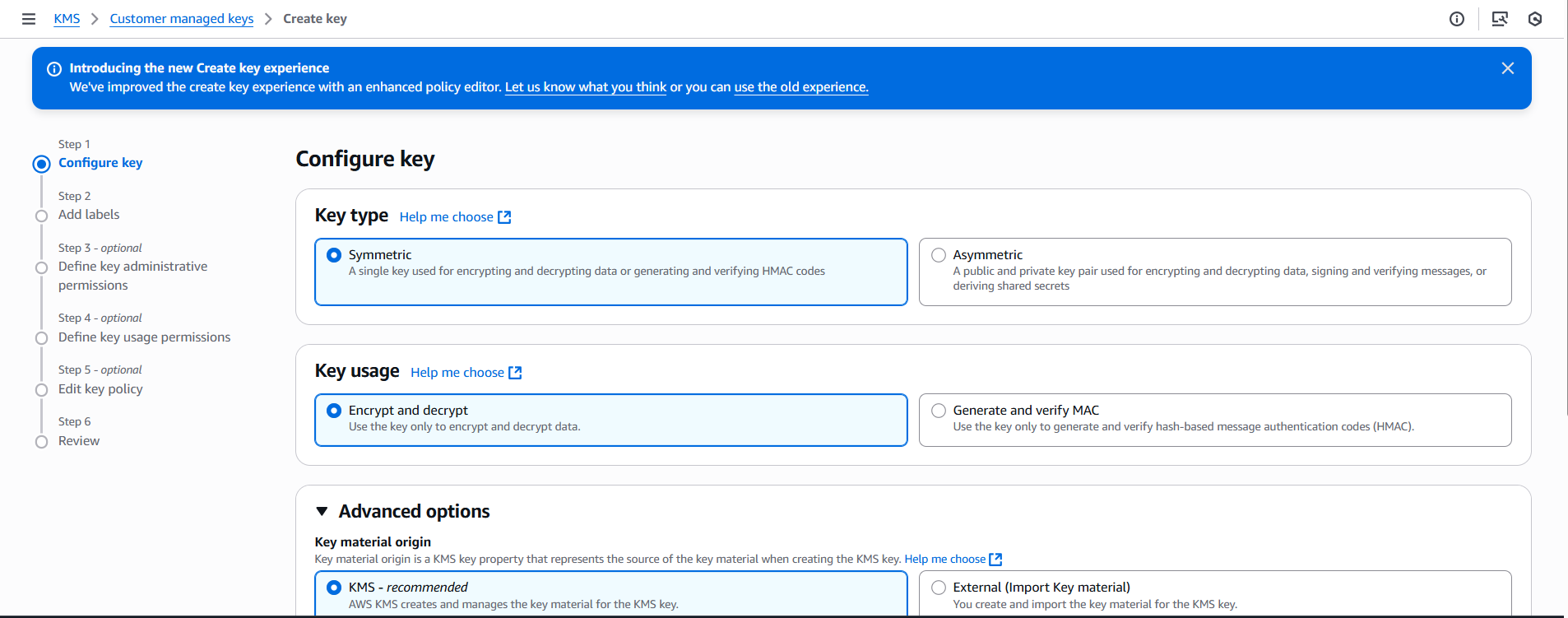
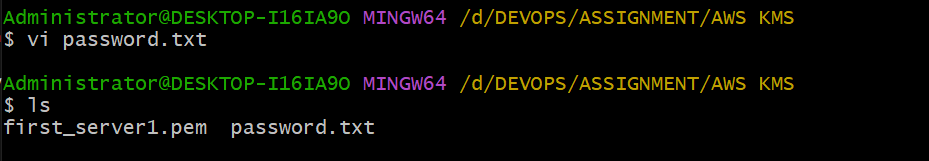
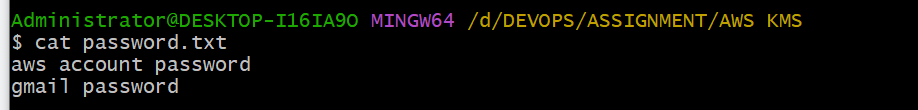
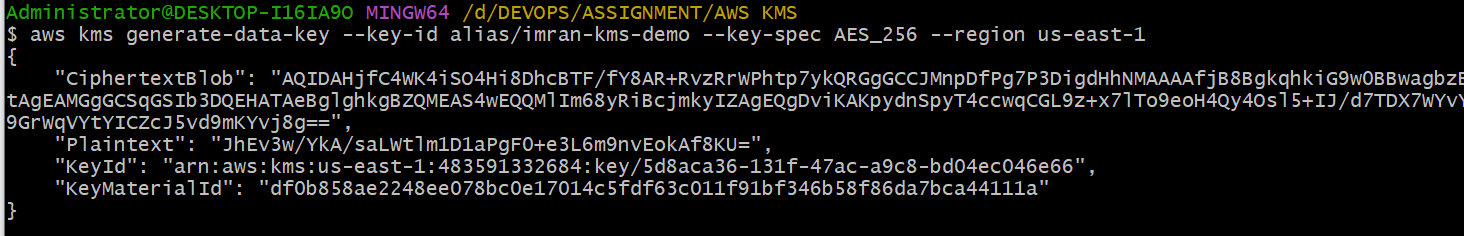
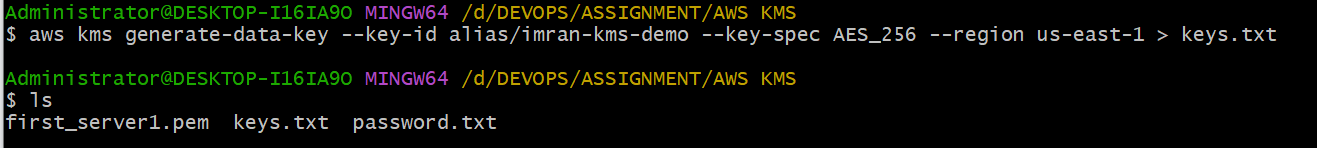
**TASK:** Use AWS KMS to generate a **data encryption key (DEK)**, encrypt a sensitive file with OpenSSL, store only the **encrypted DEK** with the ciphertext, and later **recover** the DEK via KMS to decrypt the file.

