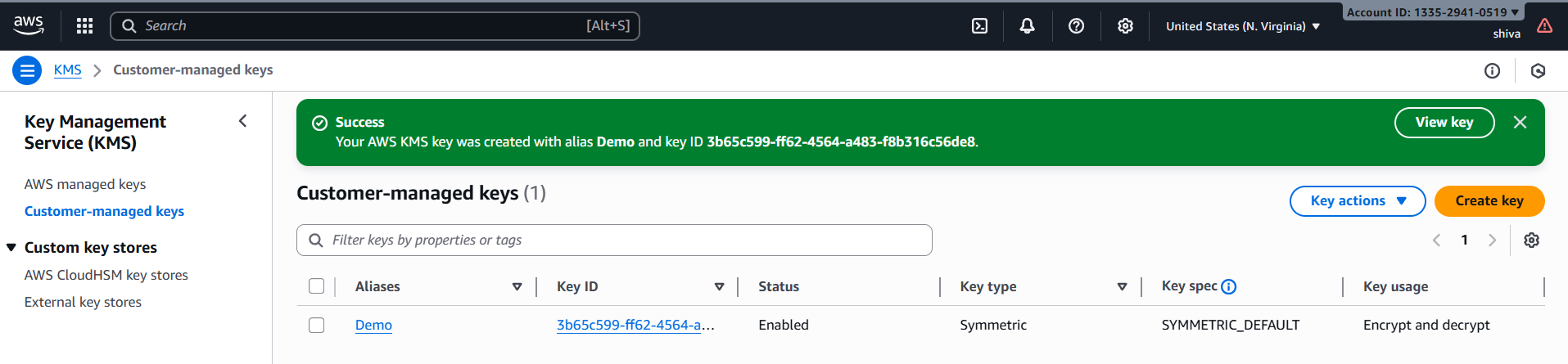
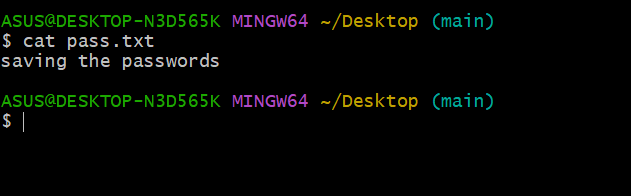
TASK: Use AWS KMS to generate a data encryption key (DEK), encrypt a   
sensitive file with OpenSSL, store only the encrypted DEK with the ciphertext,   
and later recover the DEK via KMS to decrypt the file

Created a KMS key (ex: Demo)

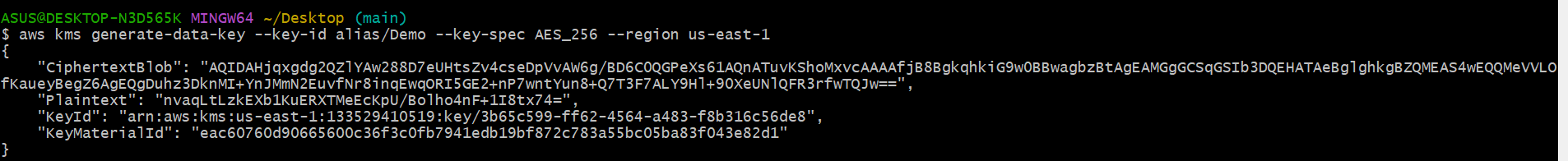


Open gitbash or any cli, creating a file with my aws accounts passwords or  
credentials

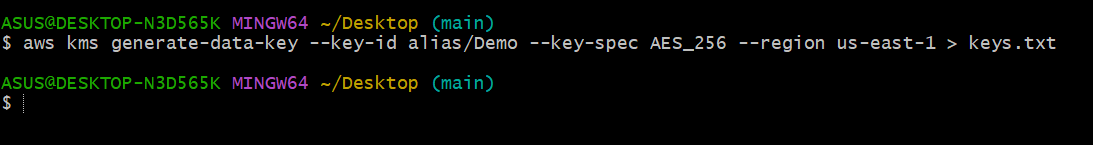


Generate a KMS Data Key (DEK) using below cmnd

aws kms generate-data-key --key-id alias/Demo --key-spec   
AES\_256 --region us-east-1

