**Celery and RabbitMQ Assignment 1**

Name : **Ruturaj Sutar**

Que 1) **What is Celery?**

Answer : Celery is a task queue implementation for Python web applications**,** used to perform work asynchronously outside of the HTTP requestresponse cycle.

Que 2) **What are the components of celery?**

Answer : 1. Producer (Web Nodes) , 2. Queue (Broker) , 3. Consumer (Worker)

1. Producer (Web Nodes) :

Producers are commonly the 'web nodes' or whatever system that is placing jobs.

1. Queue (Broker)

The concept of a broker is very simple: a queue.

1. Consumer (Worker)

Workers are the receivers or consumers.

Que 3) **What is a Broker?**

Answer : The concept of a broker is very simple: a queue. But what are the available ways to implement a queue in a computer system? One of the simplest would be to use a text file. Text files can hold a sequence of job descriptions to be executed, therefore, we could use them as the broker of our system.

Que 4) **Why should we use Celery instead of RQ?**

Answer : Celery is the clear winner in celery and RQ, RQ only supports Redis. This means less documentation on "what is a broker", but also means you cannot switch brokers in the future if Redis no longer works for you.

Que 5) **What is a Celery worker?**

Answer : Workers are the consumers. Celery workers are extremely efficient and customizable. Workers can set time-out for tasks (both before and during run-time), set concurrency levels, number of processes being run, and can even be set to autoscale.

Que 6) **What is RabbitMQ?**

Answer : RabbitMQ is a broker. RabbitMQ is lightweight and easy to deploy on premises and in the cloud. It supports multiple messaging protocols. RabbitMQ can be deployed in distributed and federated configurations to meet high-scale, high-availability requirements.

Que 7) **What is an exchange in RabbitMQ?**

Answer : Exchanges are AMQP 0-9-1 entities where messages are sent. Exchanges take a message and route it into zero or more queues. The routing algorithm used depends on the exchange type and rules called bindings. AMQP 0-9-1 brokers provide four exchange types: 1. Direct exchange, 2. Fanout exchange , 3. Topic exchange , 4. Headers exchange

Que 8) **What is the routing key in RabbitMQ?**

Answer : A binding is a relationship between an exchange and a queue. This can be simply read as: the queue is interested in messages from this exchange. Bindings can take an extra routing key parameter. To avoid the confusion with a basic publish parameter we're going to call it a binding key. The meaning of a binding key depends on the exchange type.

Que 9) **What are the types of exchanges available in RabbitMQ?**

Answer : 1. Direct exchange

A direct exchange delivers messages to queues based on the message routing key. A direct exchange is ideal for the unicast routing of messages (although they can be used for multicast routing as well).

2. Fanout exchange

A fanout exchange routes messages to all of the queues that are bound to it and the routing key is ignored.

3. Topic exchange

Topic exchanges route messages to one or many queues based on matching between a message routing key and the pattern that was used to bind a queue to an exchange.

4. Headers exchange

A headers exchange is designed for routing on multiple attributes that are more easily expressed as message headers than a routing key. Headers exchanges ignore the routing key attribute.

Que 10) **Design a diagram of Producer and Consumer application with Rabbitmq as Message Broker.**

Answer :

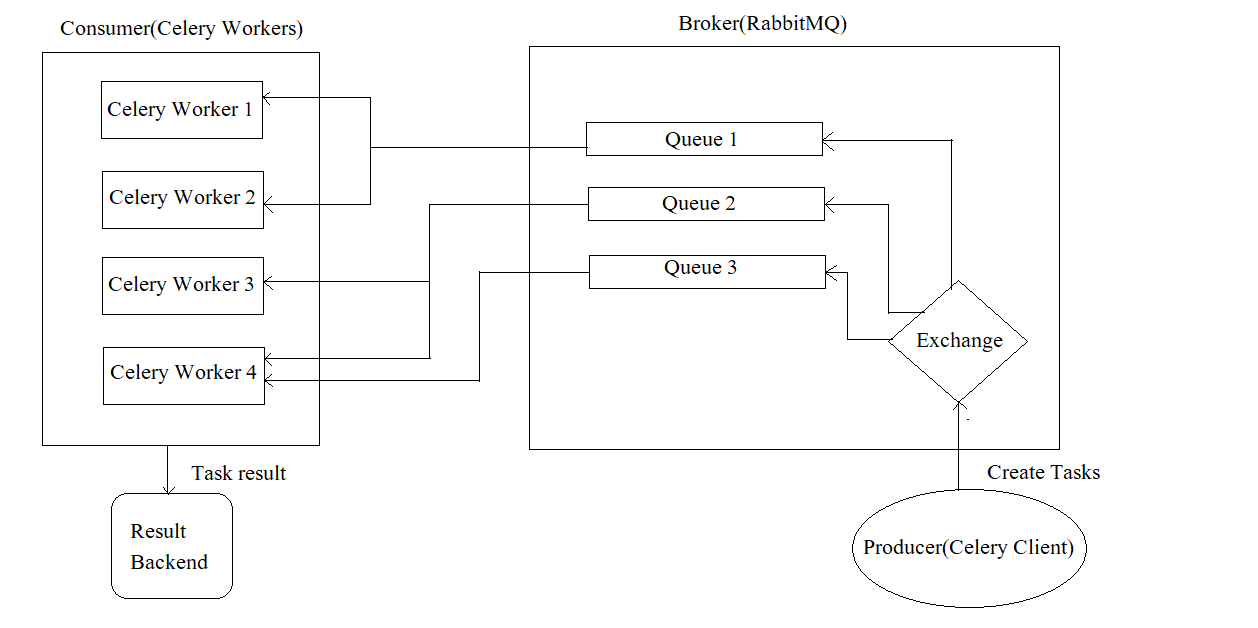


Fig . Diagram of Producer and Consumer application with Rabbitmq as Message Broker