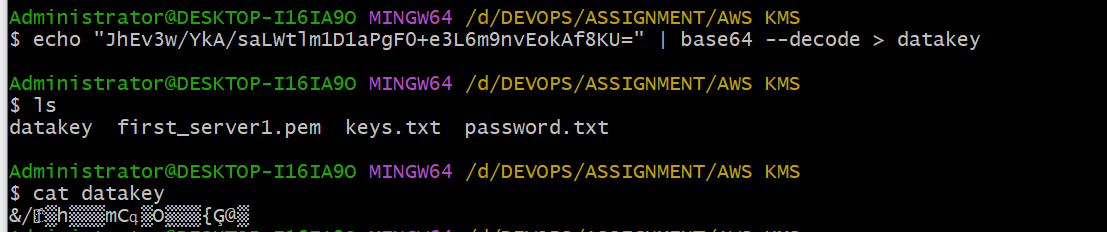
1. Created a KMS key **“imran-kms-demo”**
2. 
3. Open gitbash or any cli, creating a file with my aws accounts passwords or credentials
4. 
5. 
6. **Generate a KMS Data Key (DEK)**
7. **aws kms generate-data-key --key-id alias/imran-kms-demo --key-spec AES\_256 --region us-east-1**
8. ****

**KMS returned:**

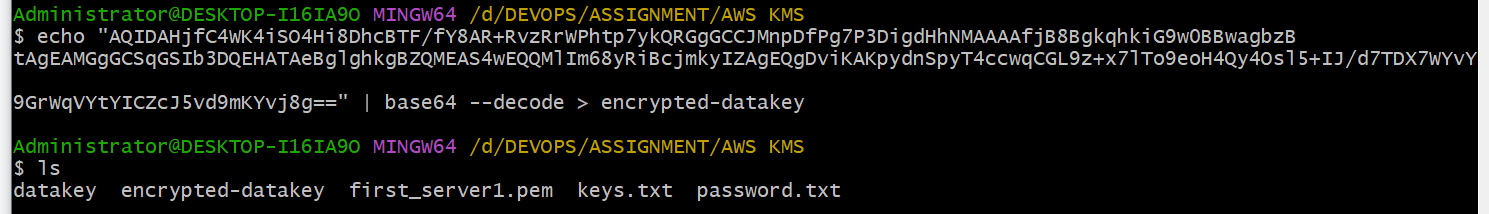
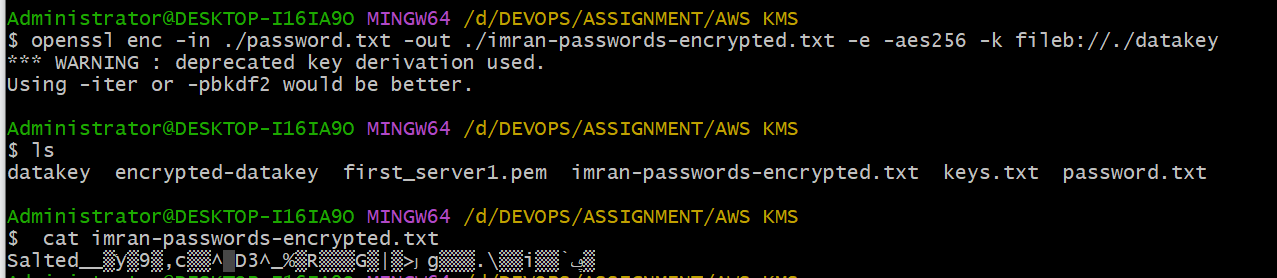
* **Plaintext (base64): the raw 256-bit data key, encoded**
* **CiphertextBlob (base64): the same data key, encrypted under your CMK**
* **KeyId: which CMK was used**

