Estresar instancia EC2

Entremos a la intancia desde BASTION HOST

Instalamos las librerías

sudo dnf install stress

sudo dnf install stress-ng

ahora ponemos el siguiente comando

stress-ng --cpu $(nproc) --timeout 600s

Estresar ELB

Estresamos la instancia ec2 y además ejecutamos el script trafico.py

Debemos tener en cuenta que debemos cambiar la variable dns en py con el del load balancer

Estresar base de datos

Para estresar debemos estresar la base datos con el .py antes usado. Además hacemos lo siguiente. Dentro de la instancia EC2, entremos a la instancia RDS

mysql -h <RDS\_ENDPOINT> -u user -ppassword simple\_shop

Luego de esto ejecutamos los siguientes comandos

CREATE TABLE IF NOT EXISTS load\_test (id INT AUTO\_INCREMENT PRIMARY KEY, data TEXT);

INSERT INTO load\_test (data) VALUES ('test');

SELECT \* FROM load\_test ORDER BY RAND() LIMIT 1000;

Despues dentro de la instancia ec2, creamos el siguiente script. RECORDAR CAMBIAR EL DB\_HOST, POR EL ENDPOINT DE LA BASE DE DATOS

vim generar\_carga.sh

#!/bin/bash

# Configuración (reemplaza con tus valores)

DB\_HOST="stack-1-dbinstance-ywt4rnxjvny7.cr7ume2ammnb.us-east-1.rds.amazonaws.com"

DB\_USER="user"

DB\_PASSWORD="password"

DB\_NAME="simple\_shop"

DURACION=600 # 10 minutos en segundos

NUM\_CONEXIONES=20 # Número de conexiones paralelas

# Registrar hora de inicio

INICIO=$(date +%s)

# Función para ejecutar consultas continuamente

ejecutar\_consultas() {

while true; do

# Calcular tiempo transcurrido

AHORA=$(date +%s)

TRANSCURRIDO=$((AHORA - INICIO))

# Salir si ha pasado el tiempo especificado

if [ $TRANSCURRIDO -ge $DURACION ]; then

break

fi

# Ejecutar consultas costosas

mysql -h $DB\_HOST -u $DB\_USER -p$DB\_PASSWORD $DB\_NAME \

-e 'INSERT INTO load\_test (data) VALUES (CONCAT("Test data ", UUID(), " ", REPEAT("x", 500)));

SELECT COUNT(\*) FROM load\_test;

SELECT \* FROM load\_test ORDER BY RAND() LIMIT 100;

UPDATE load\_test SET data = CONCAT("Updated ", data) WHERE id IN (SELECT id FROM load\_test ORDER BY RAND() LIMIT 10);

DELETE FROM load\_test WHERE id IN (SELECT id FROM load\_test ORDER BY RAND() LIMIT 1);' 2>/dev/null

# Pequeña pausa aleatoria entre 0.1 y 0.5 segundos

sleep 0.$((RANDOM % 5 + 1))

done

}

# Crear tabla si no existe

echo "Creando tabla de prueba si no existe..."

mysql -h $DB\_HOST -u $DB\_USER -p$DB\_PASSWORD $DB\_NAME \

-e "CREATE TABLE IF NOT EXISTS load\_test (

id INT AUTO\_INCREMENT PRIMARY KEY,

data TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

INDEX idx\_created (created\_at)

) ENGINE=InnoDB;" 2>/dev/null

# Iniciar carga

echo "Iniciando prueba de carga por $((DURACION/60)) minutos con $NUM\_CONEXIONES conexiones paralelas..."

# Ejecutar consultas en paralelo

for i in $(seq 1 $NUM\_CONEXIONES); do

ejecutar\_consultas &

sleep 0.1 # Pequeño retraso entre inicio de conexiones

done

# Mostrar progreso

while true; do

AHORA=$(date +%s)

TRANSCURRIDO=$((AHORA - INICIO))

if [ $TRANSCURRIDO -ge $DURACION ]; then

break

fi

RESTANTE=$((DURACION - TRANSCURRIDO))

echo -ne "Tiempo transcurrido: $TRANSCURRIDO segundos | Tiempo restante: $RESTANTE segundos\r"

sleep 1

done

# Finalizar

echo -e "\nFinalizando todas las conexiones..."

pkill -f mysql

# Resumen final

echo "Prueba de carga completada después de $DURACION segundos"

echo "Resumen de la tabla load\_test:"

mysql -h $DB\_HOST -u $DB\_USER -p$DB\_PASSWORD $DB\_NAME \

-e "SELECT COUNT(\*) AS total\_registros,

MIN(created\_at) AS primera\_insercion,

MAX(created\_at) AS ultima\_insercion

FROM load\_test;" 2>/dev/null

DESPUES DE CREAR EL SCRIPT LO EJECUTAMOS

chmod +x generar\_carga.sh

./generar\_carga.sh