

# 1) encryption

aws kms encrypt --key-id alias/tutorial --plaintext fileb://ExampleSecretFile.txt --output text --query CiphertextBlob  --region eu-west-2 > ExampleSecretFileEncrypted.base64

# base64 decode for Linux or Mac OS

cat ExampleSecretFileEncrypted.base64 | base64 --decode > ExampleSecretFileEncrypted

# base64 decode for Windows

certutil -decode .\ExampleSecretFileEncrypted.base64 .\ExampleSecretFileEncrypted

# 2) decryption

aws kms decrypt --ciphertext-blob fileb://ExampleSecretFileEncrypted   --output text --query Plaintext > ExampleFileDecrypted.base64  --region eu-west-2

# base64 decode for Linux or Mac OS

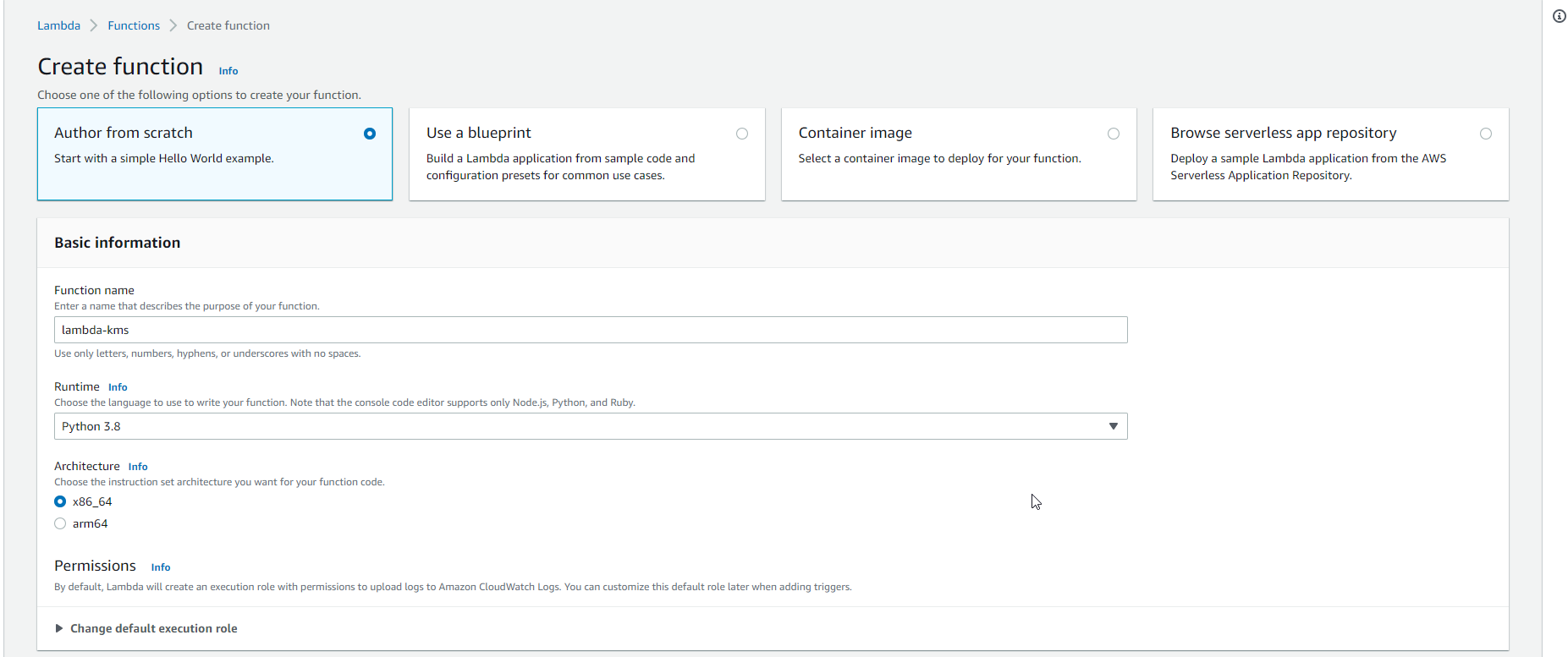
cat ExampleFileDecrypted.base64 | base64 --decode > ExampleFileDecrypted.txt

# base64 decode for Windows

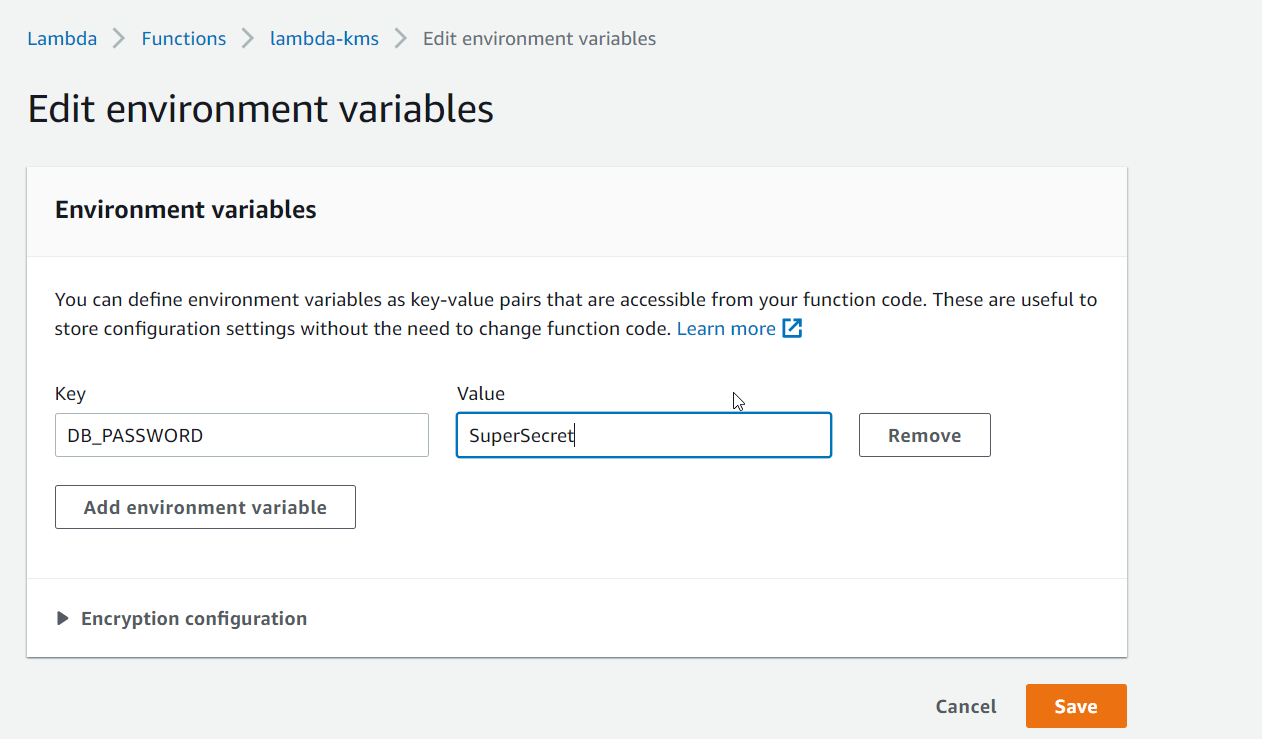
certutil -decode .\ExampleFileDecrypted.base64 .\ExampleFileDecrypted.txt

