMS), encrypt a DynamoDB table's data using that key, and test access with an IAM user to ensure that only authorized users can decrypt and access the secured data.

Tools and concepts

Services I used include AWS KMS, AWS IAM, and Amazon DynamoDB. Key concepts I learnt include encryption, key management, IAM policies, and how to combine these tools to securely control access to sensitive data stored in the cloud.

Project reflection

This project took me approximately 1 hour to complete. The most challenging part was configuring the key policy correctly to test both denied and allowed access scenarios. It was most rewarding to see how KMS and IAM work together to protect data and how easily I could control access by updating permissions.

placeholder

I did this project today to practice how to secure data in AWS using encryption and access controls, and to understand how AWS KMS, IAM, and DynamoDB work together to protect sensitive information.

This project met my goals because I was able to create and test an encryption key, see how permissions affect data access, and prove that I can manage who can encrypt and decrypt data — all of which are essential skills for building secure cloud solutions.

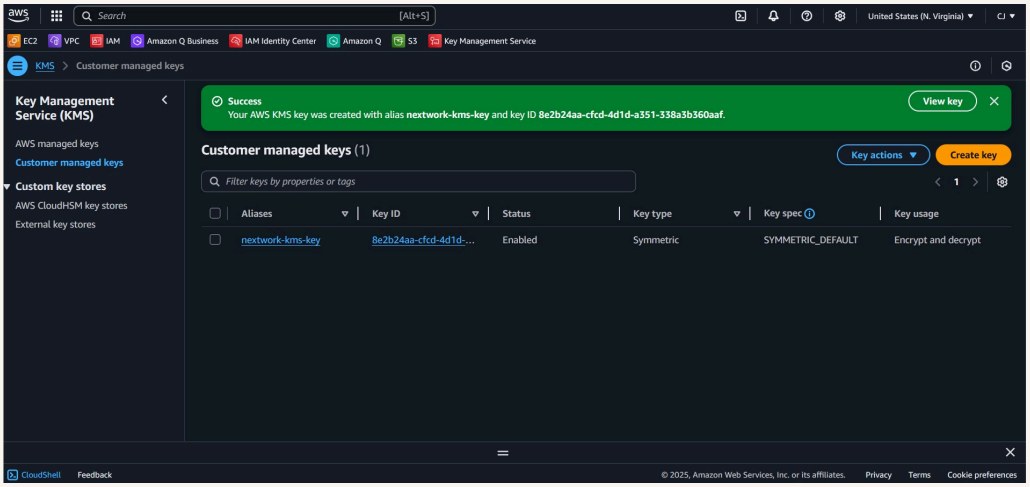
Chrispinus Jacob
NextWork Student

Encryption and KMS

Encryption is the process of converting readable data into an unreadable format to protect it from unauthorized access. Companies and developers do this to secure sensitive information such as customer records, financial data, and confidential business details. Encryption keys are special digital secrets used to lock (encrypt) and unlock (decrypt) this data — without the correct key, the information remains unreadable.

AWS KMS is **Amazon's managed Key Management Service** that helps you create, store, and control encryption keys securely in the cloud. Key management systems are important because they simplify how organizations protect sensitive data — making it easier to generate, rotate, manage, and audit encryption keys without needing to build their own complex security infrastructure.

Encryption keys are broadly categorized as symmetric and asymmetric. I set up a symmetric key because we're going to use the same key to both encrypt and decrypt our data. Symmetric keys are commonly used for encrypting data at rest in services like DynamoDB, as they are faster and easier to manage for this purpose.