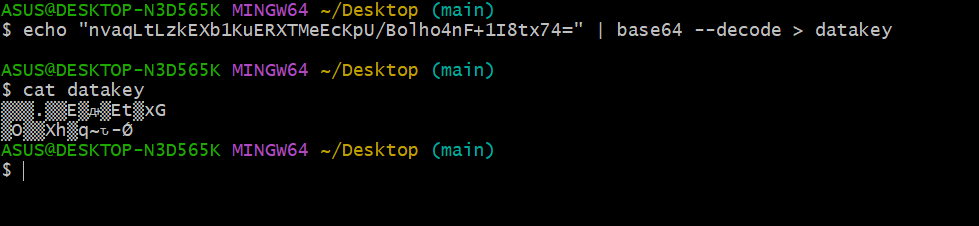


KMS returned:  
● Plaintext (base64): the raw 256-bit data key, encoded  
● CiphertextBlob (base64): the same data key, encrypted under your   
CMK  
● KeyId: which CMK was used  
I redirected the JSON to keys.txt for reuse:  
aws kms generate-data-key --key-id alias/Demo --key-spec AES\_256 –region us-east-1 > keys.txt



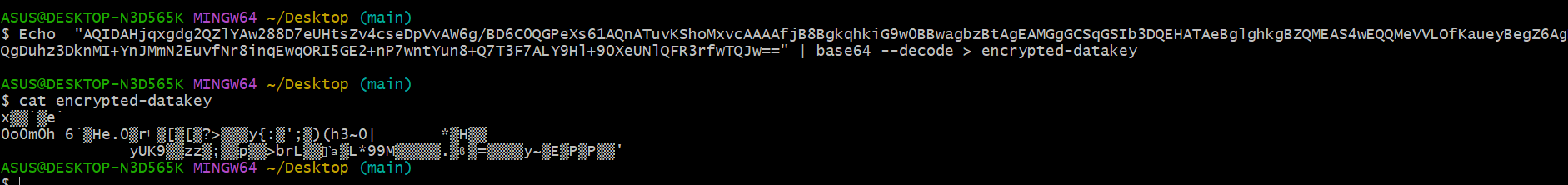
Create a binary file with the plaintext DEK  
● You base64-decoded the Plaintext into datakey (binary):  
● datakey looks like gibberish in cat because it’s raw bytes   
(expected)

echo "nvaqLtLzkEXb1KuERXTMeEcKpU/Bolho4nF+1I8tx74=" | base64 --decode > datakey



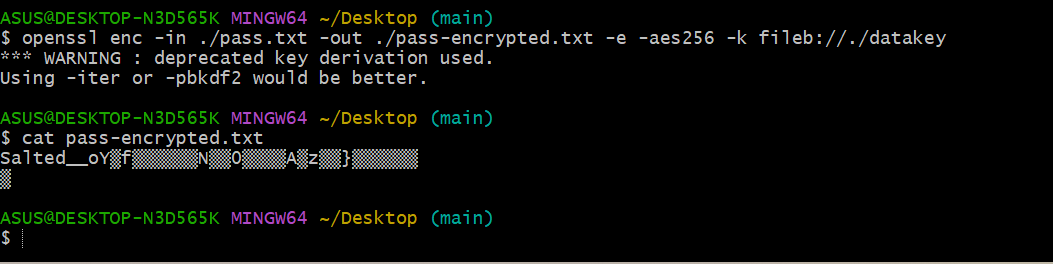
Create a binary file with the encrypted DEK  
● You base64-decoded the CiphertextBlob into encrypted-  
datakey:  
● This file is safe to store next to your ciphertext. It can only be turned   
back into the plaintext key by KMS (and only if caller is authorized)

Echo “copy the ciphertextbob” | base64 --decode > encrypted-datakey

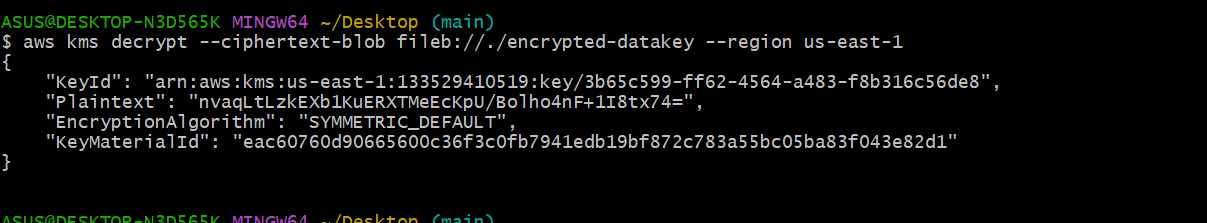


Encrypt your secrets with OpenSSL

openssl enc -in ./pass.txt -out ./pass-encrypted.txt -e -aes256 -k fileb://./datakey



(Later) Recover the plaintext DEK from KMS  
● You asked KMS to decrypt the stored encrypted DEK:  
● aws kms decrypt --ciphertext-blob fileb://./encrypted-datakey --region us-east-1  
● KMS returned the same Plaintext (base64) DEK you had originally



Decrypt your file  
● You used OpenSSL with the same parameters to get back the   
cleartext:  
● openssl enc -in ./pass-encrypted.txt -out ./pass-decrypted.txt -d -aes256 -k fileb://./datakey

pass-decrypted.txt matched your original password.txt

