#### Сравнение: RabbitMQ vs Apache Kafka vs Azure Service Bus

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | RabbitMQ | Apache Kafka | Azure Service Bus | Преимущества | Недостатки |
| Тип | Message Broker | Distributed Log | Message Broker (PaaS) | Выбор под разные сценарии | Может быть сложно выбрать правильный инструмент |
| Архитектура | Push-based | Pull-based | Push-based | Оптимизация под тип нагрузки | Разные модели требуют изучения |
| Скорость обработки | Средняя (~50K msg/sec) | Очень высокая (>1M msg/sec) | Средняя (~20K msg/sec) | Подходит под разные объемы | Kafka сложен для небольших нагрузок |
| Хранение сообщений | Ограниченное, по TTL | Долговременное (на диске) | TTL или Scheduled | Kafka для аналитики и аудита | RabbitMQ теряет сообщения по TTL |
| Поддержка очередей | Да | Топики и партиции | Да | Гибкость топологии | Kafka требует иной подход к очередям |
| Гарантии доставки | At least / at most once | At least / exactly once (сложно) | At least / exactly once | Выбор уровня гарантии | Exactly once сложен и ресурсоемок |
| Поддержка транзакций | Ограниченная | Да (с версия 0.11) | Да | Возможность атомарных операций | Может влиять на производительность |
| Управление и UI | Удобная панель | CLI и UI-инструменты | Портал Azure | Разные варианты администрирования | У Azure зависимость от облака |
| Интеграция | Хорошая с множеством систем | Лучше всего с Big Data/Streaming | Хорошая в Azure-экосистеме | Выбор по экосистеме | Ограничения вне своей экосистемы |
| Использование | Сценарии очередей, RPC | Big Data, потоковая аналитика | Enterprise messaging, cloud apps | Подход к разным задачам | Оверинжиниринг при неправильном выборе |

**Почему для мини-проекта стоит выбрать RabbitMQ?**

*Kafka требует настройки кластера и Zookeeper, а RabbitMQ запускается одной командой в Docker.*

Azure Service Bus платный и избыточен для локального тестирования.

*RabbitMQ поддерживает несколько типов обмена сообщениями (Direct, Fanout, Topic), что удобно для микросервисов.*

Kafka работает только с топиками, что менее гибко для небольших проектов.

*В RabbitMQ легко настроить обработку ошибок через DLQ, что критично для EmailService.*

В Kafka для этого нужно вручную настраивать дополнительные топики.

*RabbitMQ потребляет меньше ресурсов, чем Kafka, и не требует долгосрочного хранения сообщений.*

В мини-проекте высокая пропускная способность Kafka не нужна.

*RabbitMQ не требует облачной подписки (в отличие от Azure Service Bus).*