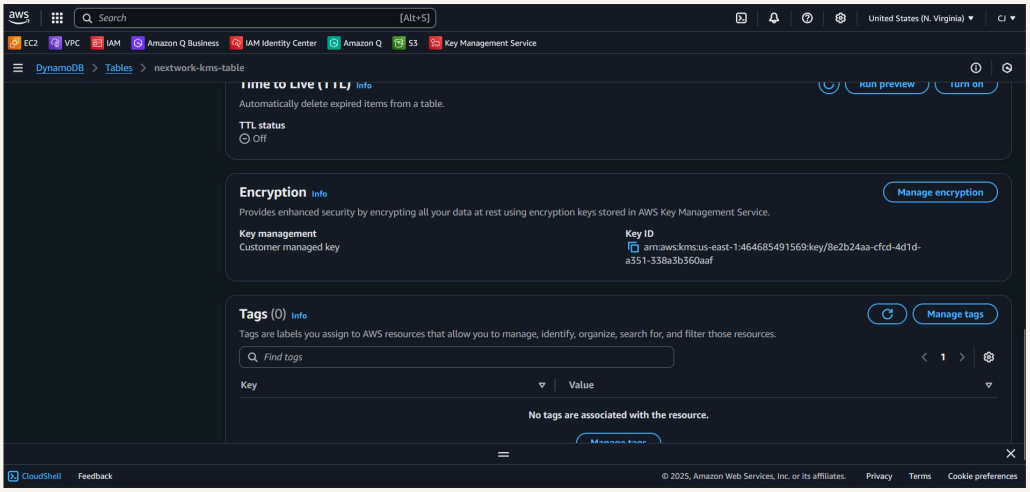

Search
[Alt+S]
United States (N. Virginia)
EC2
VPC
IAM
Amazon Q Business
IAM Identity Center
Amazon Q
S3
Key Management Service
KMS
Customer managed keys
Key Management Service (KMS)
AWS managed keys
Customer managed keys
Custom key stores
AWS CloudHSM key stores
External key stores
Success
Your AWS KMS key was created with alias nextwork-kms-key and key ID 8e2b24aa-cfcd-4d1d-a351-338a3b360aaf.
View key
Customer managed keys (1)
Key actions
Create key
Filter keys by properties or tags
Aliases
Key ID
Status
Key type
Key spec
Key usage
nextwork-kms-key
8e2b24aa-cfcd-4d1d-...
Enabled
Symmetric
SYMMETRIC_DEFAULT
Encrypt and decrypt
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Encrypting Data

My encryption key will safeguard data in DynamoDB, which is Amazon's fully managed NoSQL database service designed for fast and flexible performance at any scale. By encrypting the table, we ensure that all items stored are automatically protected at rest, giving us stronger security and compliance with data protection standards.

The different encryption options in DynamoDB include AWS owned CMK (default encryption), AWS managed CMK, and customer managed CMK. Their differences are based on who controls the encryption keys and the level of control over key management tasks like rotation, permissions, and auditing. I selected a customer managed CMK created with AWS KMS because it gives me full control to manage key policies, define who can use the key

Chrispinus Jacob
NextWork Student

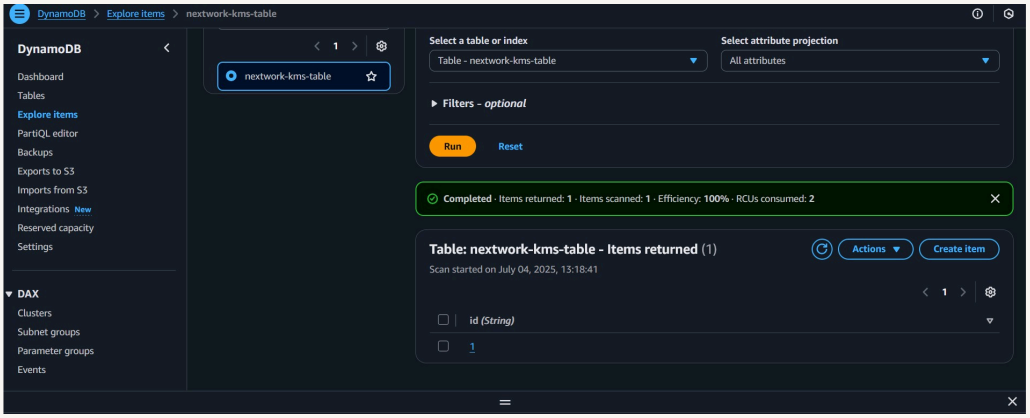


Data Visibility

Rather than controlling who has access to the key, KMS manages user permissions by using key policies and IAM policies together to define who can use, manage, or decrypt data with the key. This means that even if a user has full permissions for a service like DynamoDB, they still need explicit KMS permissions to encrypt or decrypt data protected by a customer managed key. This layered control ensures that access to sensitive data is tightly managed and auditable.

Despite encrypting my DynamoDB table, I could still see the table's items because the encryption happens automatically in the background. DynamoDB uses transparent data encryption, which means the data is encrypted at rest on disk and decrypted on the fly when accessed by an authorized user or application. As long as I have the right IAM permissions and access to the encryption key, DynamoDB handles the encryption and decryption process seamlessly.

