Investigación SOA

Nombre: Ricardo Quintana

NRC: 003

Fecha: 07/06/2024

¿Qué es SOA?

Arquitectura Orientada a Servicios (SOA): Es un enfoque de diseño de software en el cual los componentes se desarrollan como servicios independientes que pueden interactuar a través de una red. Los servicios son reutilizables, interoperables y pueden ser descubiertos y utilizados por otros servicios, facilitando la escalabilidad y flexibilidad del sistema.

¿Qué es RabbitMQ?

RabbitMQ: Es un gestor de colas de mensajes de código abierto que permite la mensajería asíncrona entre sistemas. Funciona como un intermediario que recibe mensajes de productores y los entrega a consumidores, soportando varios protocolos como AMQP, MQTT y STOMP. RabbitMQ proporciona alta disponibilidad y persistencia de mensajes.

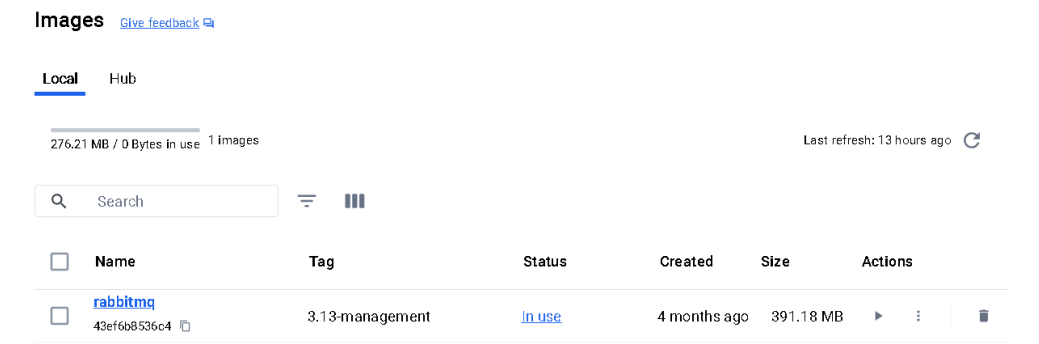
¿Por qué y para qué se usa RabbitMQ?

- Desacoplamiento de Componentes: Permite que los componentes se comuniquen sin depender de la disponibilidad inmediata de otros.
- Procesamiento en Lotes: Acumula tareas en una cola para procesarlas por lotes.
- Balanceo de Carga: Distribuye la carga de trabajo entre múltiples consumidores.
- Aseguramiento de Entrega: Garantiza la entrega de mensajes.
- Integración de Sistemas Heterogéneos: Facilita la comunicación entre sistemas y tecnologías diferentes.

Desarrollo:

1. **Instalación y Configuración de RabbitMQ**:

- **Docker**: Utilizamos Docker para ejecutar RabbitMQ.
- **Imagen de RabbitMQ**: Descargamos y ejecutamos la imagen `rabbitmq:3.13-management` con Docker para facilitar la gestión de colas a través de una interfaz web.
- **Pika**: Se instala el modulo pika con el siguiente comando "**pip install pika"**



2. **Desarrollo del Productor (Producer)**:

- Creación de un script en Python (`producer.py`) que envía mensajes a una cola de RabbitMQ.
- Uso de la librería `pika` para interactuar con RabbitMQ.

C: > Users > holaq > OneDrive > Documents > producer.py > ...

```python
import pika

def send_message(message):
    try:
        # Conectar a RabbitMQ
        connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
        channel = connection.channel()

        # Declarar una cola
        channel.queue_declare(queue='email_queue')

        # Enviar el mensaje
        channel.basic_publish(exchange='', routing_key='email_queue', body=message)
        print(f" [x] Sent '{message}'")

        # Cerrar la conexión
        connection.close()
    except pika.exceptions.AMQPConnectionError as e:
        print(f" [x] Could not connect to RabbitMQ: {e}")

if __name__ == "__main__":
    send_message('Hola, este es un mensaje de prueba para enviar por correo electrónico.')

```

3. **Desarrollo del Subscriptor (Subscriber)**:

- Creación de un script en Python (`subscriber.py`) que recibe mensajes de una cola de RabbitMQ.
- Uso de la librería `pika` para la conexión y el consumo de mensajes.
- Configuración del envío de correos electrónicos usando `smtplib`.

```python
import pika
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

def send_email(subject, body, to_email):
    # Configura los detalles del correo electrónico
    from_email = 'holaquehace1209@gmail.com'
    from_password = 'vsss twnn gvbf xliy'

    # Crear el mensaje
    msg = MIMEMultipart()
    msg['From'] = from_email
    msg['To'] = to_email
    msg['Subject'] = subject

    msg.attach(MIMEText(body, 'plain'))

    # Enviar el correo
    try:
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(from_email, from_password)
        text = msg.as_string()
        server.sendmail(from_email, to_email, text)
        server.quit()
        print(" [x] Email sent successfully")
```

Ejecución y Pruebas:

- Ejecutamos el productor para enviar un mensaje (python producer.py).
- Ejecutamos el subscriptor para recibir el mensaje y enviarlo por correo electrónico(python subscriber.py).

```
C:\Users\holaq\Documents>python producer.py
 [x] Sent 'Hola, este es un mensaje de prueba para enviar por correo electrónico.'

C:\Users\holaq\Documents>python subscriber.py
 [*] Waiting for messages. To exit press CTRL+C
 [x] Received Hola, este es un mensaje de prueba para enviar por correo electrónico.
 [x] Email sent successfully
```

Hola, este es un mensaje de prueba para enviar por correo electrónico.