**KMS AND LAMDA PRACTICE**

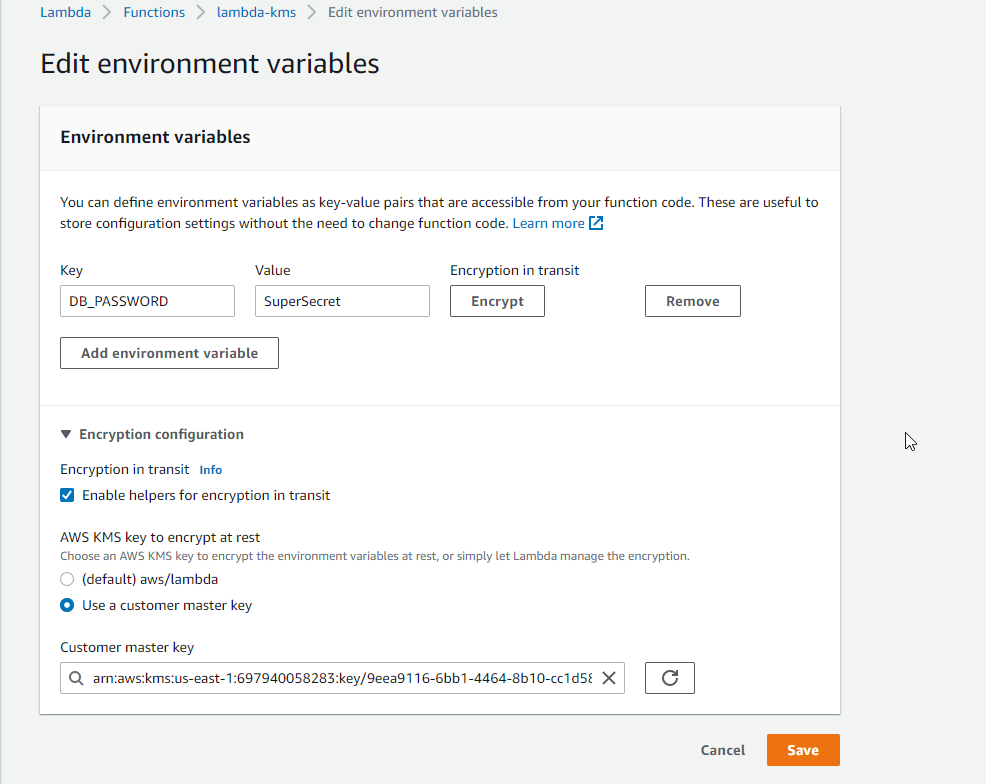
Create lambda



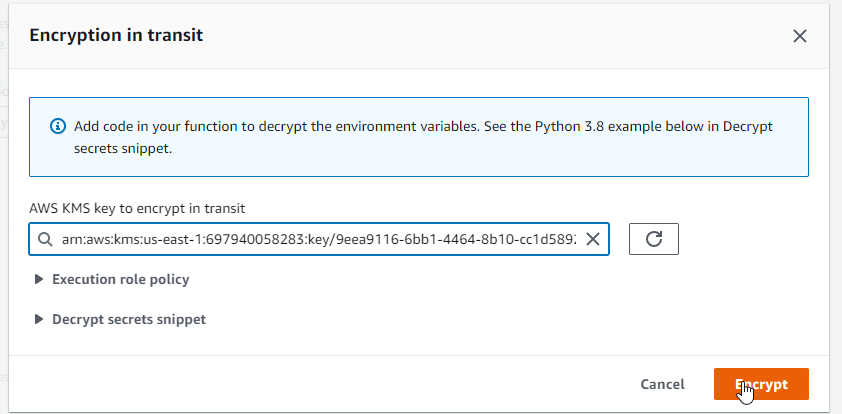
Go to configuration

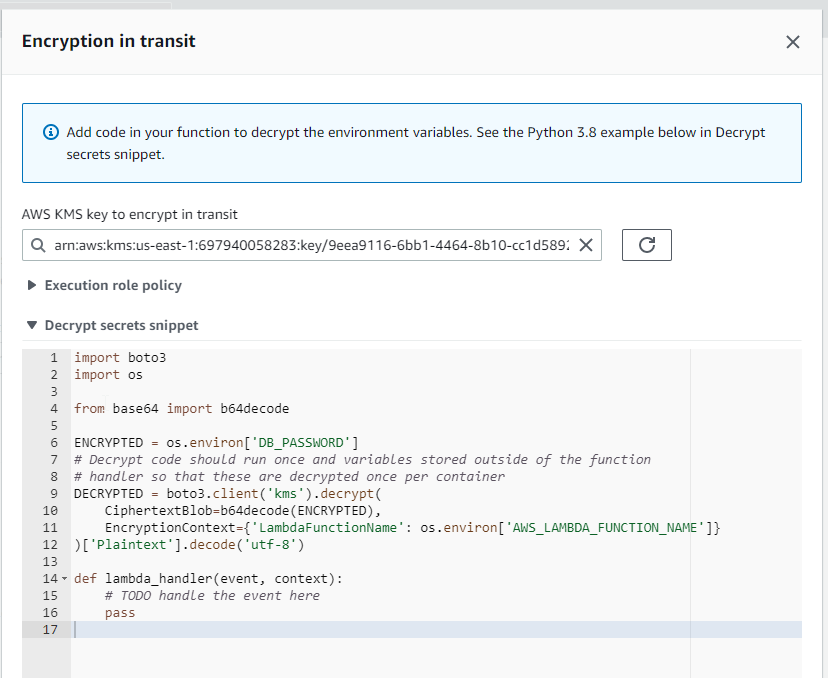


Same as

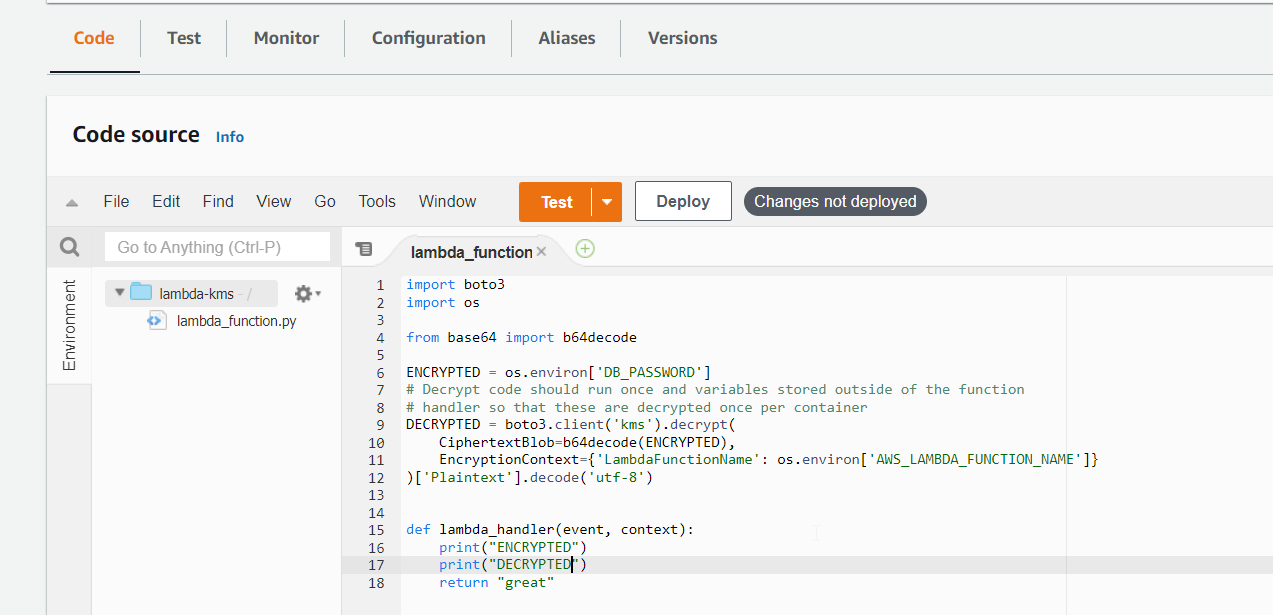


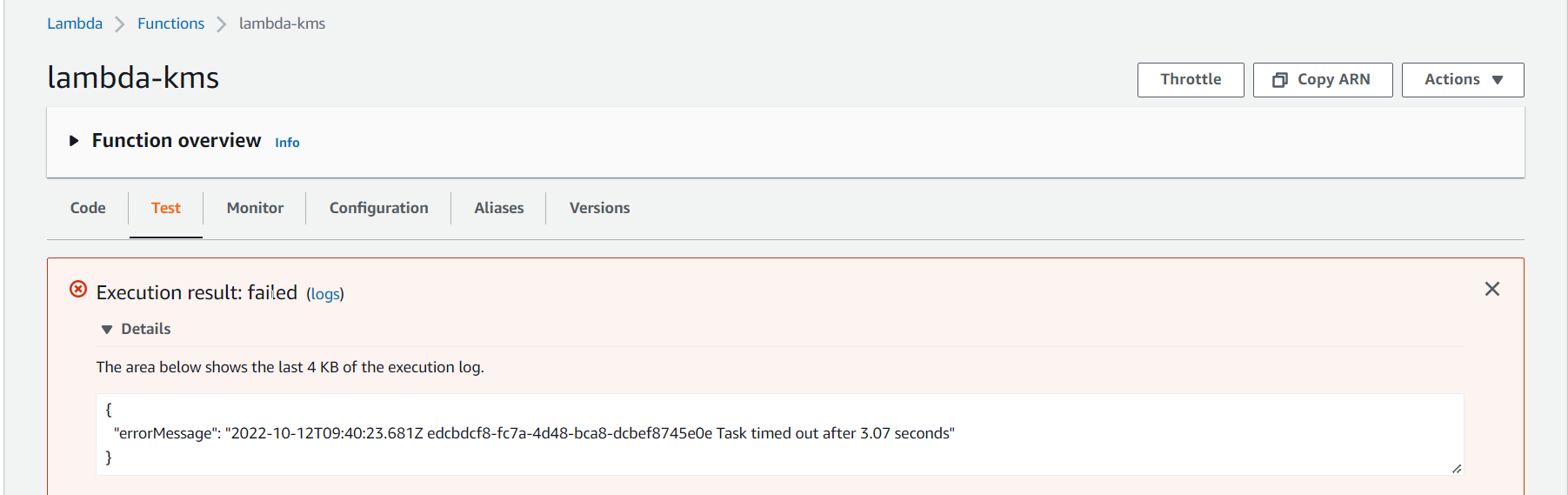
Click encrypt





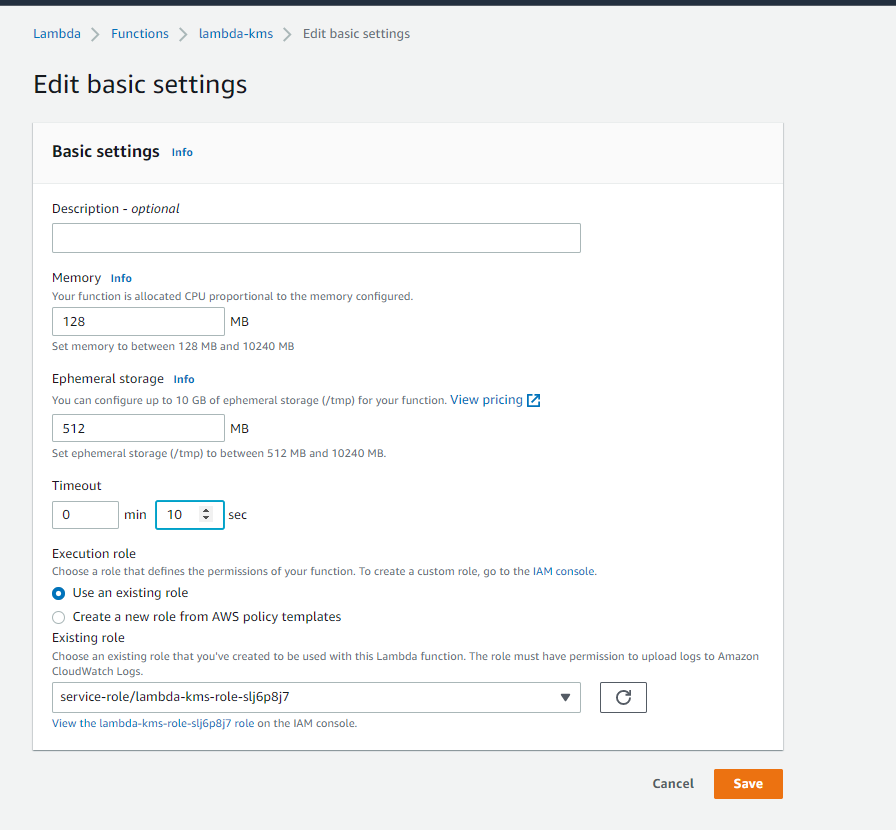
Paste and deploy and test on lambda



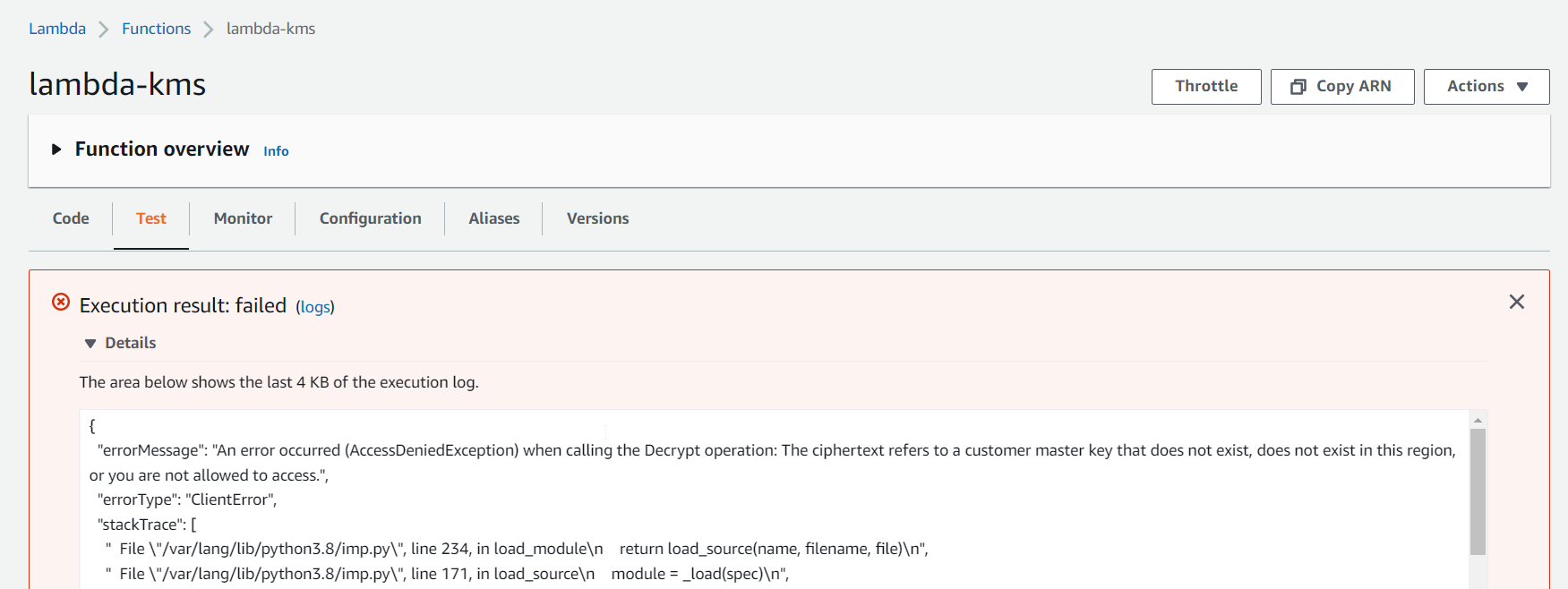


Go to configuration and general





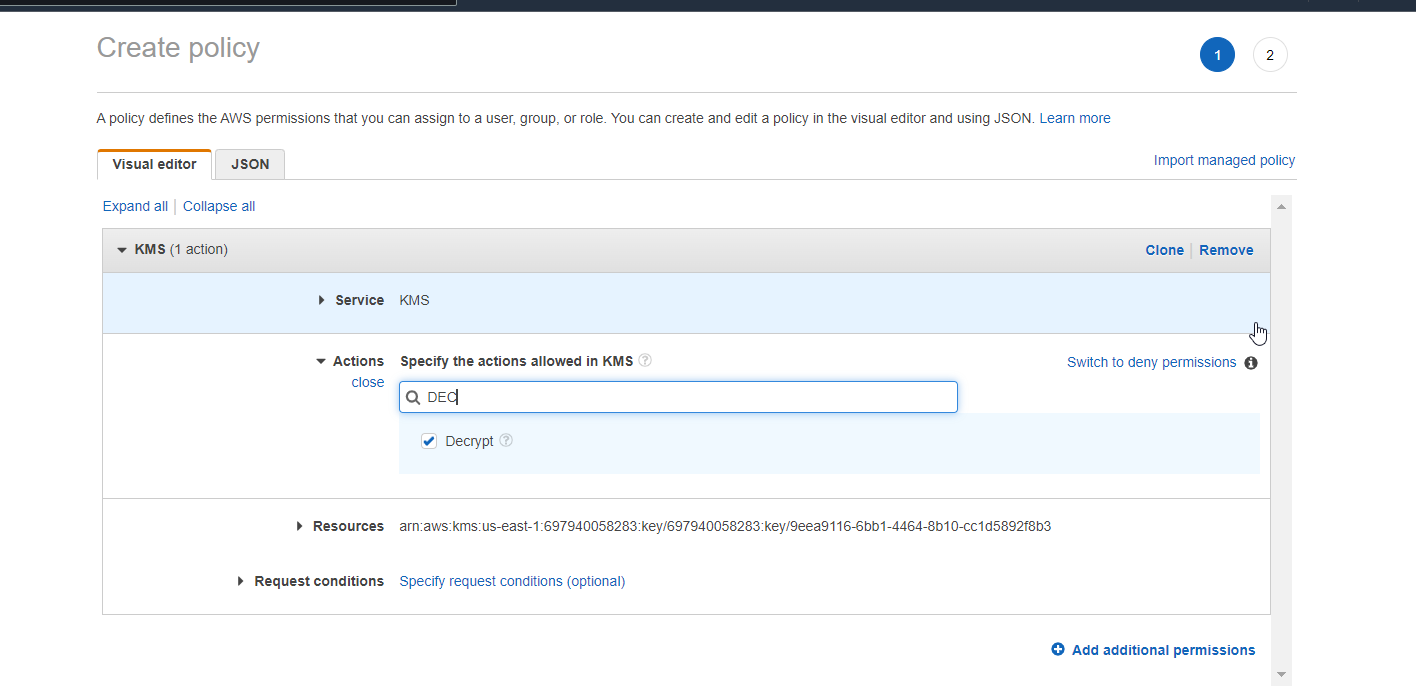
Again test our function

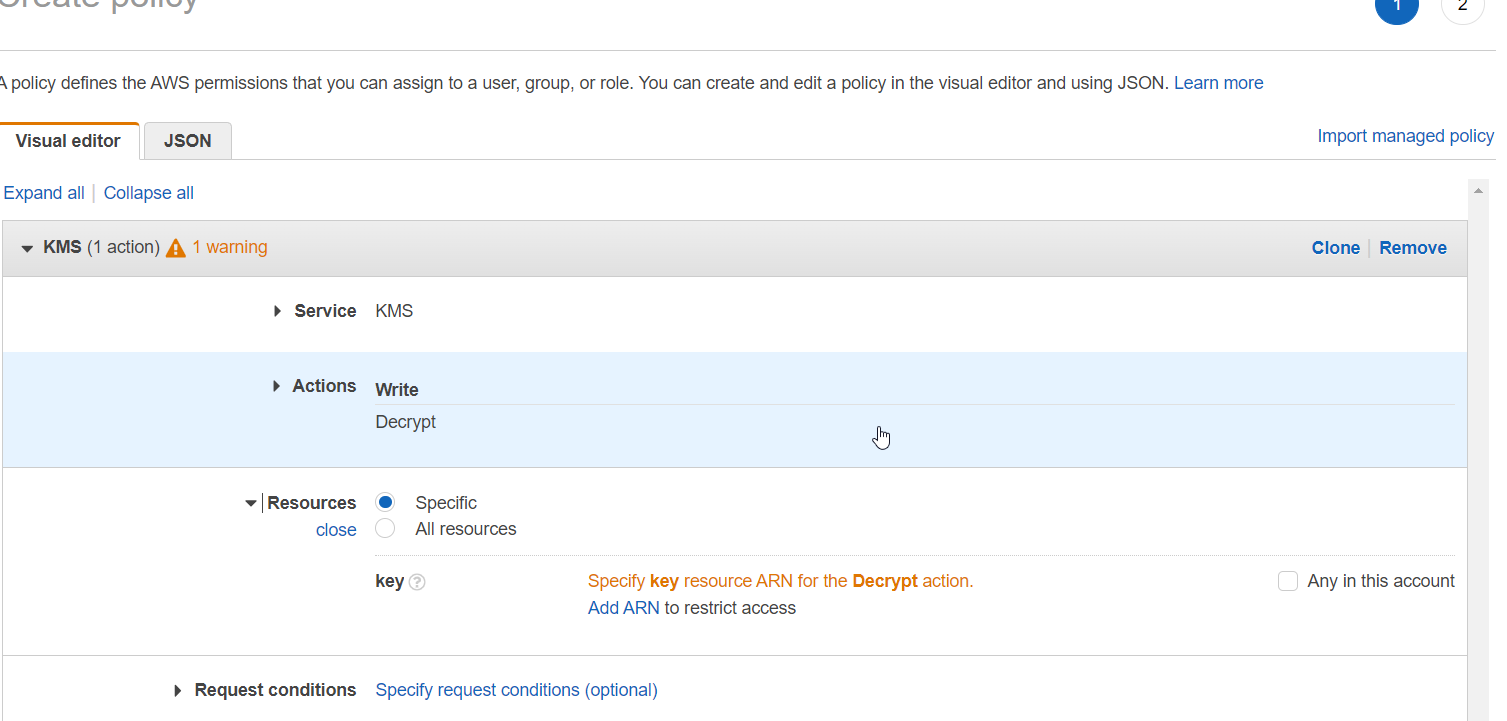


ADD IAM POLICY

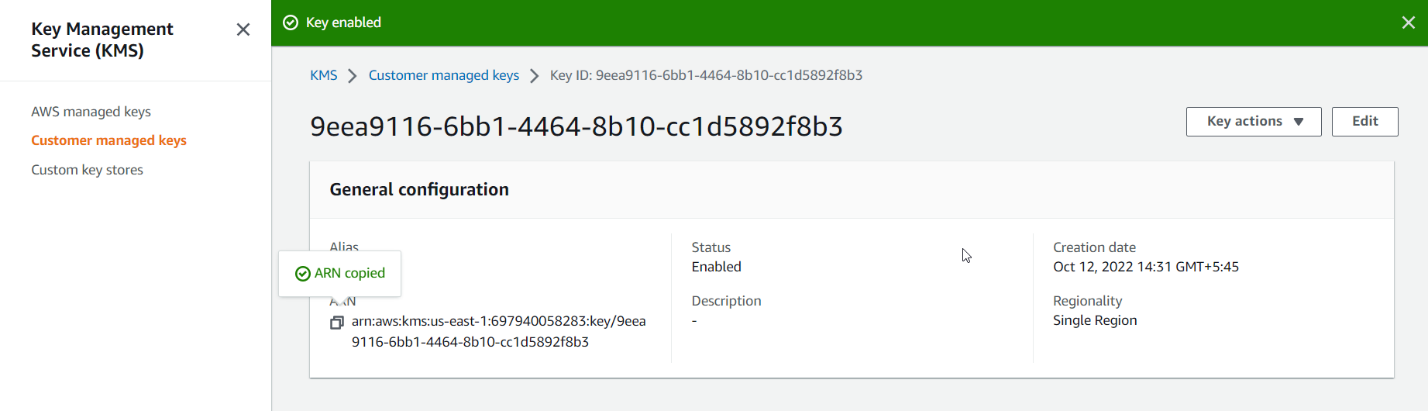
