

Jan Ilczuk

296569

rabbitmq

24 marca 2020

zad 1a

Potwierdzenie	Restart po przetworzeniu wiadomości	Restart w trakcie przetwarzania wiadomości
Po otrzymaniu wiadomości	TAK	NIE
Po przetworzeniu wiadomości	TAK	TAK

Który sposób potwierdzeń zapewnia większą niezawodność?

Po przetworzeniu wiadomości, bo wiadomości się nie gubią jak consumer crashnie.

Co się stanie, jeśli nie będziemy potwierdzać wiadomości ani po otrzymaniu, ani po przetworzeniu?

Rabbitmq zrobi requeue wiadomości i ponownie ją wyśle

zad 1b

przez QoS

<pre>Z1 CONSUMER Waiting for messages... Received: 1 Received: 1 Received: 1 Received: 1 Received: 1 </pre>	<pre>Z1 CONSUMER Waiting for messages... Received: 5 Received: 5 Received: 5 Received: 5 Received: 5</pre>
--	--

po QoS

<pre>Z1 CONSUMER Waiting for messages... Received: 1 Received: 1 Received: 5 Received: 1 Received: 5</pre>	<pre>Z1 CONSUMER Waiting for messages... Received: 5 Received: 1 Received: 5 Received: 1 Received: 5</pre>
--	--

zad 2b

Direct:

Kluczami było “ciasto” i “kanapka”. Consumer 1 miał oba te klucze w queue, a 2 Consumer miał tylko “ciasto”. Producer wysłał na początku wiadomość z kluczem “ciasto” i oba Consumerzy dostali tą wiadomość. Później wysłał z kluczem “kanapka” i tylko 1 Consumer to dostał, a później wysłał z kluczem który żaden Consumer nie miał i nikt nie dostał

<pre>Z2 PRODUCER Enter key: ciasto Enter message: hej Sent ciasto:hej Enter key: kanapka Enter message: boczek Sent kanapka:boczek Enter key: zly Enter message: nieistotne Sent zly:nieistotne Enter key:</pre>	<pre>Z2 CONSUMER key: ciasto key: kanapka created queue: amq.gen-e_OkkLjq4sj7CnE98ZIRoQ Waiting for messages... Received: hej Received: boczek </pre>
	<pre>Z2 CONSUMER key: ciasto created queue: amq.gen-59KrYn1TTpnHM9H0gPey8A Waiting for messages... Received: hej</pre>

Topic:

Jest 3 consumerów. Jeden z kluczem “#”, drugi z kluczem “blue.*”, trzeci z “blue”. Consumer z kluczem “#” działa jak Fanout. Producer wysyła na początku klucz “blue”. 1 i 3 dostają wiadomość. Producer wysyła wiadomość z kluczem “blue.green”. Wszyscy dostają wiadomość. Producer wysyła z kluczem “red”. 1 dostaje wiadomość.

```
Z2 PRODUCER
Enter key:
blue
Enter message:
is nice
Sent blue:is nice
Enter key:
blue.green
Enter message:
is ok
Sent blue.green:is ok
Enter key:
red
Enter message:
is bad
Sent red:is bad
```

```
Z2 CONSUMER
key: #
created queue: amq.gen-6FrhMbKvW_aupBlIX-ZJbw
Waiting for messages...
Received: is nice
Received: is ok
Received: is bad
|
```

```
Z2 CONSUMER
key: blue.*
created queue: amq.gen-mjufUwt0ID2I8rt6edqT4g
Waiting for messages...
Received: is ok
```

```
Z2 CONSUMER
key: blue.#
created queue: amq.gen-k09QBjMEqhnltrwj8iQImg
Waiting for messages...
Received: is nice
Received: is ok
|
```