1. **I redirected the JSON to keys.txt for reuse:**
2. **aws kms generate-data-key --key-id alias/imran-kms-demo --key-spec AES\_256 --region us-east-1 > keys.txt**
3. ****
4. **Create a binary file with the plaintext DEK**

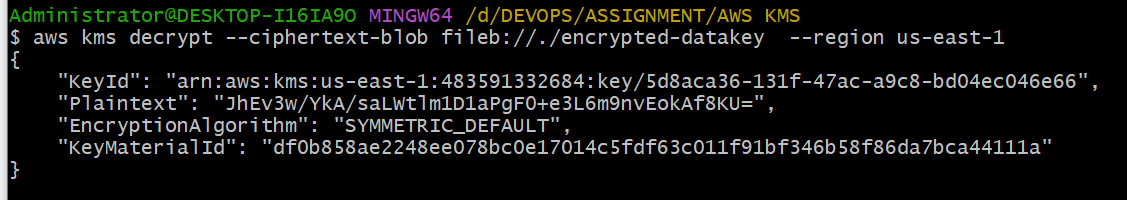
* **You base64-decoded the Plaintext into datakey (binary):**
* **datakey looks like gibberish in cat because it’s raw bytes (expected).**

1. ****
2. **Create a binary file with the encrypted DEK**

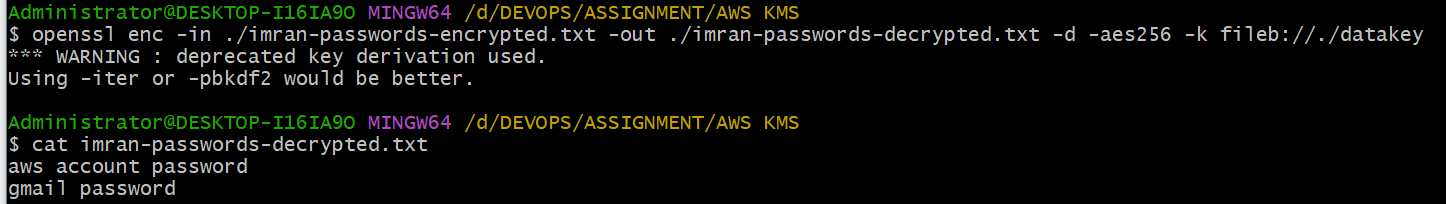
* **You base64-decoded the CiphertextBlob into encrypted-datakey:**
* **This file is safe to store next to your ciphertext. It can only be turned back into the plaintext key by KMS (and only if caller is authorized).**

1. **Echo “copy the ciphertextbob” | base64 --decode > encrypted-datakey**
2. ****
3. **Encrypt your secrets with OpenSSL**
4. **openssl enc -in ./password.txt -out ./imran-passwords-encrypted.txt -e -aes256 -k fileb://./datakey**
5. **Result: imran-passwords-encrypted.txt (unreadable binary).**
6. ****
7. **(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.**

1. ****
2. **Decrypt your file**

* **You used OpenSSL with the same parameters to get back the cleartext:**
* **openssl enc -in ./imran-passwords-encrypted.txt -out ./imran-passwords-decrypted.txt -d -aes256 -k fileb://./datakey**
* **imran-passwords-decrypted.txt matched your original password.txt ✅**

1. ****