KMS DECRYPT

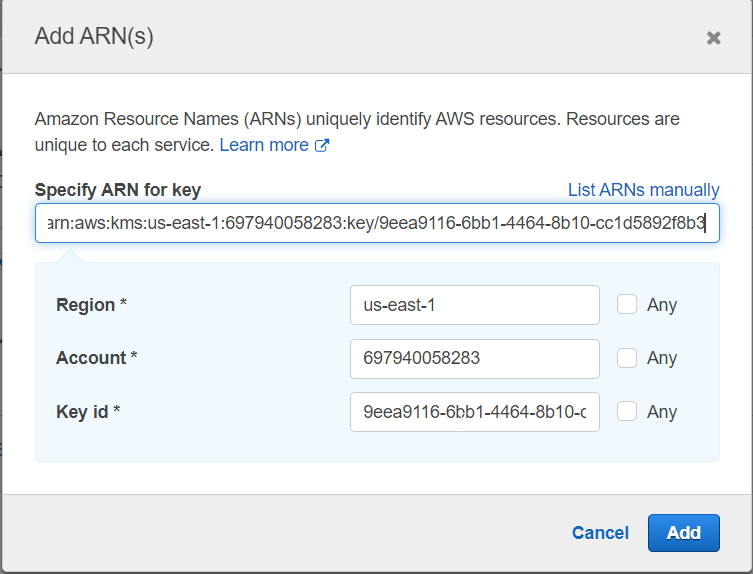
Create inline policy

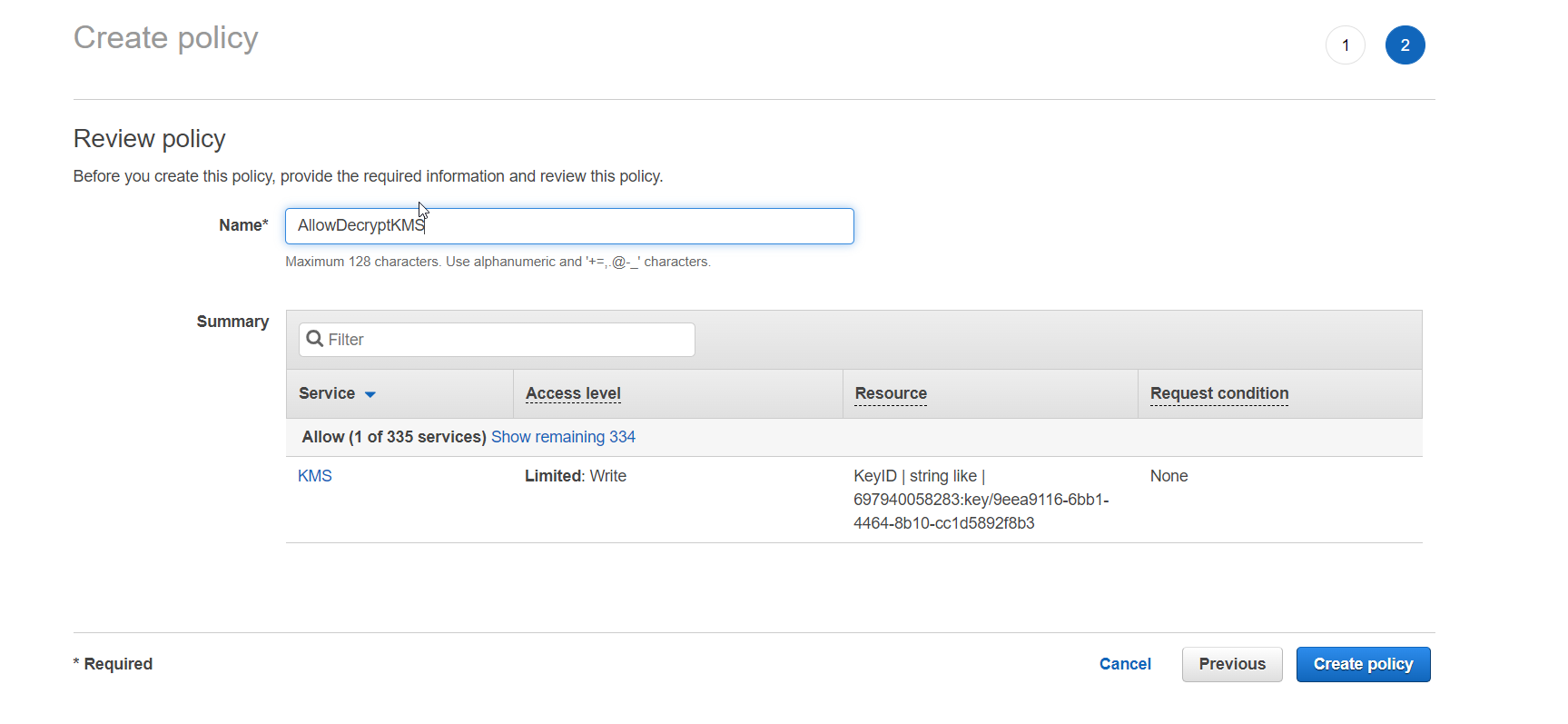




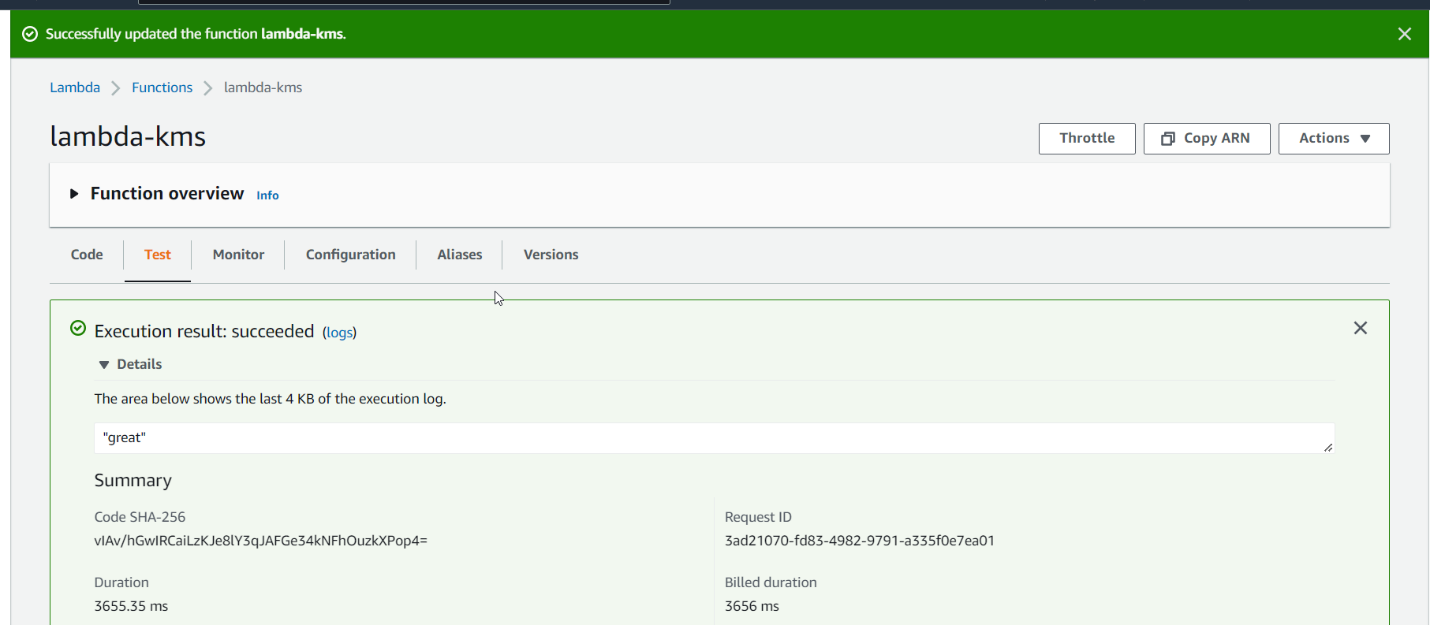
Copy arn key from KMS

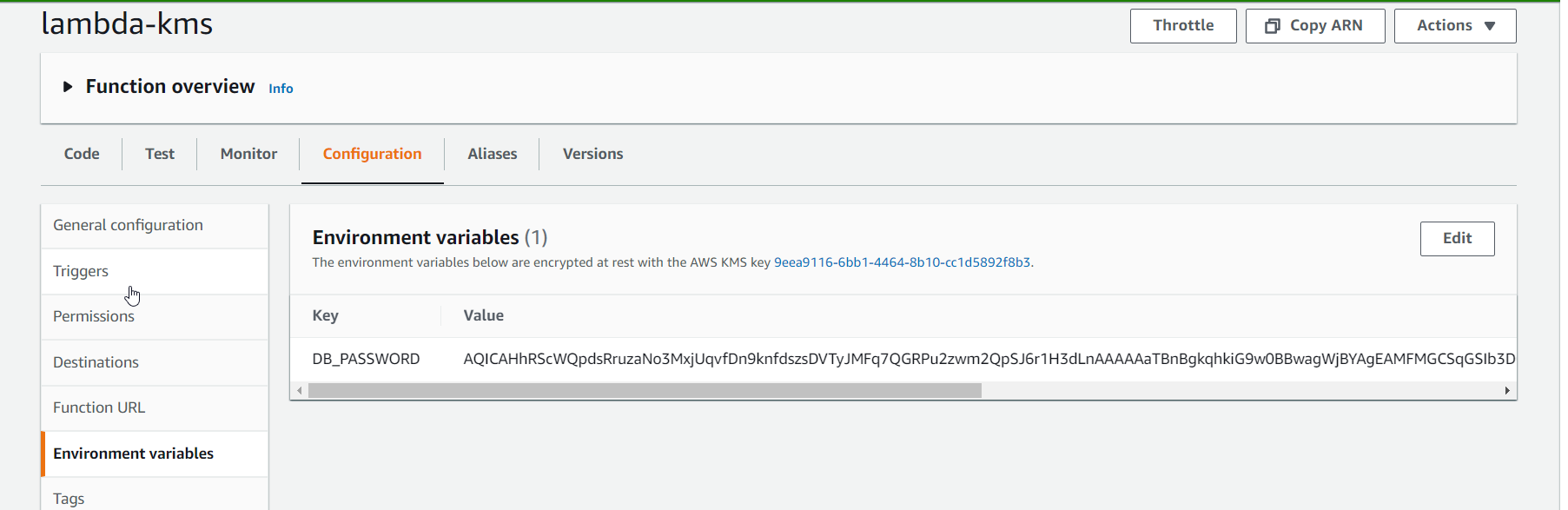






Go back to code and test

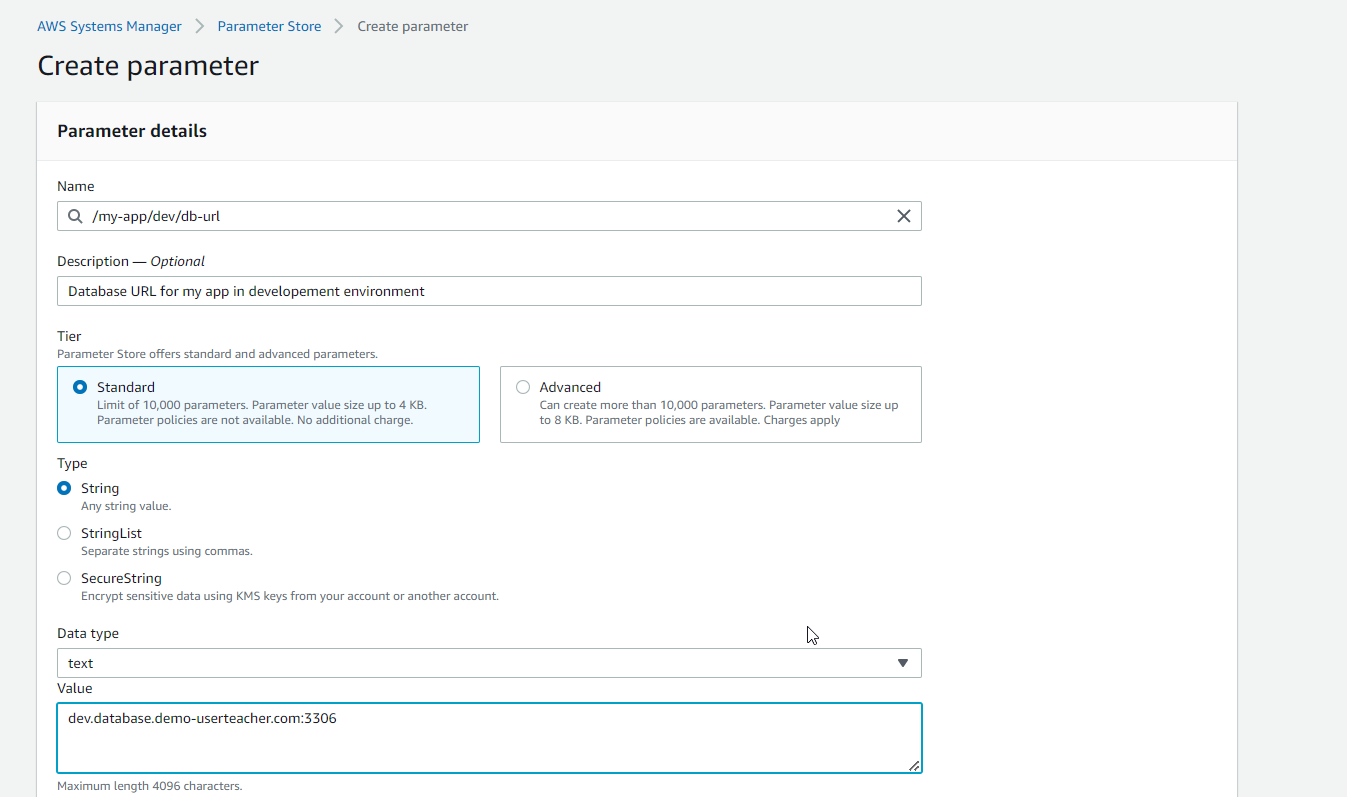


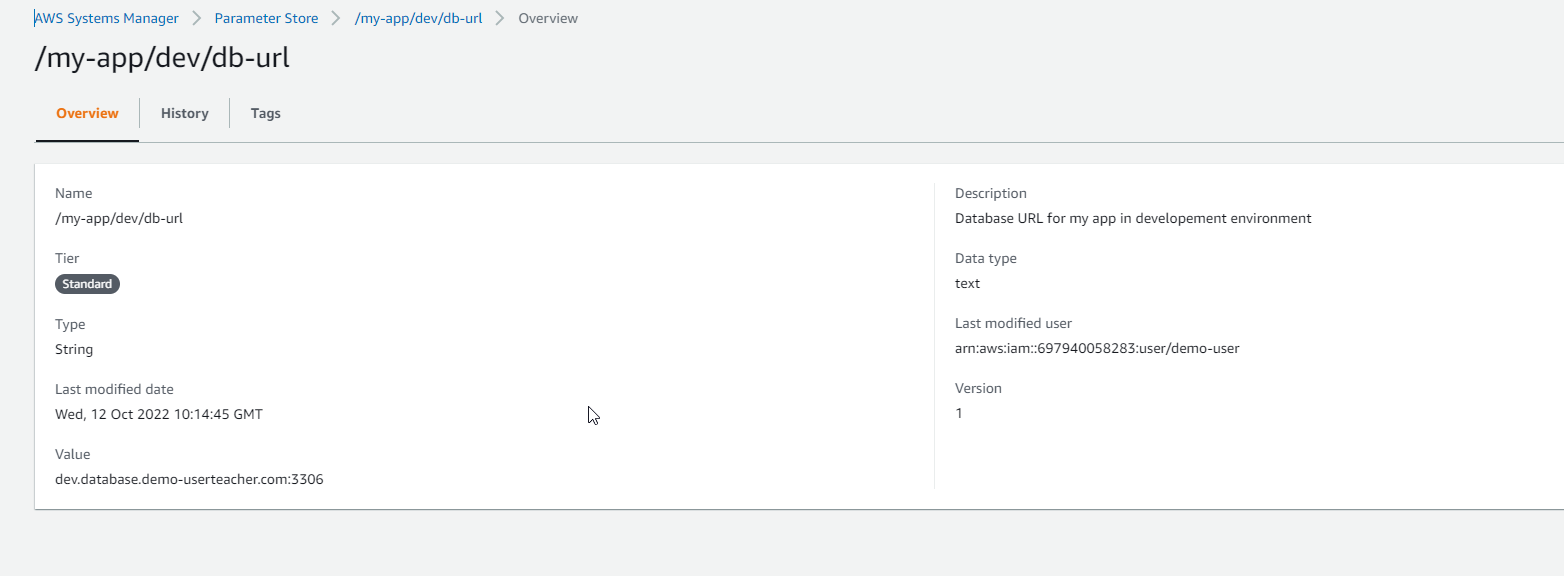


Ssm parameter

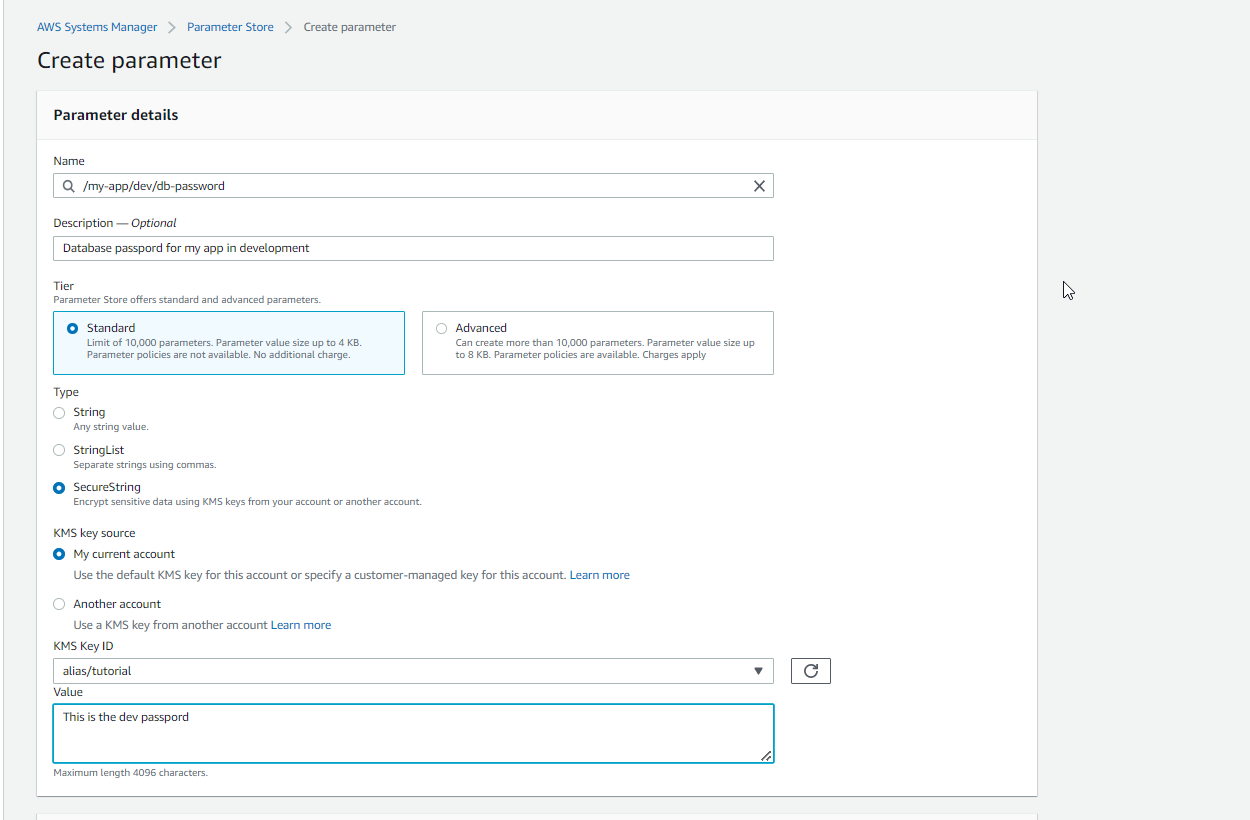


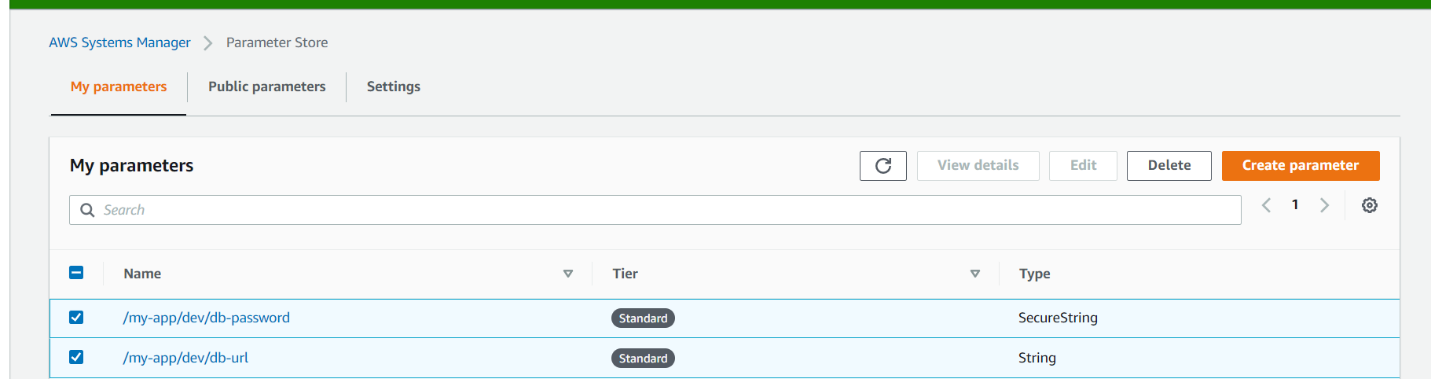
Click parameter store



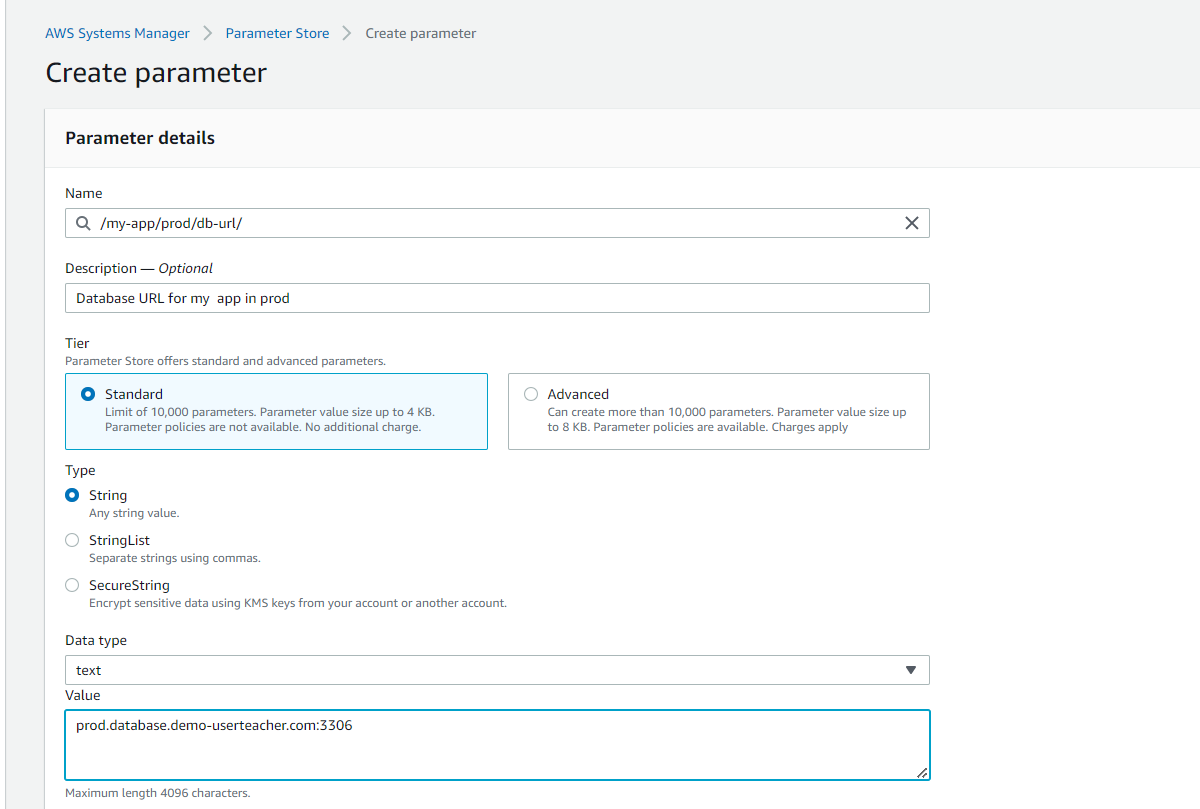


Create another parameter

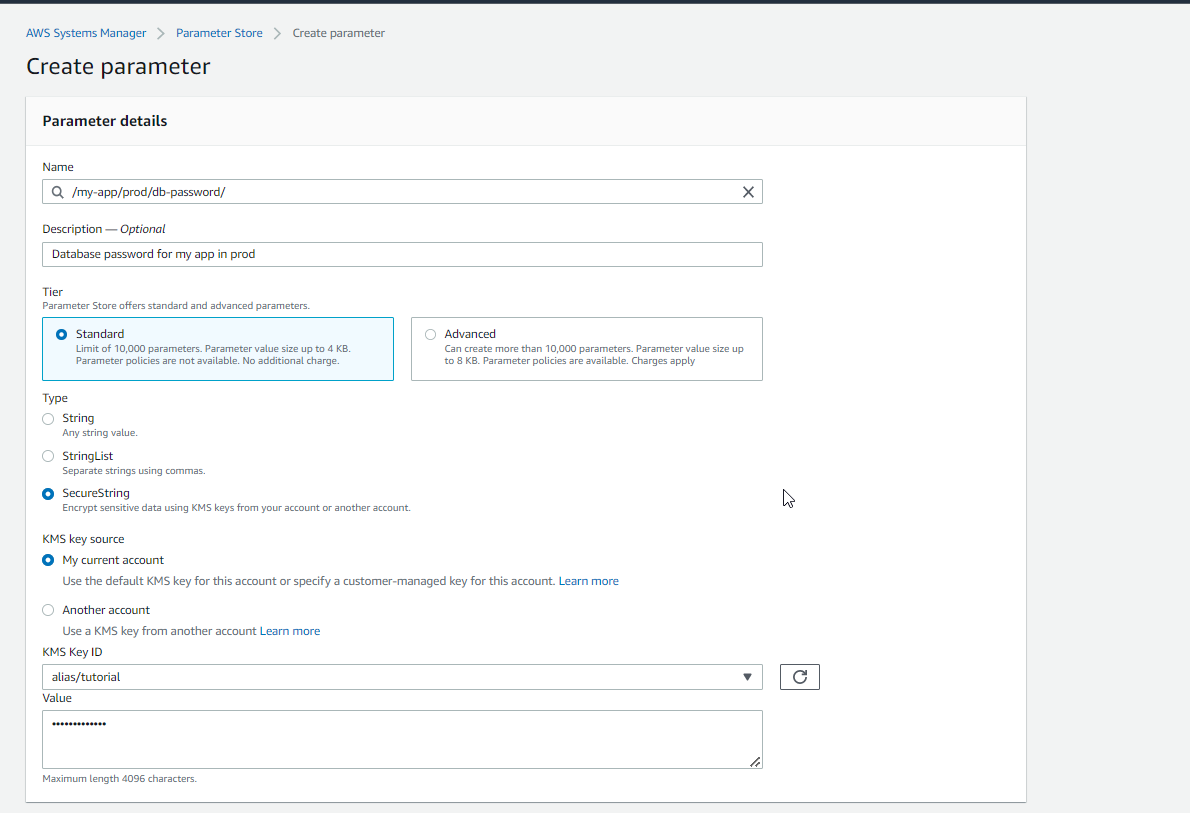




Create another parameter



Create last parameter



