# PRACTICA 5

COMPUTACIÓN EN LA NUBE

### Contenido

INTRODUCCIÓN	2
ACTIVIDADES	3
Implemente su aplicación de ejemplo en el lenguaje deseado y despliéguela en amazon (contenedor o instancia)	3
2. Divida su aplicación en tres aplicaciones distintas (fA, fB, fC) y modifiquelas para que cada aplicación le pase su resultado a la siguiente usando colas (e.g. SQS o redis) y desplieguelas en AWS. En el ejemplo anterior 'x' e 'y' se mandarian por colas	
3. Configure un 'topic' en AWS SNS para que fA se subscriba y tome su dato de entrada de ahi. En el ejemplo anterior, fA recibiría 'w' de un topic	8
ESTIMACIÓN DE GASTOS1	2

## INTRODUCCIÓN

A lo largo de esta práctica se va a tener una primera toma de contacto con el sistema de colas SQS proporcionado por AWS.

#### **ACTIVIDADES**

1. Implemente su aplicación de ejemplo en el lenguaje deseado y despliéguela en amazon (contenedor o instancia)

La aplicación seleccionada ha sido la misma que se daba de ejemplo. En esta aplicación se pueden diferenciar 3 funciones que basicamente hace las funciones de una "cola" ya que se le pasa el dato de una función a la siguiente.

```
import time
def fA(DataA):
    print(DataA)
    time.sleep(5)
    return(DataA+"A")

def fB(DataB):
    print(DataB)
    time.sleep(3)
    return(DataB+"B")

def fC(DataC):
    print(DataC)
    time.sleep(4)
    return(DataC+"C")

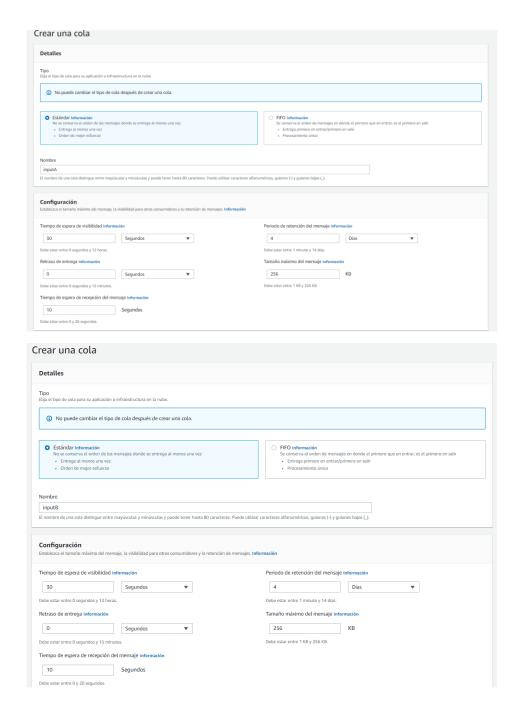
w = _Inicio:'
x = fA(w)
y = fB(x)
z = fC(y)
print(z)
```

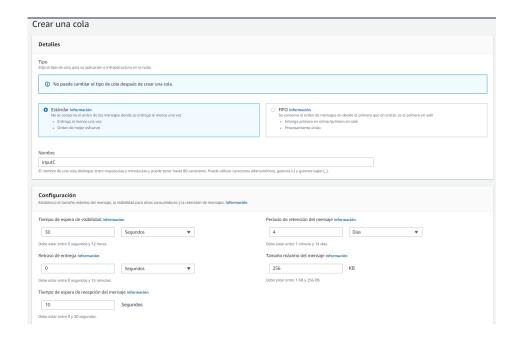
```
nikhil@DESKTOP-M2DV3PQ: ~

nikhil@DESKTOP-M2DV3PQ:~$ python3 practica5.py
Inicio:
Inicio:A
Inicio:AB
Inicio:ABC
```

2. Divida su aplicación en tres aplicaciones distintas (fA, fB, fC) y modifiquelas para que cada aplicación le pase su resultado a la siguiente usando colas (e.g. SQS o redis) y desplieguelas en AWS. En el ejemplo anterior 'x' e 'y' se mandarian por colas

En este caso como hay que diferenciar 3 funciones, se procede a crear 3 colas con nombres inputA, inputB e inputC con las siguientes configuraciones.





