

Software Engineer Test

Please implement the following backend system on your own laptop using any of the programming language(s) and open-source frameworks, libraries that you are familiar with within 3 hours. You can access any online resource during the test.

The system demonstrates a simple example of micro-services architecture using a messaging middleware to decouple the system components. The system processes the image payload in the query and returns the result back to the user.

It includes the following 3 components:

- C1:

A web application that offers a REST API that simply dispatch the request to backend processing service C3 via a message broker C2. Users can use the following curl command to query this API:

```
>>curl -F "image=@./test_image.jpg" localhost:8080/query
```

when the request is processed successfully, the following json result will be shown, assuming the test image is 82kb:

```
>>{"status": "SUCCESS", "size": "82kb" }
```

You can use any Web Framework to implement this service, for example, Spring, Laravel, Flask, Django, Ruby on Rails.

- C2:

A message broker receives request from C1 and either publish the message to C3 or wait for C3 to retrieve the message from it. Similarly, C3 also use C2 to send the response back to C1.

You don't need to implement this service as there are a few well established open source implementations. To name a few, Redis, RabbitMQ, Kafka.

- C3:

A service that either listens or polls C2 and processes the message, sends the result back to C2. In this test, you don't need to process the image but just **return the image size in KByte**.

You can use any image for the test, and you are not required to handle the failure cases. When you finish the test, you will be required to demonstrate your implementation and give explanations.

Added Bonus:

If your system can scale up by adding new instances of C3 service.

If you can dockerize and deploy your applications with docker-compose.

If you can mirror the image horizontally and return the new image to C1 which then saves the mirrored image on the disk.