# GET PARAMETERS

aws ssm get-parameters --names /my-app/dev/db-url /my-app/dev/db-password

# GET PARAMETERS WITH DECRYPTION

aws ssm get-parameters --names /my-app/dev/db-url /my-app/dev/db-password --with-decryption

# GET PARAMETERS BY PATH

aws ssm get-parameters-by-path --path /my-app/dev/

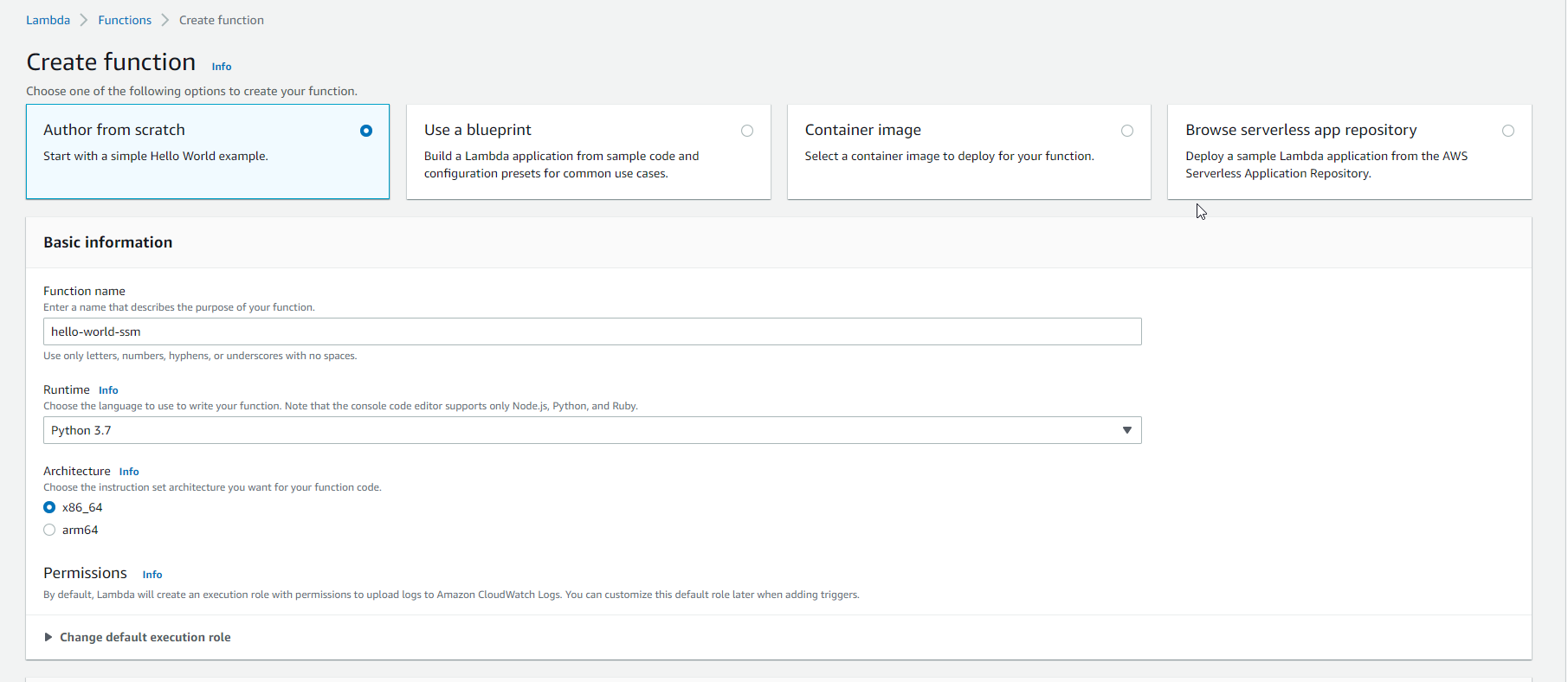
# GET PARAMETERS BY PATH RECURSIVE

aws ssm get-parameters-by-path --path /my-app/ --recursive

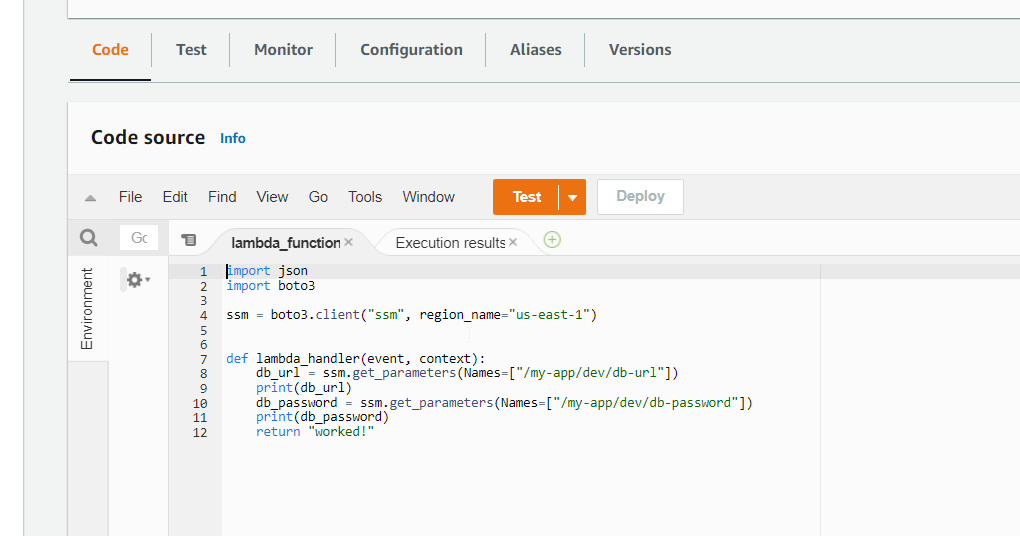
# GET PARAMETERS BY PATH WITH DECRYPTION

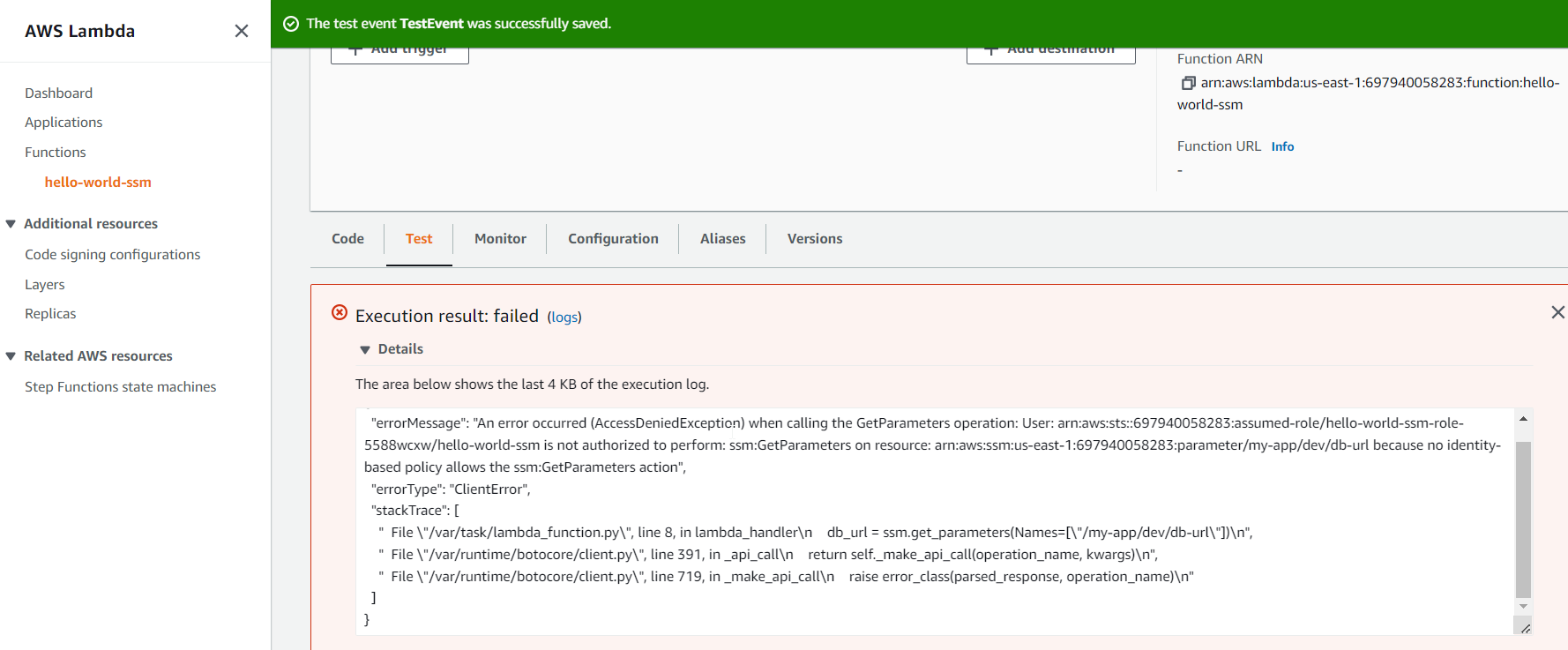
aws ssm get-parameters-by-path --path /my-app/ --recursive --with-decryption

