

Микросервис на Go
шина данных,
контейнеры,
мониторинг

Игорь Мосягин



Шина данных на сегодня

<https://nsq.io> от bitly

Написан на go: быстрая и лёгкая

Компоненты:

- nsqd: демон, отвечающий за сообщения

- nsqllookupd: координатор (местный zookeeper)

- nsqadmin: дашборд и немного админка

Гарантии nsq

сообщения могут быть доставлены больше 1 раза

сообщения не отсортированы

нет встроенной репликации

eventual consistency

больше [тут](#) (от самих bitly) и [тут](#) (в сравнении с кафкой):

код клиента на ВТОРОМ питоне: <https://gist.github.com/3925092>

микросервис на Go

быстро компилируется

network objects are first-class citizens

удобно асинхронится

болезненно скудный синтаксис

невозможно неприятный маскот

автор не смог написать ОС, написал ЯП

поддерживается гуглом, непонятно что будет потом

ресурсы про go

<https://golang.org>

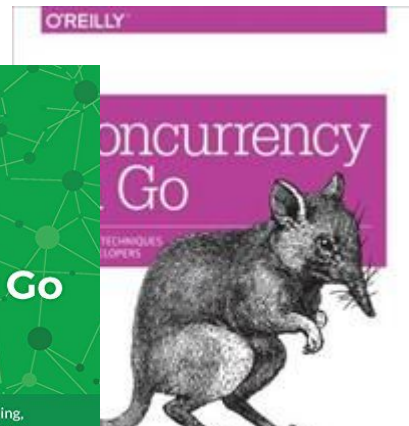
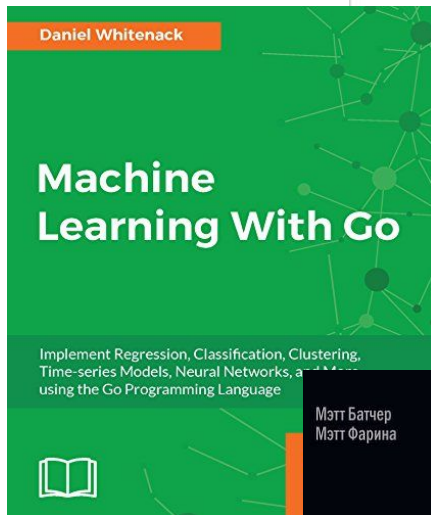
<https://learnxinyminutes.com/docs/go/>

отличный курс от mail.ru на курсе

алсо, книжки

и пет-проджекты: go идеален для raspberry-pi

(check out <https://gobot.io/>)



про микросервисы

<https://samnewman.io/>

<https://martinfowler.com/microservices/>

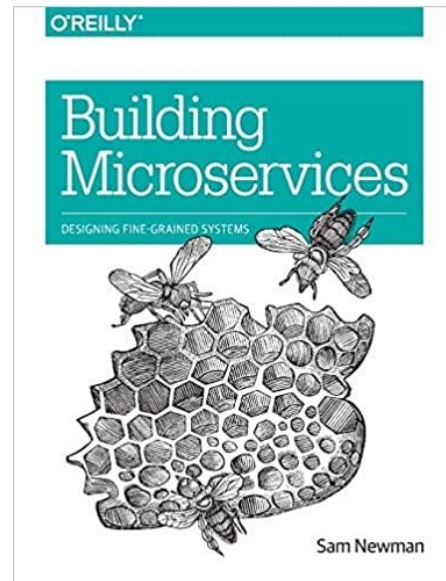
делает что-то одно,, но хорошо

новый сервис с нуля за один спринт

легко масштабировать

нужна хорошая сеть и контракты

команде хватит две пиццы



алсо, <https://podlodka.io/212>

Какие ещё есть очереди

- Kafka
 - “а у всех всё равно кафка в проде”
 - At least once, можно настроить exactly once
- NATS
 - сравнение с другими платформами: <https://docs.nats.io/compare-nats>
 - At most once, at least once, и с доп. настройками Exactly once
 - переписан с Ruby на go
- Apache Rocketmq
 - сравнение с другими платформами: <https://rocketmq.apache.org/docs/motivation/>
 - есть транзакции и упор на low latency, high reliability

Немного о безопасности

- User Input как таковой в шине скорее отсутствует
- Большинство имплементаций не умеют делать сложный RBAC
- OWASP Top-10 в каком-то смысле перекладывается на то, как можно писать микросервисы
<https://owasp.org/www-project-top-ten/>
- У вас by design почти нигде в очередях нет rate limiting
- Доступ обычно регулируется сегментацией или user+password

demo

<https://github.com/cra/npl-dataeng-demo>

“один день из жизни разработки микросервисов”

Ближе к концу демо будет небольшая интерактивная часть, вам пригодится возможность открывать ссылки в интернете с разных браузеров

demo newgame+ mode

1. Сделать больше настроек на шаге 4*
 - a. переменные окружения (<https://github.com/kelseyhightower/envconfig>)
 - b. ENV и ARG в Dockerfile
 - c. настройки, которые можно захотеть тюнить тоже лучше вынести в env
2. Сейчас сообщение не сохраняется нигде (кроме прометея)
 - a. сохраните его в что-то что вы ещё не пробовали Boltddb/Redis/Mongodb/Aerospike
 - b. добавьте выгрузку агрегатов из этой конечной точки в psql по расписанию
 - c. больше событий и их параметров на стороне отправки в server.js
3. Выбросить X и заменить на Y, сравнить скорость
 - a. заменить go на async python, сравнить скорость. Никогда больше не писать на go
 - b. поменять nsq на nats/kafka/rabbitmq/rocketmq
 - c. где-то воткнуть no-code решение типа airtable или integromat

get in touch!

околоайтишный трындёж: <https://newpodcast2.live>

соцсети: [@shrimpsizemoose](#)

больше ссылок <https://mosyag.in/>