Posteriormente, se deben actualizar las credenciales en la carpeta .aws. Para ello se procede a copiar todo el contenido proporcionado por el Lab AWS relacionado a las credenciales y se pega en el archivo credentials.

```
nikhil@DESKTOP-M2DV3PQ: ~/.aws

nikhil@DESKTOP-M2DV3PQ: ~$ cd .aws

nikhil@DESKTOP-M2DV3PQ: ~/.aws$ ls

config credentials

nikhil@DESKTOP-M2DV3PQ: ~/.aws$ vim credentials

nikhil@DESKTOP-M2DV3PQ: ~/.aws$ _
```

Como no están instaladas una serie de herramientas esenciales para esta práctica, se procede a instalarlas con las siguientes instrucciones a ejecutar:

- Sudo apt update
- Sudo apt install Python3-pip
- python3 -m pip install boto3

Luego se deben crear una serie de scripts para vincularlos con las colas.

En primer lugar, se crea un script con nombre inputA, esto va a indicar básicamente el dato a insertar en primer lugar antes de comenzar con el paso de mensajes entre colas. En esta función simplemente se escribe el mensaje inicial "Inicio:"

```
import boto3

if __name__ == "__main__":
    sqs = boto3.resource('sqs', region_name='us-east-1')
    input_queue = sqs.get_queue_by_name(QueueName='inputA')
    response = input_queue.send_message(MessageBody='Inicio:')
```

Luego se crea un script que mande los mensajes de una cola a la siguiente, en este caso como se han creado 3 colas A, B y C pues se van a relacionar las primeras dos colas. Los nombres de las colas vienen especificados en el campo QueueName.

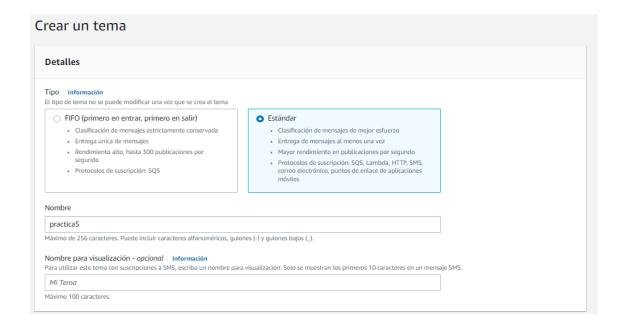
Este último script cambia un poco ya que no hay una cola Output, solo Input. Esto se debe a que es la última cola a tratar.

```
    nikhil@DESKTOP-M2DV3PQ: ~
                                                                                                   nport boto3
    name
        input_queue = sqs.get_queue_by_name(QueueName='input
while(True):
             for message in input_queue.receive_messages():
                 message_body = message.body
                  new_message = message_body +
                 print(f
                  if message.delete():
                      print('
 nikhil@DESKTOP-M2DV3PQ; ~
 ikhil@DESKTOP-M2DV3PQ:~$ python3 inputA.py
nikhil@DESKTOP-M2DV3PQ:~$ python3 colaA.py
Mensaje recibido: Inicio:
Mensaje a enviar: Inicio:A
Message deleted successfully
[7]+ Stopped python3 colanikhil@DESKTOP-M2DV3PQ:~$ python3 colaB.py
                                  python3 colaA.py
Mensaje recibido: Inicio:A
Mensaje a enviar: Inicio:AB
Message deleted successfully
[8]+ Stopped
                                  python3 colaB.py
            TOP-M2DV3PQ:~$ python3 colaC.py
Mensaje recibido: Inicio:AB
Mensaje a enviar: Inicio:ABC
Message deleted successfully
[9]+ Stopped
                                  python3 colaC.py
 ikhil@DESKTOP-M2DV3PQ:~$ _
```

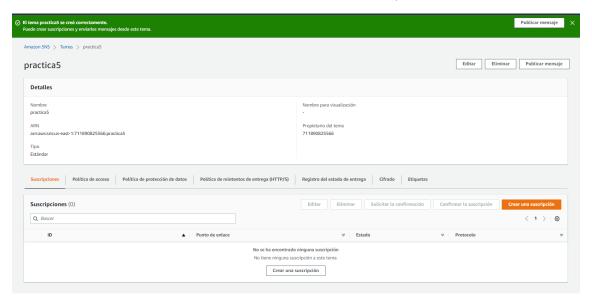
Una incidencia que ocurrió fue que el código de Python se queda ejecutando indefinidamente por ello siempre hay que terminar el programa con el uso de los comandos "CTRL + Z". Esto se puede solucionar introduciendo un "break" al final de cada script para romper el bucle while.

3. Configure un 'topic' en AWS SNS para que fA se subscriba y tome su dato de entrada de ahi. En el ejemplo anterior, fA recibiría 'w' de un topic.