SSM parameter on Lambda

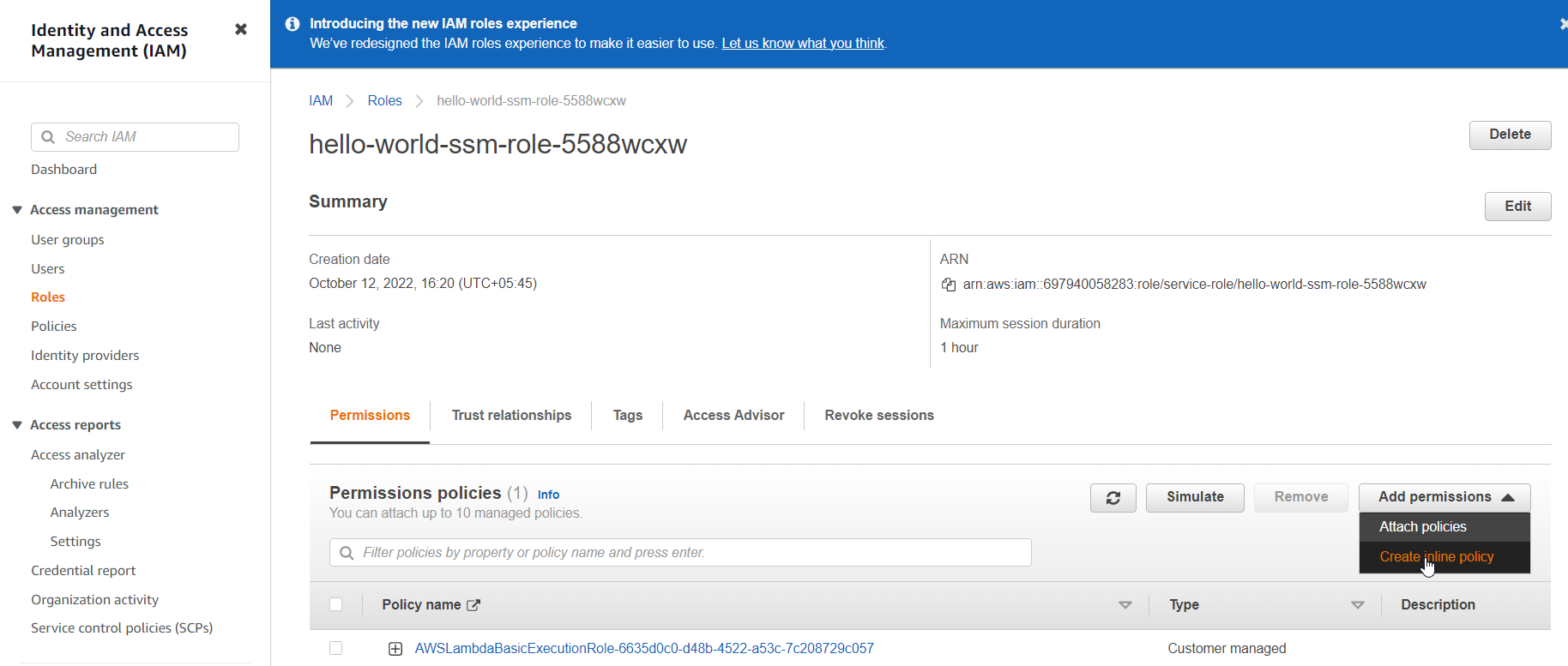


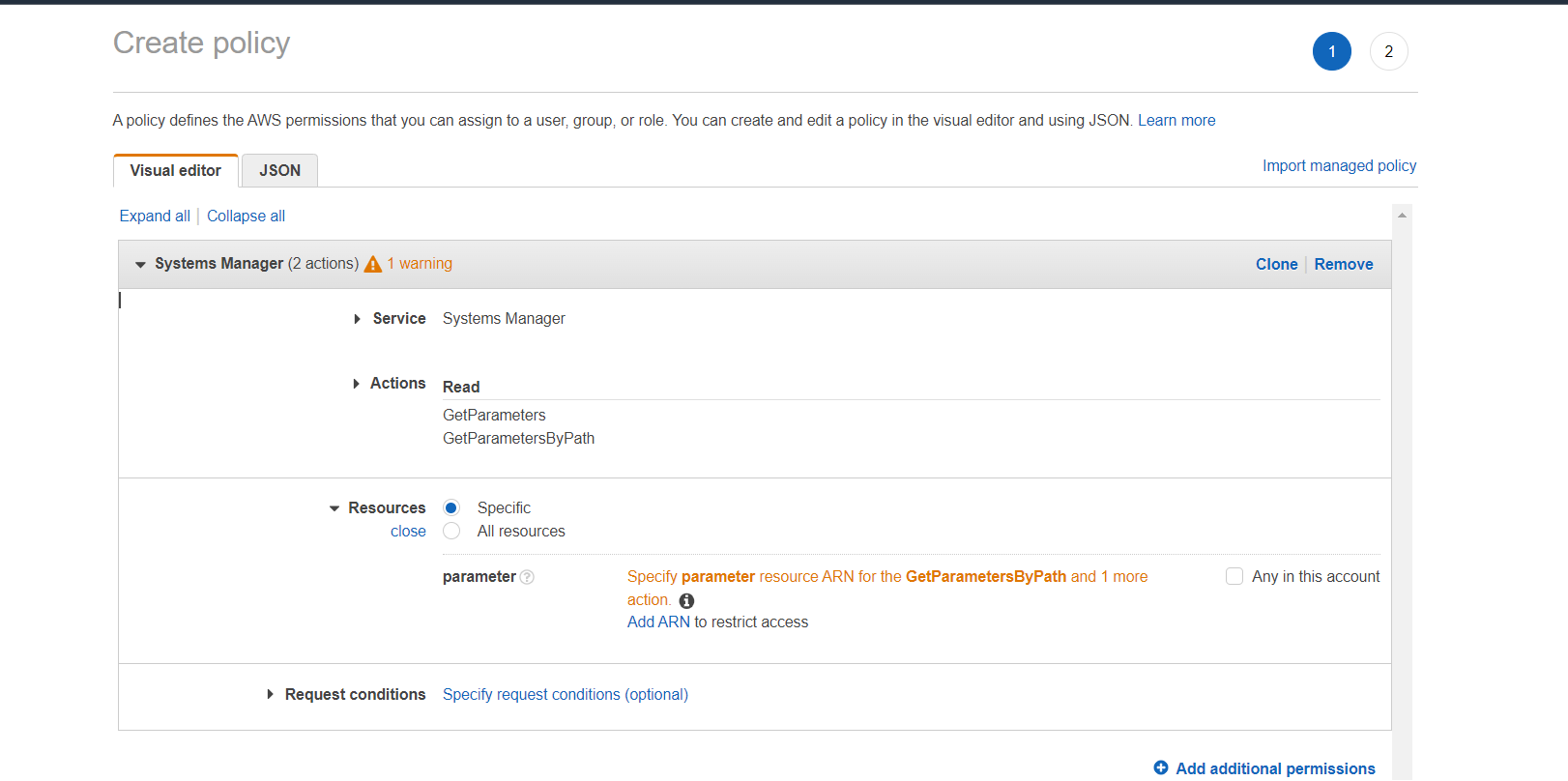
Lambda error



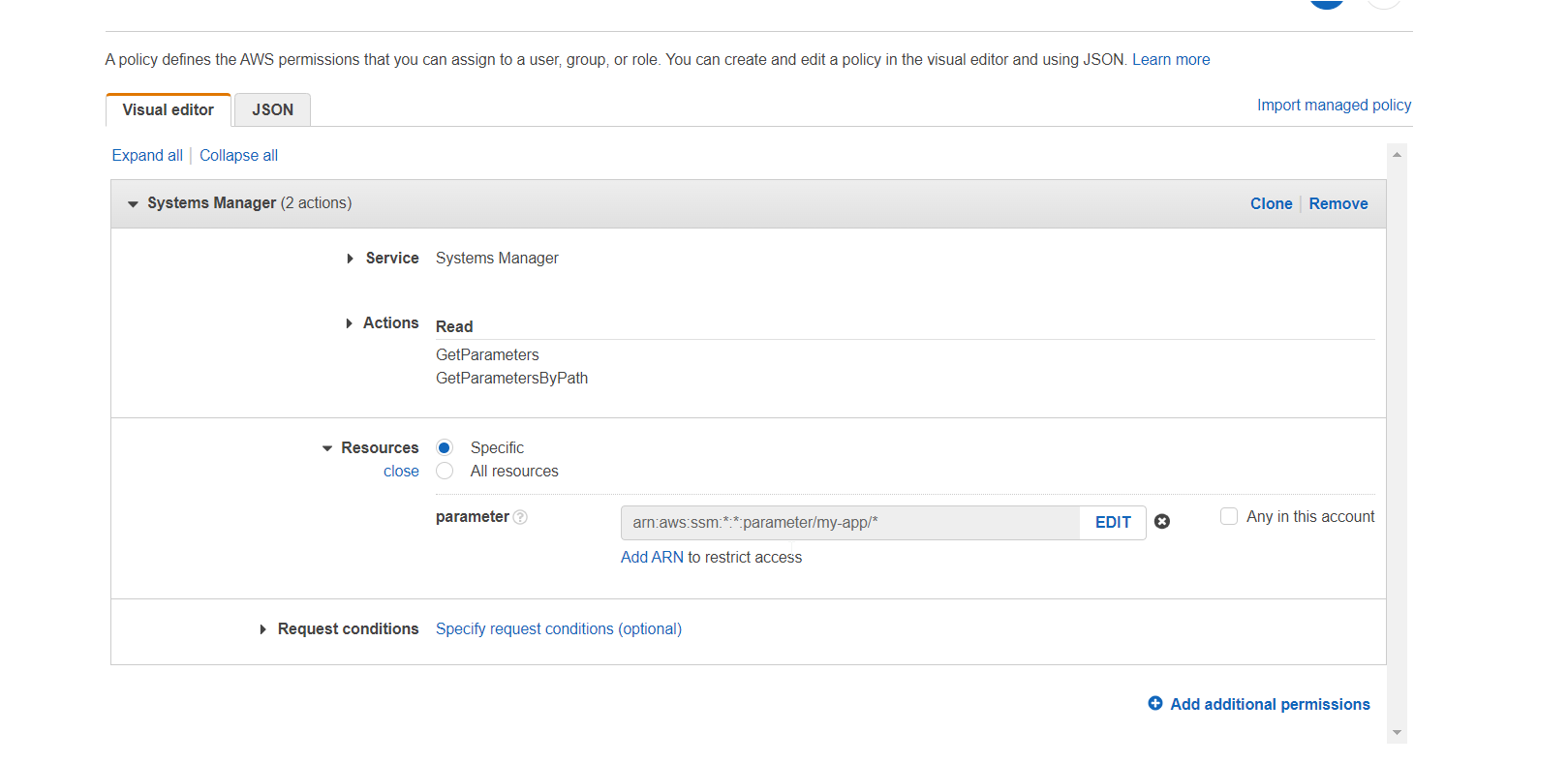


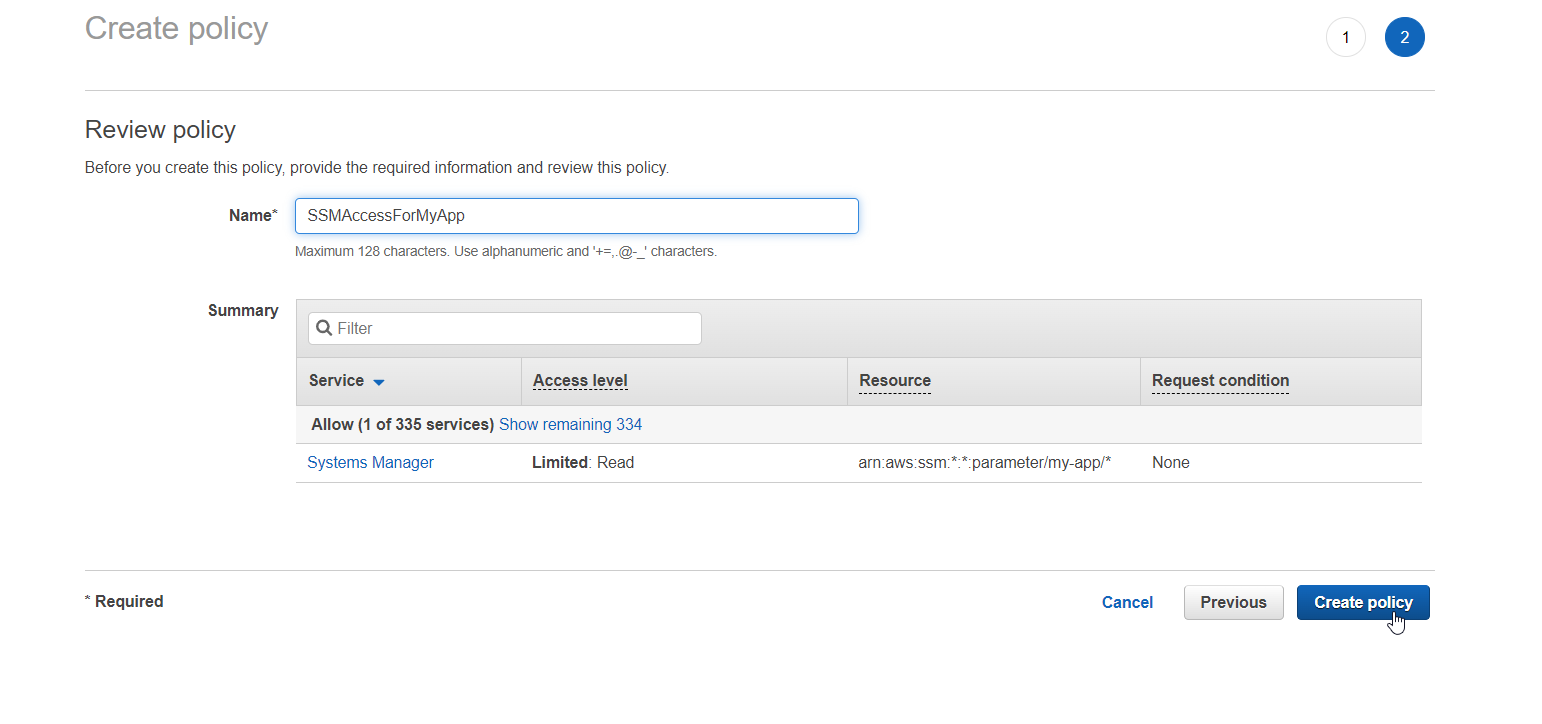
ADD IAM ROLE



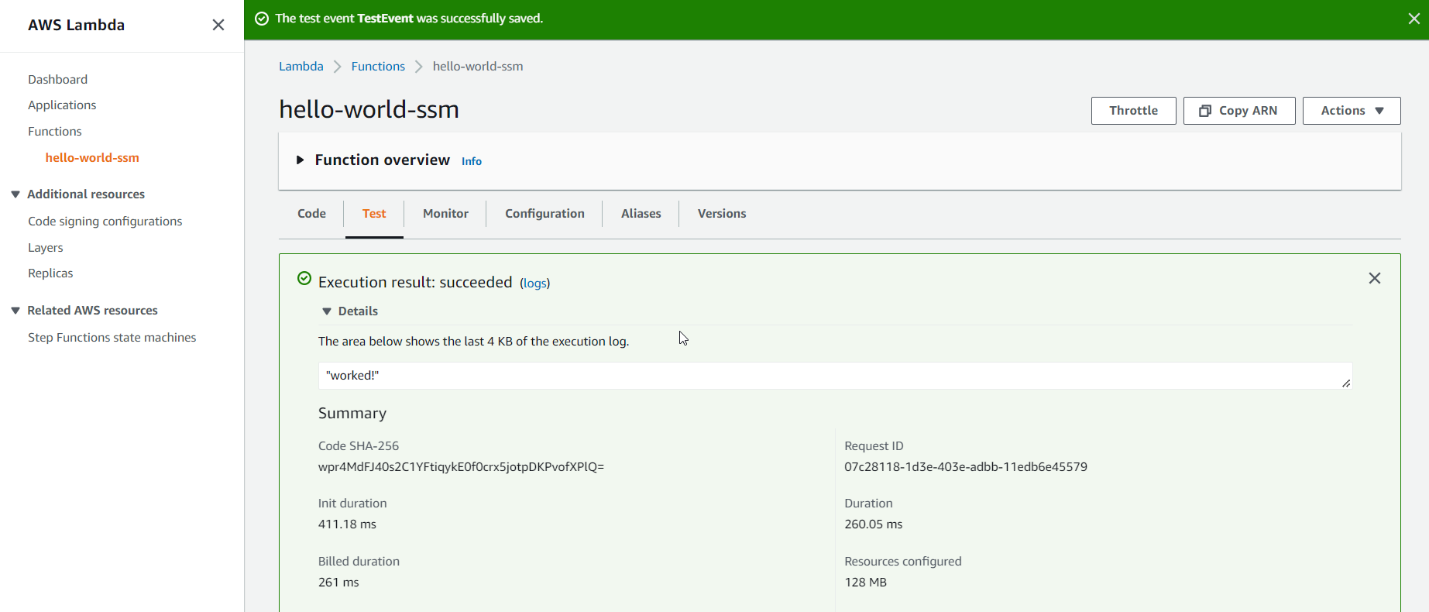




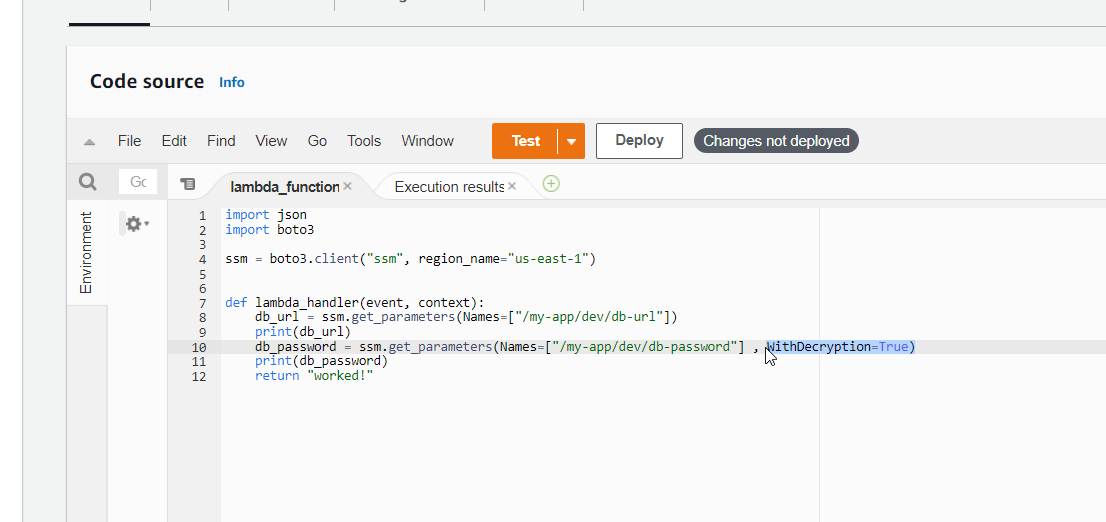




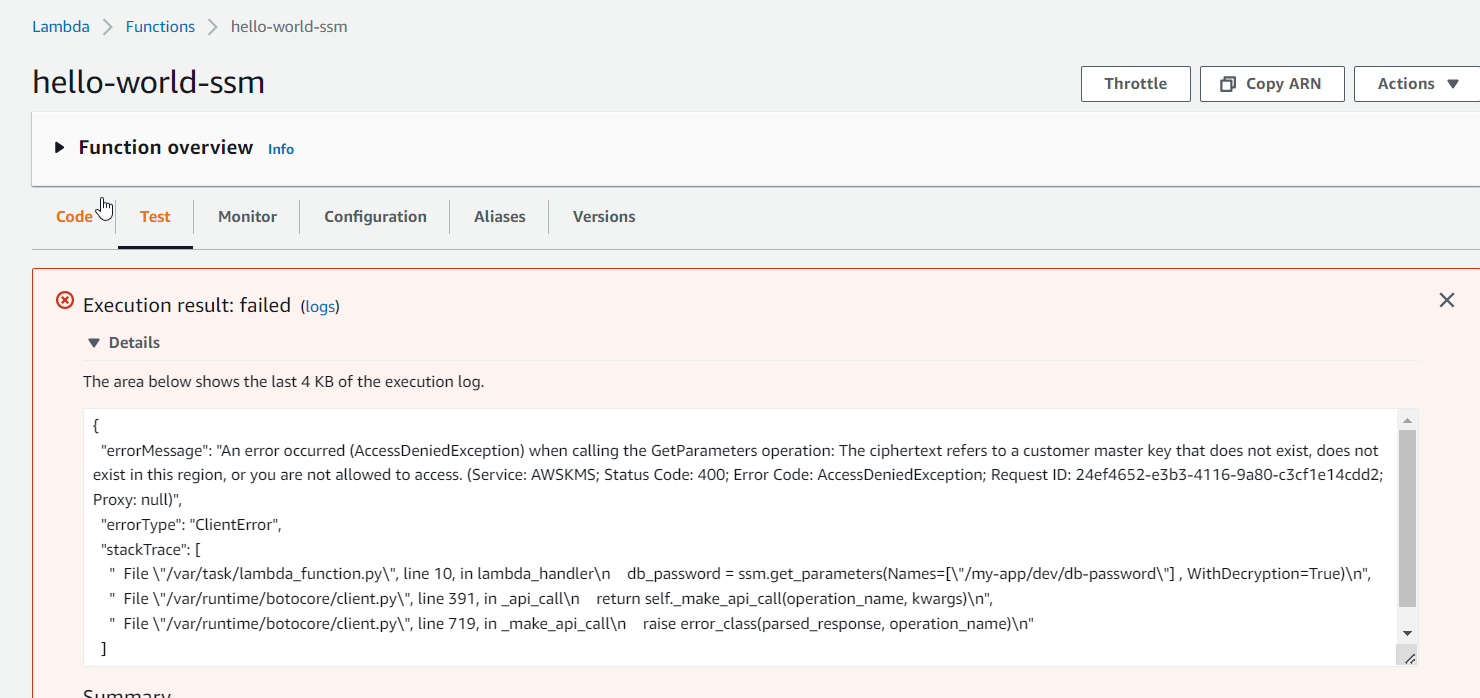
Go to lambda and test



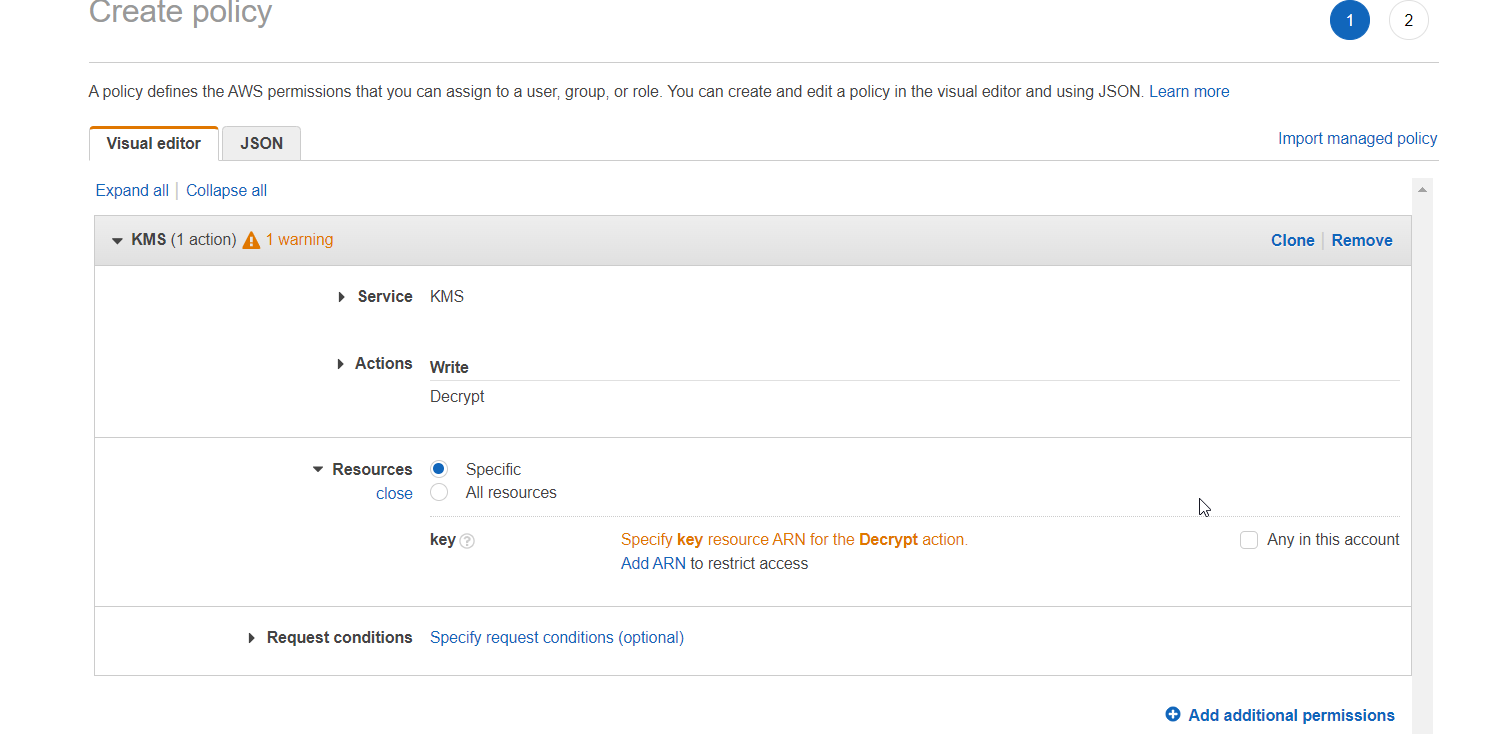
Run following code



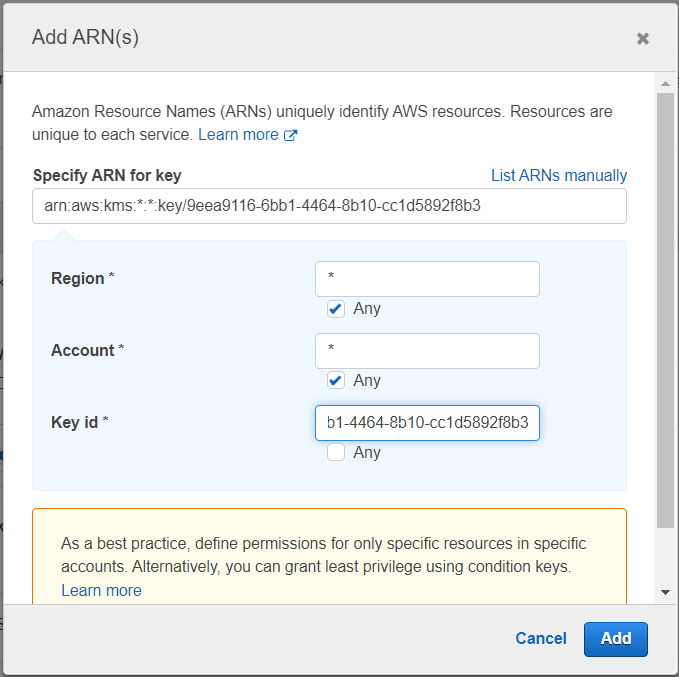