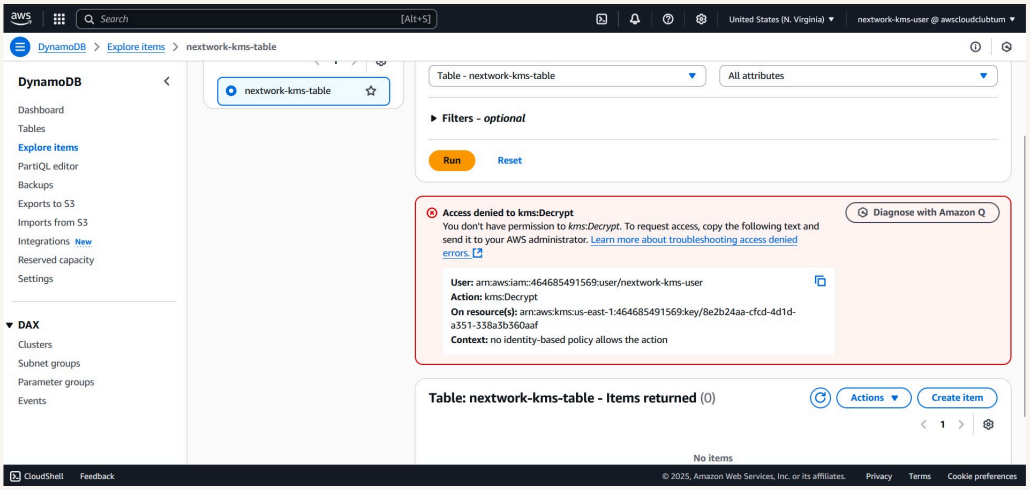
Chrispinus Jacob
NextWork Student



Denying Access

I configured a new IAM user to test access controls on the encrypted DynamoDB table. The permission policies I granted this user are full DynamoDB access (AmazonDynamoDBFullAccess) , but not the KMS decrypt permissions for the customer managed key. This means the user has all the permissions needed to read and write data in DynamoDB, but will still be blocked from accessing the encrypted data because they can't decrypt it without the required key access.

After accessing the DynamoDB table as the test user, I encountered an access denied error because the user did not have permission to decrypt the data with the customer managed KMS key. This confirmed that DynamoDB encryption with KMS works as expected, blocking unauthorized users from reading or writing encrypted data even if they have full DynamoDB access permissions.


aws
Search
[Alt+S]
United States (N. Virginia)
nextwork-kms-user @ awscloudclubtum
DynamoDB > Explore items > nextwork-kms-table
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
nextwork-kms-table
Table - nextwork-kms-table
All attributes
Filters - optional
Run
Reset
Access denied to kms:Decrypt
You don't have permission to kms:Decrypt. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.
Diagnose with Amazon Q
User: arn:aws:iam::464685491569:user/nextwork-kms-user
Action: kms:Decrypt
On resource(s): arn:aws:kms:us-east-1:464685491569:key/8e2b24aa-cfcd-4d1d-a351-338a3b360aaf
Context: no identity-based policy allows the action
Table: nextwork-kms-table - Items returned (0)
Actions
Create item
No items
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

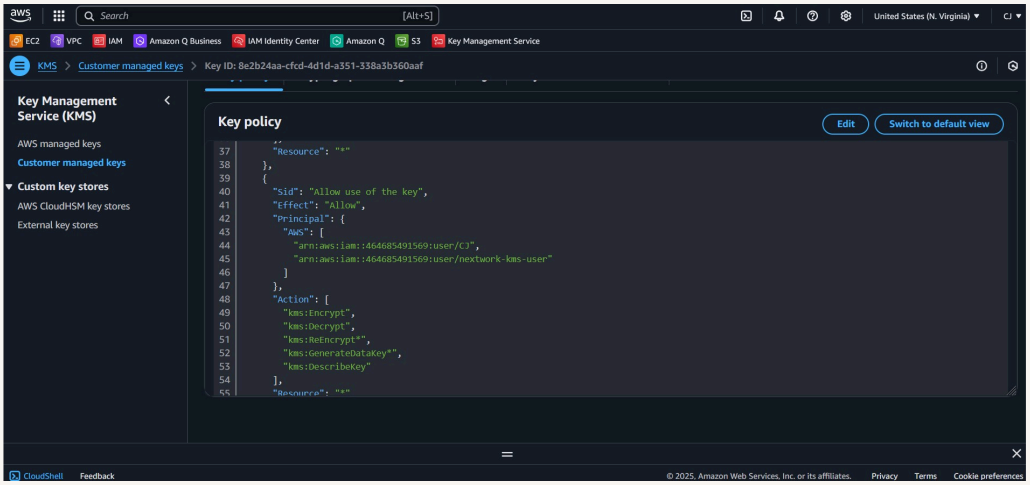
EXTRA: Granting Access

To let my test user use the encryption key, I granted the user permission to decrypt and encrypt data with the KMS key. My key's policy was updated to include the test user's IAM ARN and allow `kms:Decrypt` and `kms:Encrypt` actions, which means the user can now read and write the encrypted DynamoDB data. This shows how KMS policies can be adjusted to securely enable access when needed.

Using the test user, I retried accessing the encrypted DynamoDB table. I observed that the user could now successfully read and write the data, which confirmed that the updated KMS key permissions worked as expected— allowing authorized users to decrypt and interact with the protected data when granted the right access.

Encryption secures data instead of simply hiding it from view. I could combine encryption with fine-grained IAM policies and resource-based policies to control who can access, manage, and decrypt sensitive data, ensuring that only trusted users and applications have the permissions they need while everyone else is blocked by default.

Chrispinus Jacob
NextWork Student



The place to learn & showcase your skills

Check out nextwork.org for more projects