RabbitMQ example

```
git clone https://github.com/knehez/isi.git
cd isi/rabbitmq-python
docker-compose up -d rabbitmq
```

This is a docker-compose.yml file that defines a single service called rabbitmq using the official rabbitmq Docker image with management plugin.

```yaml
version: '3.7'
services:
  rabbitmq:
    image: 'rabbitmq:3-management'
    ports:
        - 5672:5672
        - 15672:15672
```

consumer.py

```python
import pika
import sys
import os

connection =
pika.BlockingConnection(pika.ConnectionParameters(host='rabbitmq'))
channel = connection.channel()

channel.queue_declare(queue='hello')


def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)


channel.basic_consume(
    queue='hello', on_message_callback=callback, auto_ack=True)

print(' [*] Waiting for messages. ')

while (True):
    channel.start_consuming()
```

producer.py

```python
import pika

connection =
pika.BlockingConnection(pika.ConnectionParameters(host='rabbitmq'))
channel = connection.channel()
```

```python
channel.queue_declare(queue='hello')

channel.basic_publish(exchange='', routing_key='hello', body='Hello World!')
print(" [x] Sent 'Hello World!'")
connection.close()
```

Task 1: open port: 15672 → **RabbitMQ** management extension pw: guest:guest

Launch producer

```
docker-compose up -d producer
```

Launch consumer

```
docker-compose up -d consumer
```

Check the **hello** queue form the management console.

Task 2: send a message by hand and check the console of the consumer:

```
docker-compose logs -f consumer
```