I got this error

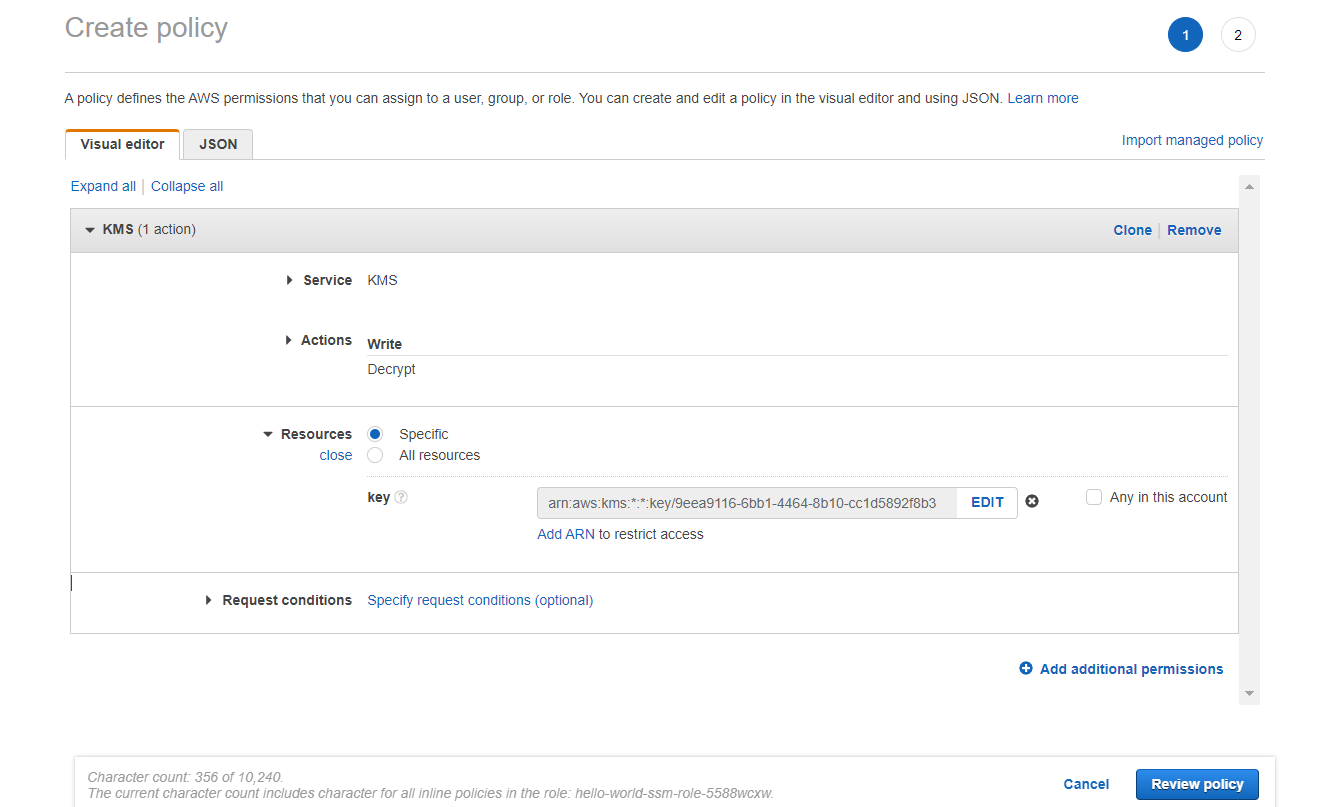


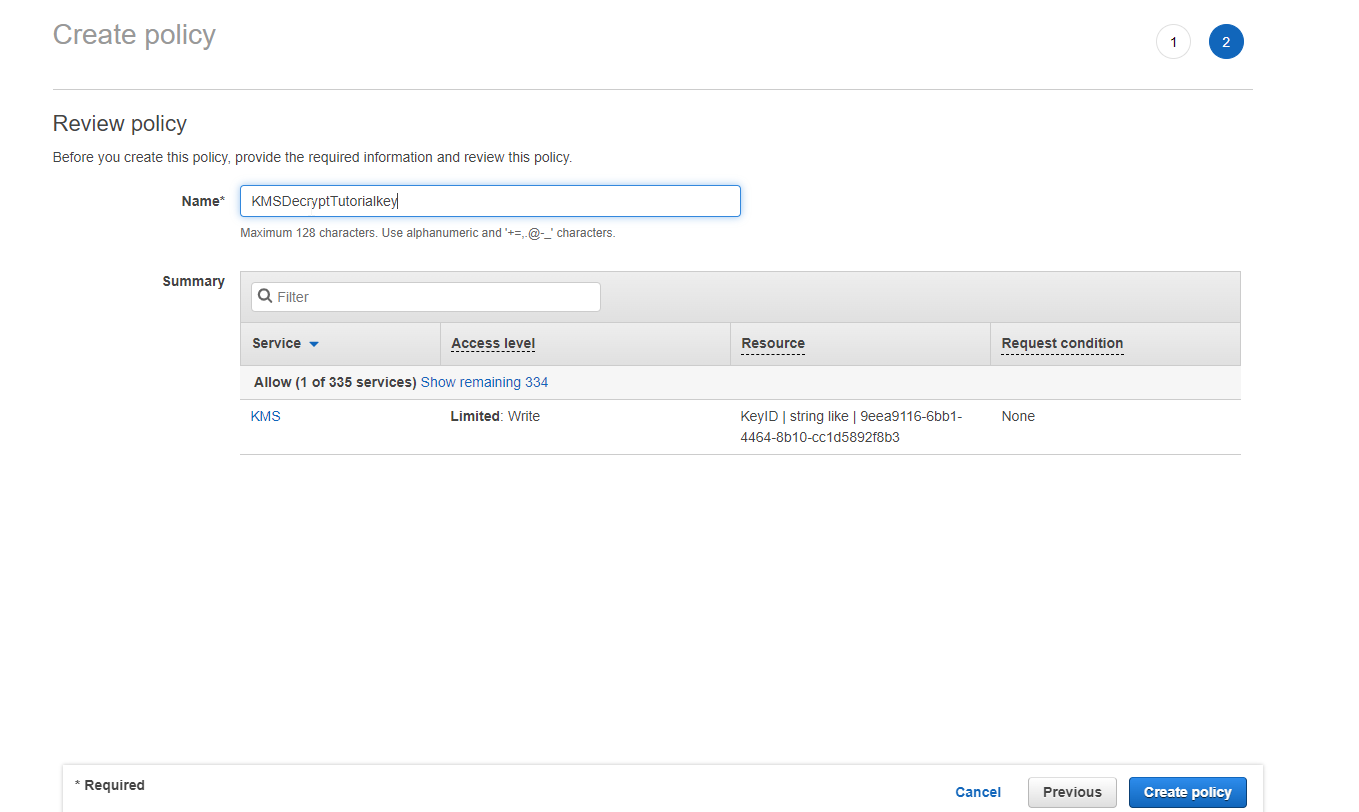
Go to IAM ATTACH KMS WRITE DECRYPT



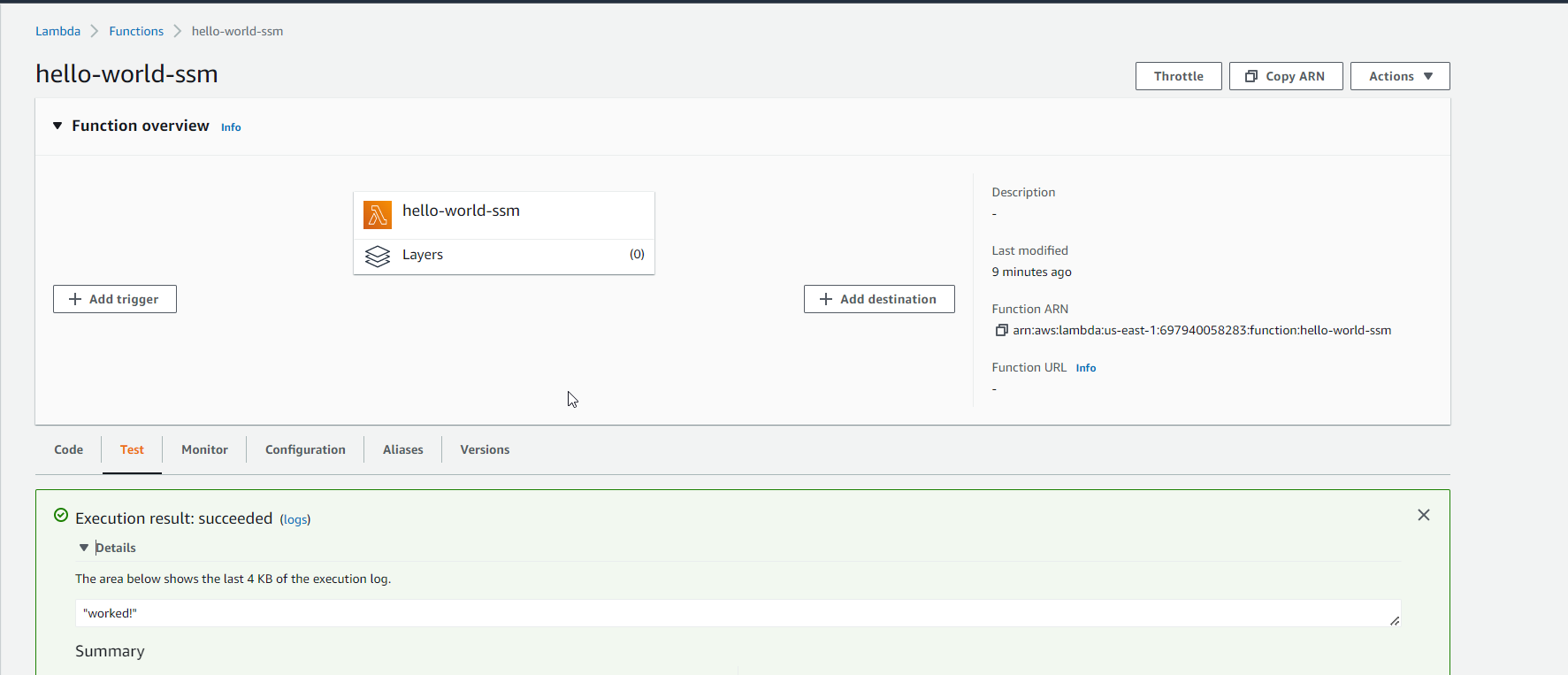




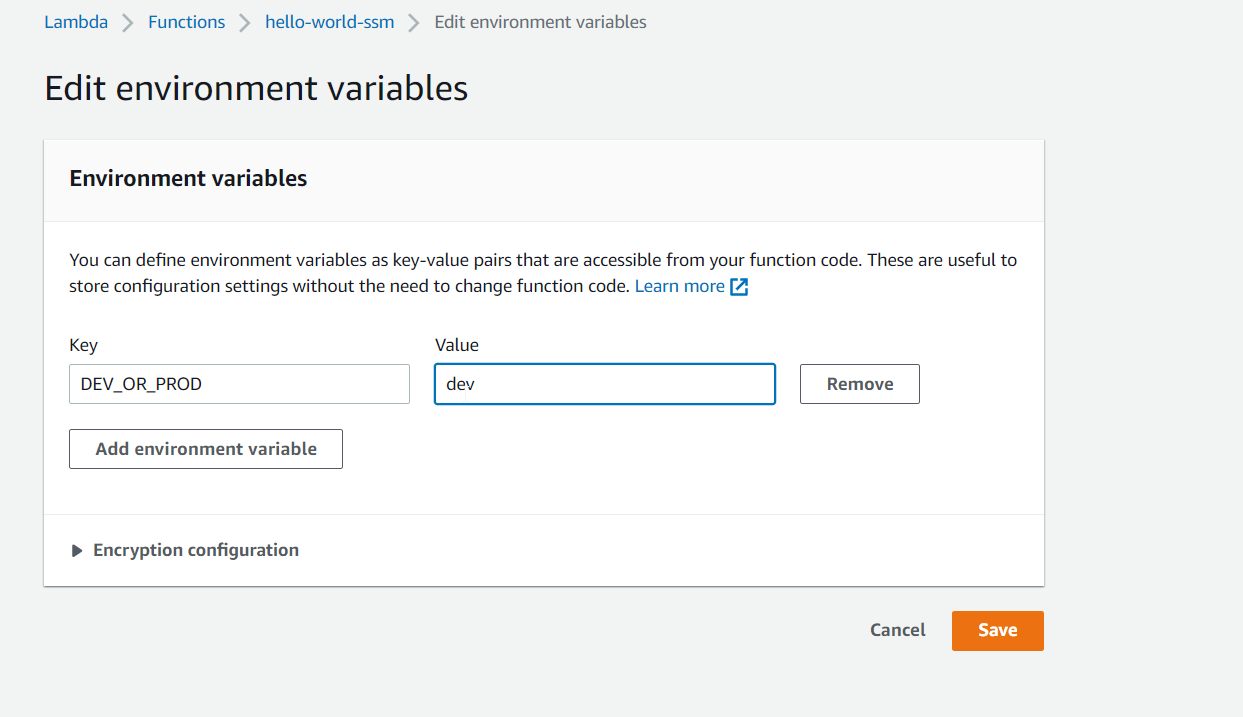




Now it works



Go to environment6 variable



import json

import boto3

import os

ssm = boto3.client('ssm', region\_name="us-east-1")

dev\_or\_prod = os.environ['DEV\_OR\_PROD']

def lambda\_handler(event, context):

    db\_url = ssm.get\_parameters(Names=["/my-app/dev/db-url" + dev\_or\_prod + "/db-url"])

