

# 面試作業

以下為作業內容：

Implement a simple distributed message queue system using Go and either RabbitMQ or Apache Kafka. The system should have the following components:

1. **Publisher:** A Go application that can publish messages to the message queue.
2. **Consumer:** A Go application that can consume messages from the message queue and can schedule and start Kubernetes job tasks based on messages received from the message queue.
3. **Message Queue:** Either RabbitMQ or Apache Kafka as the message queue system.

The system should support the following features:

- Publish messages with a specific routing key or topic.
- Consume messages from a specific queue or topic.
- Handle message acknowledgments and redelivery in case of consumer failures.
- Implement basic error handling and logging.
- Schedule and start Kubernetes job tasks based on messages received from the message queue.

Additionally, the system should include:

- A README file with instructions on how to set up and run the applications.
- Unit tests for the publisher and consumer applications.
- A Dockerfile, Docker Compose file or any other environment files for easy deployment and testing.

Please upload your solution to a public GitHub repository when finished.

Note: The assignment can be adapted to your own ideas, please feel free to contact us to discuss further. Given the popularity of AI these days, using generative AI is also a way to go. Additionally, you may need to set up a local Kubernetes cluster (e.g. using Minikube or kind) or have access to a cloud-based Kubernetes environment for testing and development purposes.