

Работа консьюмера и гарантии доставки.

Мы ранее рассмотрели, как сделать надежной хранение сообщений в очереди. Нужно включить режим записи на диск и попросить продюсера отправлять "надежные" сообщения, с `delivery_mode = 2`. Вы потеряете в производительности, но приобретете в отказоустойчивости.

Давайте разберем теперь реализацию гарантий доставок. Как вы знаете, RabbitMQ работает по модели `push`. Это значит, что он самостоятельно отправляет сообщения консьюмеру. Отправка со стороны брокера происходит сразу же, как только появилось новое сообщение. Давайте рассмотрим, какие гарантии можно настроить, чтобы избежать потери данных, если консьюмер "упал" или перегружен и не может получить/обработать сообщение.

- Для регулировки нагрузки на консьюмер необходимо использовать настройку `Prefetch count` на стороне консьюмера

`Prefetch count` (количество предварительных сообщений): эта настройка определяет, сколько сообщений брокер будет предварительно отправлять консьюмеру, прежде чем будет ожидать подтверждения обработки. При установке значения в 0 настройка отключается, и брокер будет немедленно отправлять все доступные сообщения консьюмеру.

Когда вы устанавливаете значение `Prefetch count = 1`, фактически, брокер будет отправлять одно сообщение консьюмеру, и только после того как консьюмер подтвердит обработку этого сообщения, брокер отправит следующее сообщение. Таким образом, величина `Prefetch count` в данном случае ограничивает количество неподтвержденных (необработанных) сообщений, которые могут находиться в пути к консьюмеру. Проще говоря, настройка `Prefetch count` регулирует, сколько сообщений может быть активно в обработке у консьюмера, прежде чем ему потребуется подтвердить обработку хотя бы одного из них. Регулируя эту настройку вы можете регулировать нагрузку на консьюмер.

- `Ack / Nack (at least once`, как минимум один раз)

`Acknowledge`: подтверждение "да, я успешно обработал это сообщение". Когда консьюмер успешно обрабатывает сообщение, он отправляет подтверждение (`ack`) брокеру. Это гарантирует, что сообщение было надежно обработано, и оно может быть удалено из очереди.

`Negative Acknowledge`: подтверждение "нет, обработать сообщение не получилось". Когда обработка сообщения может быть неудачной, консьюмер может отправить отрицательное подтверждение (`nack`) для сообщения, указав, что оно не может быть успешно обработано. Это позволяет вернуть сообщение обратно в очередь. Кроме того, механизм "отрицательного подтверждения" также автоматически срабатывает при разрушении канала связи с консьюмером, что обеспечивает безопасный возврат сообщений в очередь в случае, если консьюмер выходит из строя из-за сетевых проблем.

Напомним, при использовании данного механизма нужно предусмотреть обработку дублей сообщений, если это необходимо (гарантия exactly once). Ранее мы разбирали, что они могут появляться из-за сетевых проблем. Например, консьюмер может успешно получить и обработать сообщение, но его ответ может не дойти до брокера.

И также затронем AutoAck. Это автоматическое подтверждение брокером всех отправленных сообщений. Т.е. брокер не ждет от консьюмера ответов и продолжает отправлять сообщения, как только они появляются в очереди. Механизм позволяет реализовать гарантию at most once (отправил и забыл). Работает быстрее других механизмов, но надежность вашей системы снижается.

Для взаимодействия продюсер-брокер предусмотрен тот же механизм Ack / Nack. Брокер сообщений отправляет положительное или отрицательное подтверждение для продюсера или вы просто отключаете эти подтверждения, повышая скорость работы системы. Есть еще дополнительные настройки.

Помните, мы затронули параметр confirm? Если он проставлен в сообщении, то брокер дополнительно уведомляет продюсера что сообщение успешно доставлено до консьюмера. Этот механизм позволяет продюсеру полноценно понимать, что происходит с его сообщениями. Обычно его используют для реализации паттерна "запрос-ответ", где очень важно знать о состоянии сообщения, т.к. процесс без ответа просто блокируется.

Также в качестве гарантий доставки от продюсера к брокеру используется параметр mandatory в сообщении. Если его указать true, то сообщение будет возвращено продюсеру или передано в отдельную очередь (зависит от разработки), если для него не нашлось параметров binding. Это позволяет не упускать сообщения, если роутинг сработал не так, как вы рассчитывали. Продюсеру нужно быть готовым обрабатывать такие сообщения и выполнять нужную логику на своей стороне.