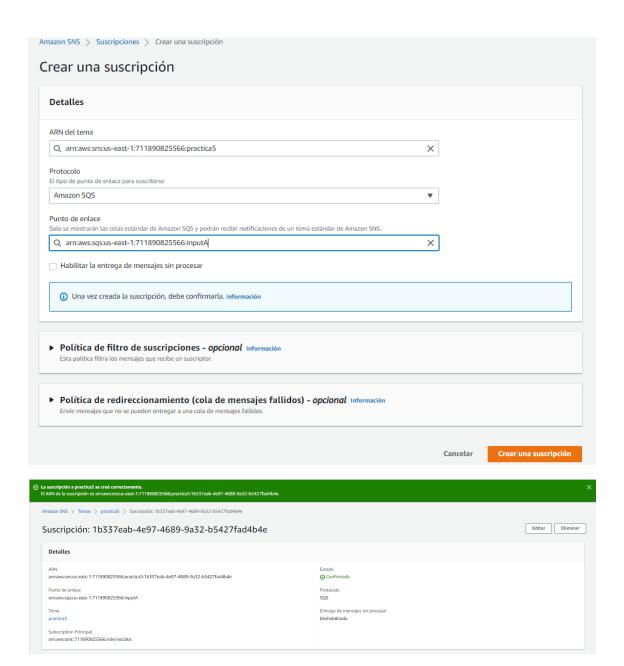
En primer lugar, se crea el tema al que se debe suscribir posteriormente. El tipo de tema es el estándar y se le asigna un nombre simbólico.



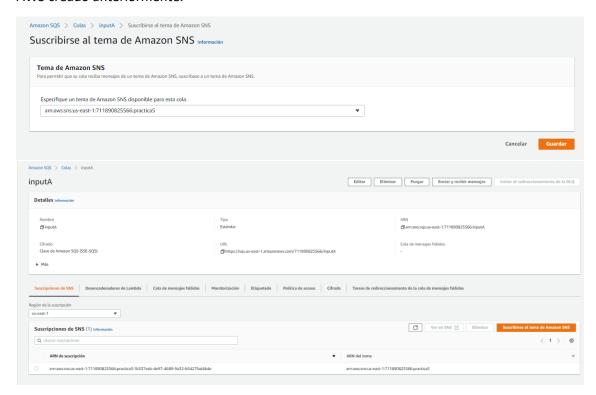
Posteriormente se debe seleccionar el botón de crear una suscripción.



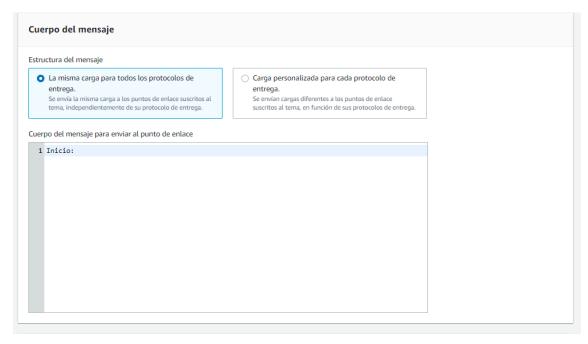
Para crear una suscripción de una cola, se debe seleccionar el protocolo SQS y después elegir la primera cola, puesto a que va a ser la primera encargada de pasar los mensajes.



Luego se accede a dicha cola, en este caso inputA y se procede a realizar la suscripción al tema AWS creado anteriormente.



Por último, se introduce el mensaje que se desea pasar por la primera cola, en este caso el mensaje es "Inicio:"



Al ejecutar la cola, ya se ve en el JSON la etiqueta "Message" que hace referencia al mensaje que se ha introducido anteriormente. Y al final del JSON se ve el respectivo mensaje A que es el que le pertenece mandar a esta cola.

### ESTIMACIÓN DE GASTOS

Como se va a usar la instancia más simple que es t2.micro, el coste de esta instancia es de 0.0116\$/h. Esto al mes se traduce como 8.35\$.

El coste EBS es de 0.10\$ GB/mes. Por lo cual al utilizar 8GB se consumen 0.80\$ GB/mes.

El coste mensual de esto se queda en 9.15\$/mes.

Escalándolo a un año de uso se quedaría en 109.8\$.

Lo bueno de esta práctica es que utiliza Amazon SQS lo cual es gratis. (Cabe destacar que es gratis hasta el primer millón de consultas mensuales, pero se trata de una cifra muy dificil de alcanzar).