**PRACTICA MODULO 6**

**APARTADO E**

**Jorge Marcos Martos**

Para la presente practica se pide realizar tres scripts de *bash* para interactuar con AWS a través de *aws cli*.

Los tres scripts deberán iniciar tres procesos, el primero deberá comprobar si existe una cola de SQS, crearla si no existe y vaciarla si ya existe, para después mandar 100 mensajes. El segundo proceso deberá crear o destruir instancias según el flujo de trabajo de cada una de las instancias de EC2, la comprobación de la carga de trabajo de cada instancia se realizará cada 15 segundos, el tercer proceso eliminará los mensajes uno a uno cada 6 segundos.

**1. Primer Proceso:**

Código (primera parte):

Text

Description automatically generated

Para el primer proceso hemos definido una serie de funciones para luego ser ejecutadas con una función principal llamada ***main****,* el script del primer proceso lo hemos llamado ***send\_messages.sh***, este proceso creara una cola en caso de que no exista o la creara en caso de que no, luego la vaciara en caso de que este llena y mandara 100 mensajes a la cola de SQS.

La primera función que definimos es ***queue\_availability*** la cual comprueba que ya exista una cola SQS, en caso de que no exista esta función creara una, en caso de que exista se informara de que ya existe una con un mensaje, luego se guardara la URL de la cola creada o ya existente en la variable *queue\_url.*

La segunda función que hemos definido es ***messages****,* la cual manda 100 mensajesa la cola antes creada, usando la variable *queue\_url*.

La tercera función definida es ***check\_queue*** la cual comprueba cuantos mensajes hay en la cola, para ello se hace uso del comando *aws sqs get-queue-attributes* sobre nuestra cola y, usando *jq* para hacer consultas dentro del *json* dado, obtenemos el numero aproximado de mensajes que se encuentran en la cola y que capturaremos en la variable *number\_messages.*

Código (segunda parte):

Text

Description automatically generated

Definimos la función ***purge\_send\_messages*** la cual comprueba que la lista este vacía o no haciendo uso de la variable que nos daba de salida la función anterior (*check\_queue*), en caso de que contenga mensajes esta función vaciara la cola y acto seguido empezara a enviar los mensajes usando la función anteriormente definida para tal uso (*messages*).

Por ultimo, definiremos la función principal la cual hemos llamado ***main***. Esta función hará uso de las anteriores funciones en el siguiente orden: *queue\_availability, check\_queue, purge\_send\_messages.* Al final del script ejecutaremos esta función principal que iniciara el primer proceso.

**2. Segundo Proceso:**

Código (primera parte):

Text

Description automatically generated

El segundo proceso funciona de manera similar, se definirán las funciones necesarias para el proceso y se ejecutaran usando una función principal que use el resto de funciones en un orden especifico, iniciando así el segundo proceso, el cual se ha llamado ***check\_create\_instances.sh.*** Este proceso creara y destruirá instancias EC2 según el numero de mensajes que haya en ese momento en la cola. Se trata de hacer una simulación en la que se simule un aumento de la carga de trabajo y que se produzca un escalado de instancias de forma automática. Para esta simulación se presupone que cada instancia puede procesar 10 mensajes de manera optima, y que solo puede haber un máximo de 5 instancias.

Empezamos el script obteniendo la URL de la cola que creamos o seleccionamos durante el primer proceso y guardándola en la variable *queue\_url*, seguidamente declararemos otra variable donde guardaremos el identificativo de la AMI de Linux elegible para usar gratis de AWS, esta variable es *ami\_id*.

La primera función es similar a una que definimos es del proceso anterior, que aquí hemos llamado ***check\_messages***, esta función capturara en una variable el numero de mensajes que tenga la cola en ese momento (para este proceso se usara esta función cada 15 segundos).

La siguiente función que definimos es ***launch\_instance***, la cual se encarga de lanzar una instancia *t2.micro* a través del comando *aws ec2 run-instances.* Este comando usara también la id de la AMI que anteriormente guardamos en la variable *ami\_id*.

La función que definimos a continuación es ***destroy\_instance***, esta tomara la identificación de la ultima instancia lanzada y la guardara en la variable *instance\_id*, a continuación, se usara el comando *aws ec2 terminate-instances* usando el identificativo de la instancia previamente capturado.

Después definimos una función que nos dice el numero de instancias funcionando (con estatus *running*) en el momento que usamos la función, el numero de instancias se guarda en la variable *running\_instances.*

Código (segunda parte):

Text

Description automatically generated

Por ultimo, definimos la función principal que ejecutara todo el proceso a la cual hemos llamado ***main***, esta función usara un bucle *while* en el que tendrá en cuenta el numero de mensajes en la cola, el bucle seguirá corriendo mientras que haya mensajes en la cola. A continuación usara las funciones *check\_messages* y *check\_instances* que se iniciaran con cada bucle para ver cuantos mensajes e instancias hay en ese momento, nótese que entre un bucle y otro pasan 15 segundos, como puede verse al final del bucle en el cual tenemos el comando *sleep 15.*

Una vez obtenido el numero de mensajes y el numero de instancias se creara la variable *threshold* el cual nos dará el umbral que decidirá si se tienen que crear o destruir instancias, para ello se seguirá la formula *((numero de mensajes + 9)/10 – numero de instancias funcionando).*

Después usaremos este umbral y el numero de instancias para que en el caso de que el umbral sea mayor a 0 (y mientras el numero de instancias no sea mayor a 5) se lance una instancia y en el caso de que el umbral sea menor a 0 se destruya una instancia con cada bucle.

**3. Tercer Proceso**

Código:

Text

Description automatically generated

El tercer proceso se resolverá usando un bucle *while* que espere 6 segundos entre una ejecución y otra, una vez iniciado el bucle este chequeara si hay mensajes que recibir a través de la variable *msg* que usara el comando *aws sqs receive-message* para ello. Si ha recibido algún mensaje este se eliminará usando el comando *aws sqs delete-message* para eliminarlo, lo normal es que entre espera y espera de 6 segundos reciba mas de un mensaje, lo que el comando de *aws cli* hará será eliminar el ultimo.

**4. Resultados**

Una vez iniciados los tres procesos estos funcionaron perfectamente, se crearon no mas de 5 instancias y se acabaron eliminando todas cuando ya no había mas mensajes que borrar. El proceso tomo 10 minutos en ser ejecutado, en la siguiente captura podemos ver parte del proceso:

Text

Description automatically generated with medium confidence

Estas son las 5 instancias que creo y termino durante el proceso:

Graphical user interface, application

Description automatically generated