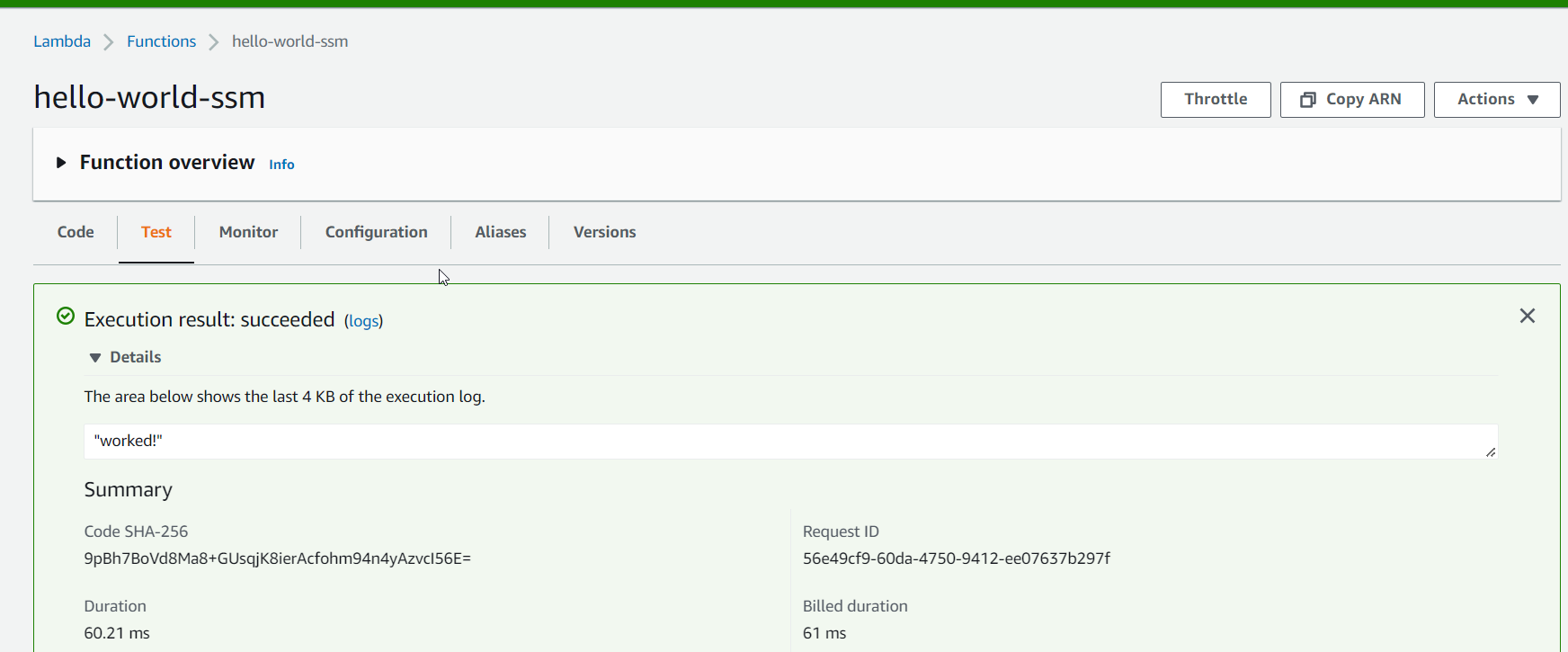
    print(db\_url)

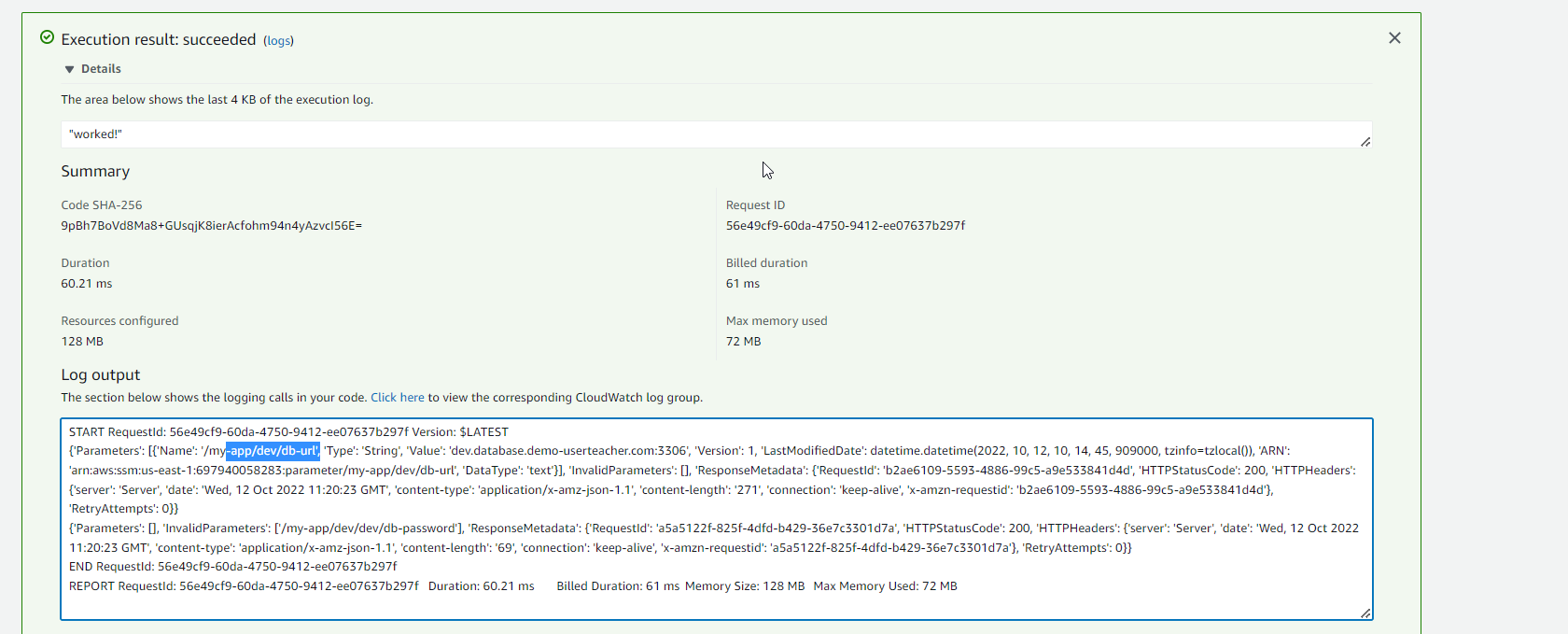
    db\_password = ssm.get\_parameters(Names=["/my-app/dev/db-password" + dev\_or\_prod + "/db-password"], WithDecryption=True)

    print(db\_password)

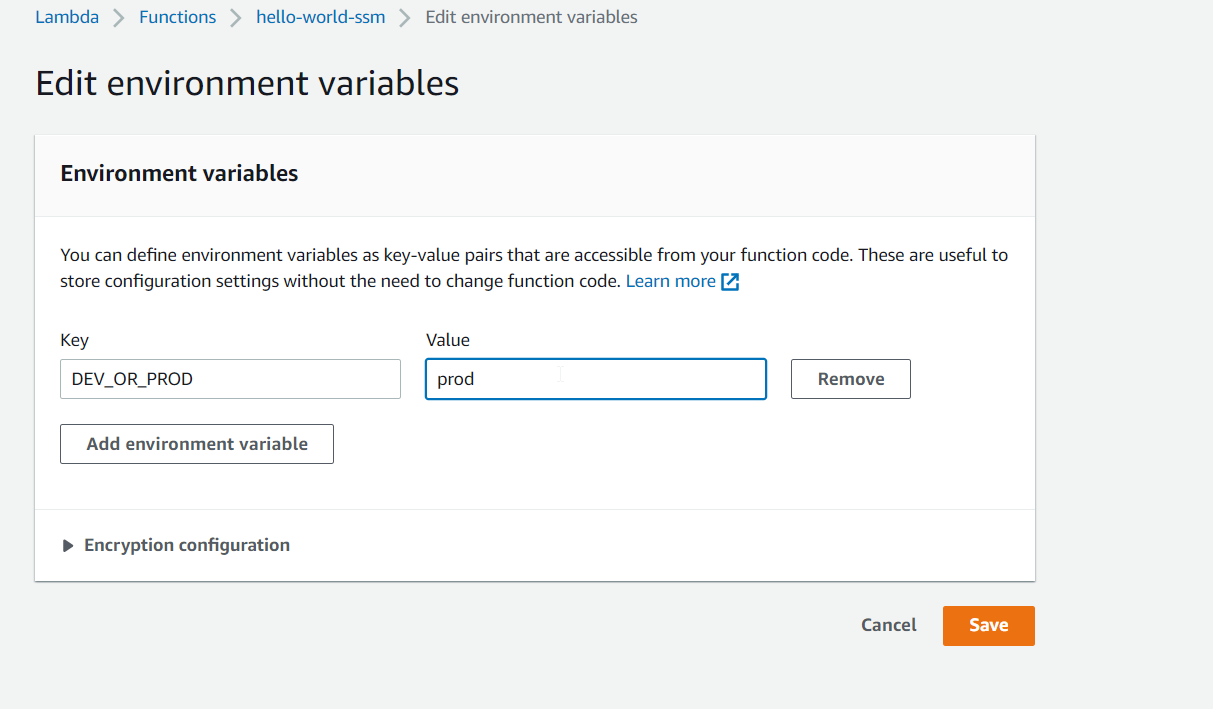
    return "worked!"

now

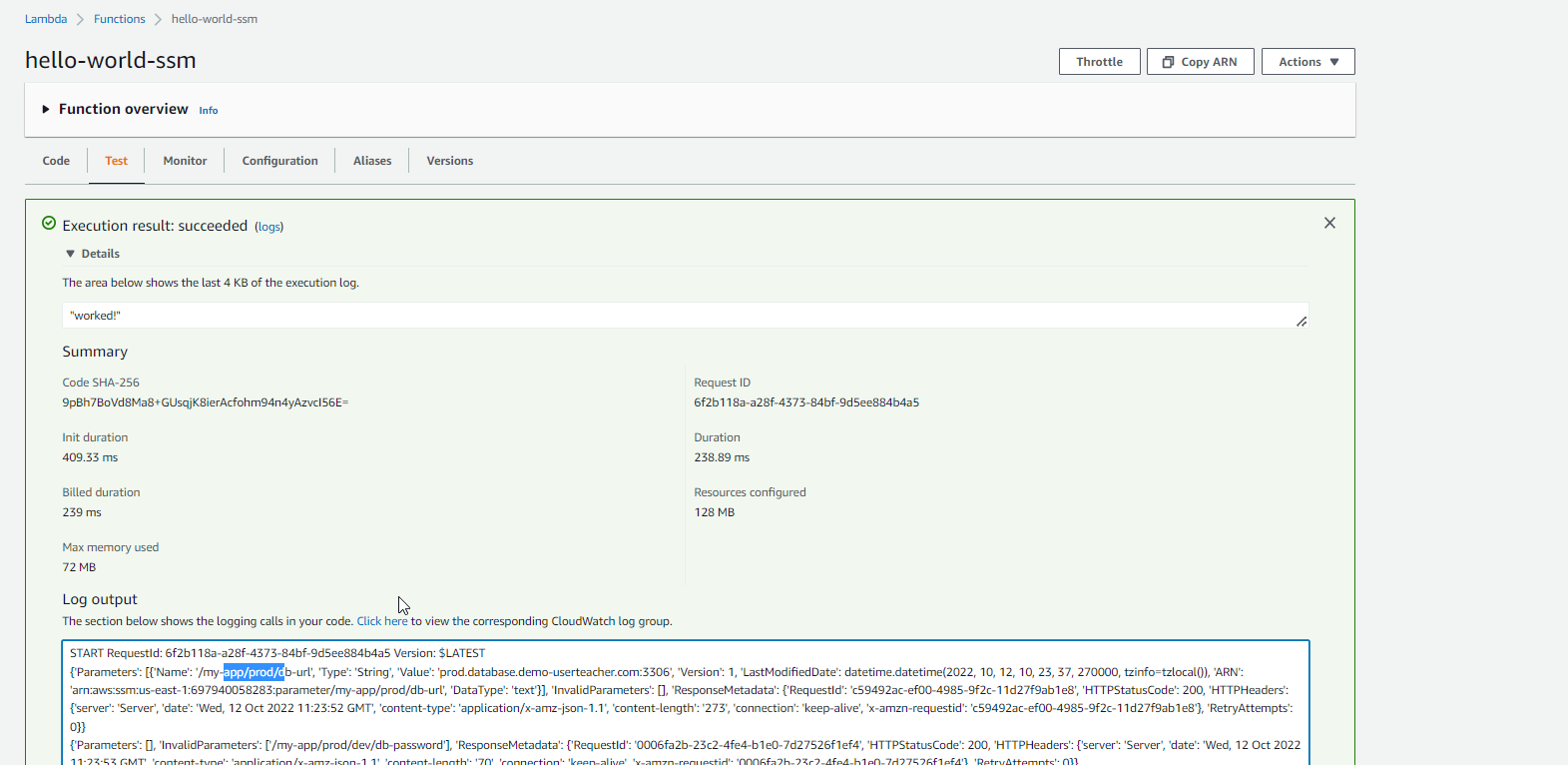




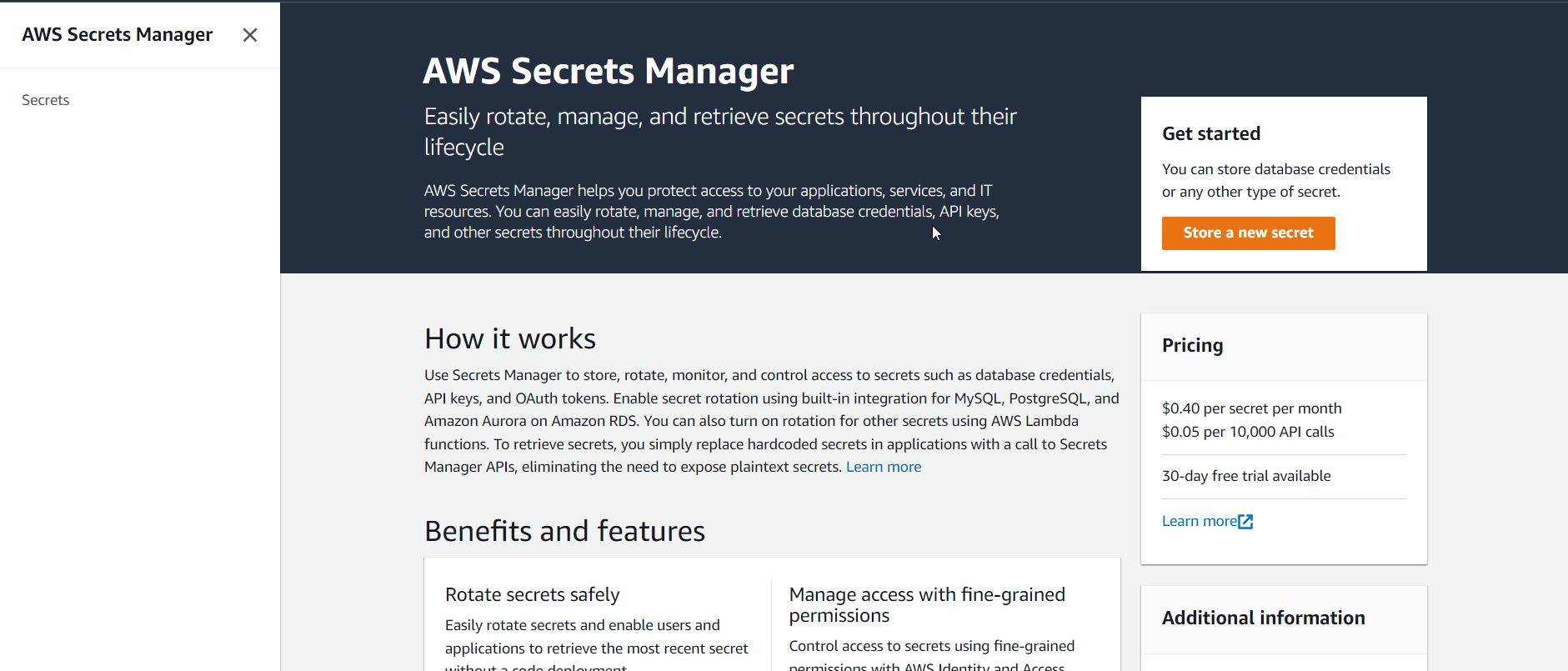
Again if we changed environment variable

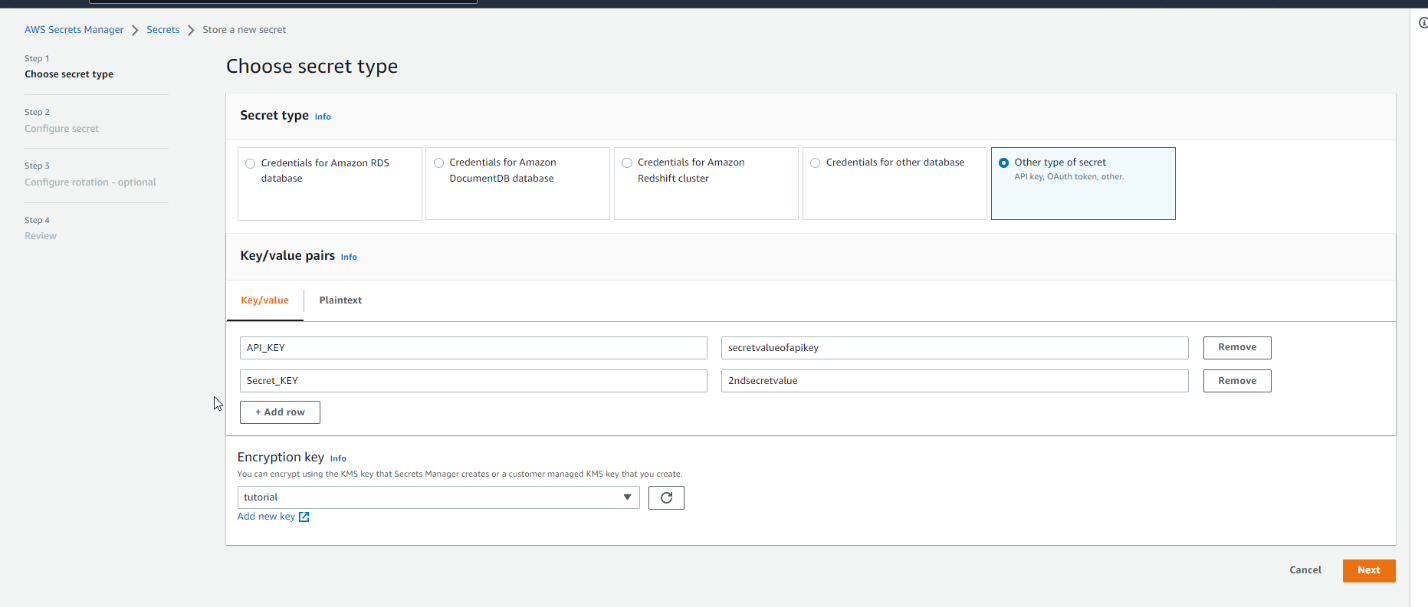


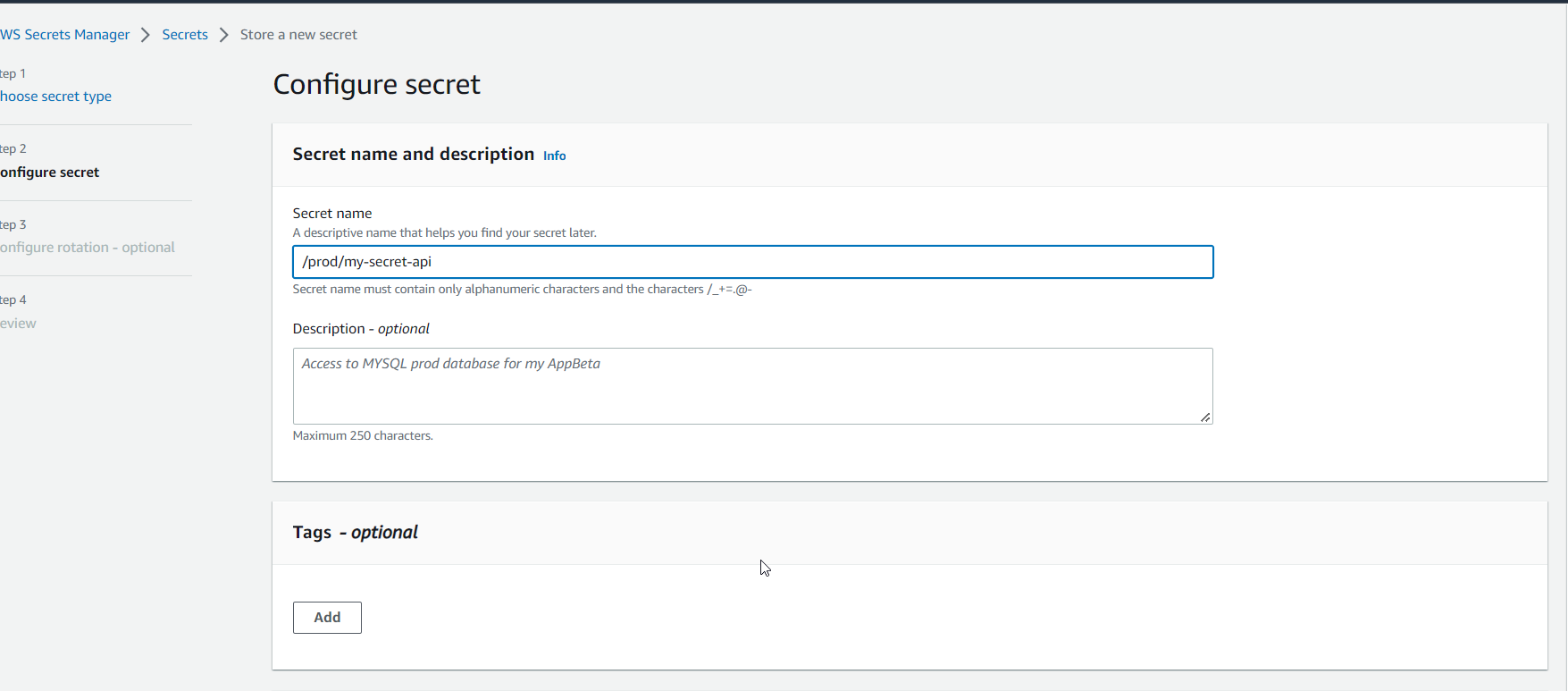
We get prod

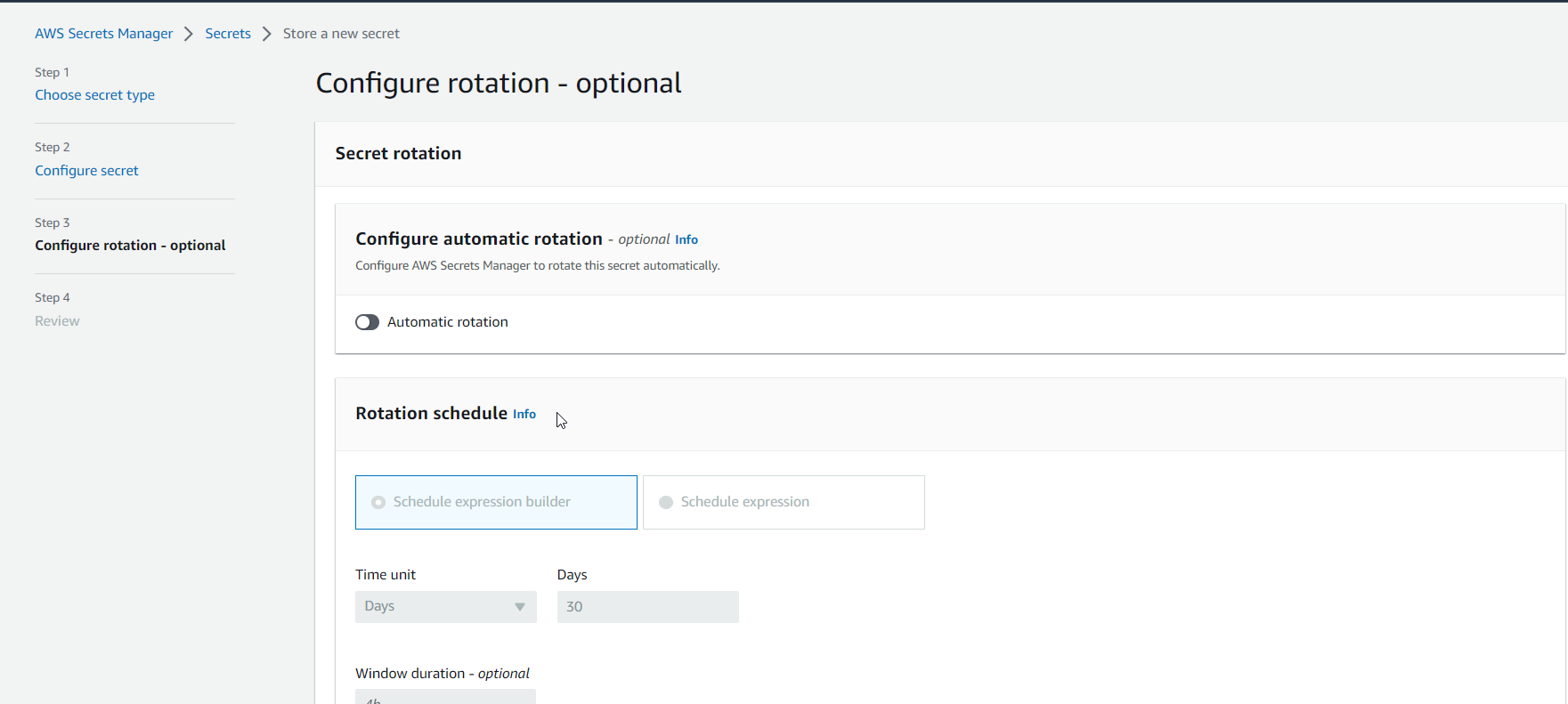


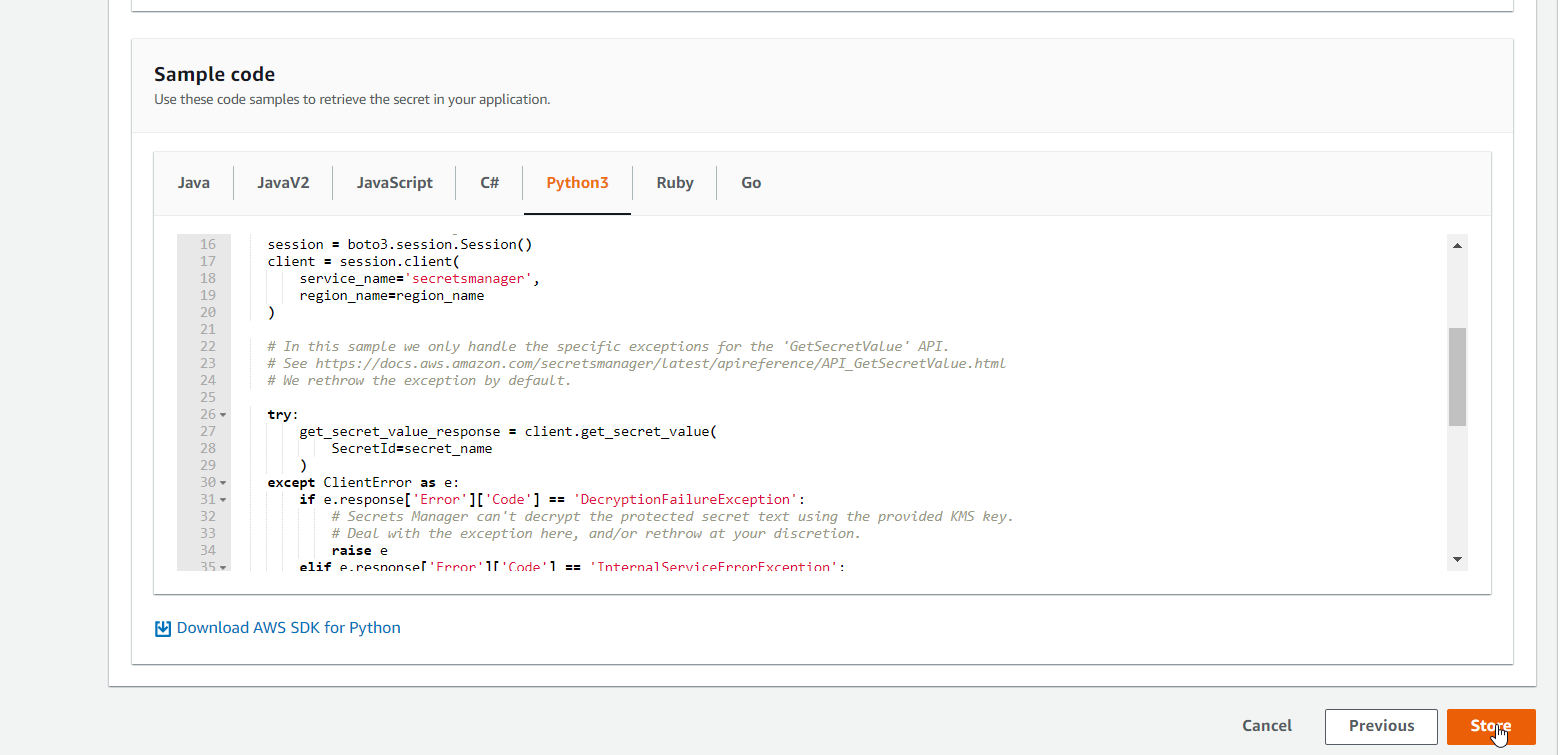
Secret manager

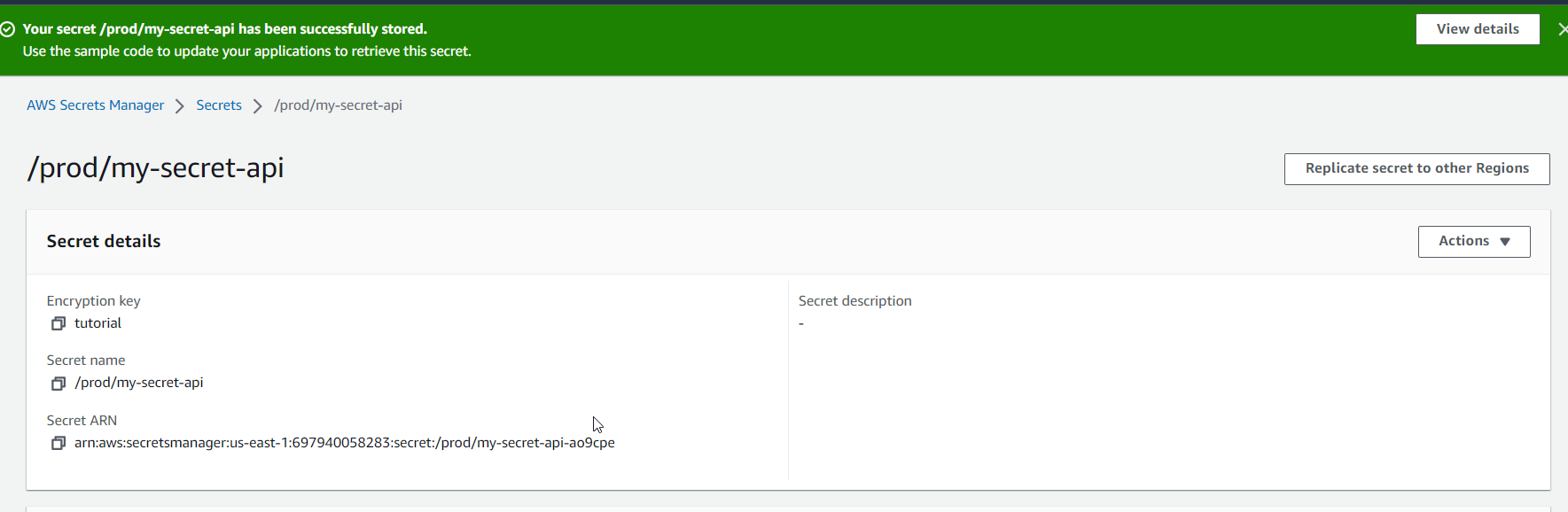












Clean up

