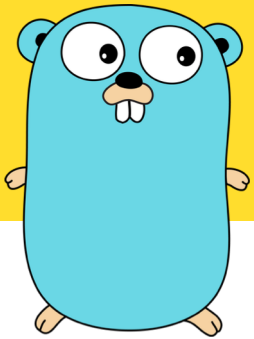


Production-ready microservices

Tinkoff.ru

План занятия

1. Технические ручки
2. Настройка параметров окружения
3. Мониторинг
4. Диагностика
5. Доступность

Технические ручки

Да зачем на них вообще тратить время?

Healthcheck probes

GET /liviness

GET /readiness

HTTP 200 - OK

HTTP 500 - Error

Startup status

/statusz

```
{
  ”status”:”OK”,
  “version”:”v0.1-13-g148112e1”,
  “config”: …
}
```

```
{
  ”status”:”Error”,
  ”message”: “can’t init database connection: i/o timeout: dial tcp 1.1.1.1”,
  “version”:”v0.1-13-g148112e1”,
  “config”: {
    …
  }
}
```

Z-pages

Z-Pages

- Allows processes report their own dashboards.
- Z-Pages have no sampling.

TraceZ Summary

Span Name	Running	Latency Samples									Error Samples
		[>0us]	[>10us]	[>100us]	[>1ms]	[>10ms]	[>100ms]	[>1s]	[>10s]	[>100s]	
HttpServer/traceconfigz	0	0	0	0	0	0	0	0	0	0	0
HttpServer/tracez	1	0	0	0	0	1	0	0	0	0	0
Recv.helloworld.Greeter.SayHello	0	0	0	0	0	10	0	0	0	0	1

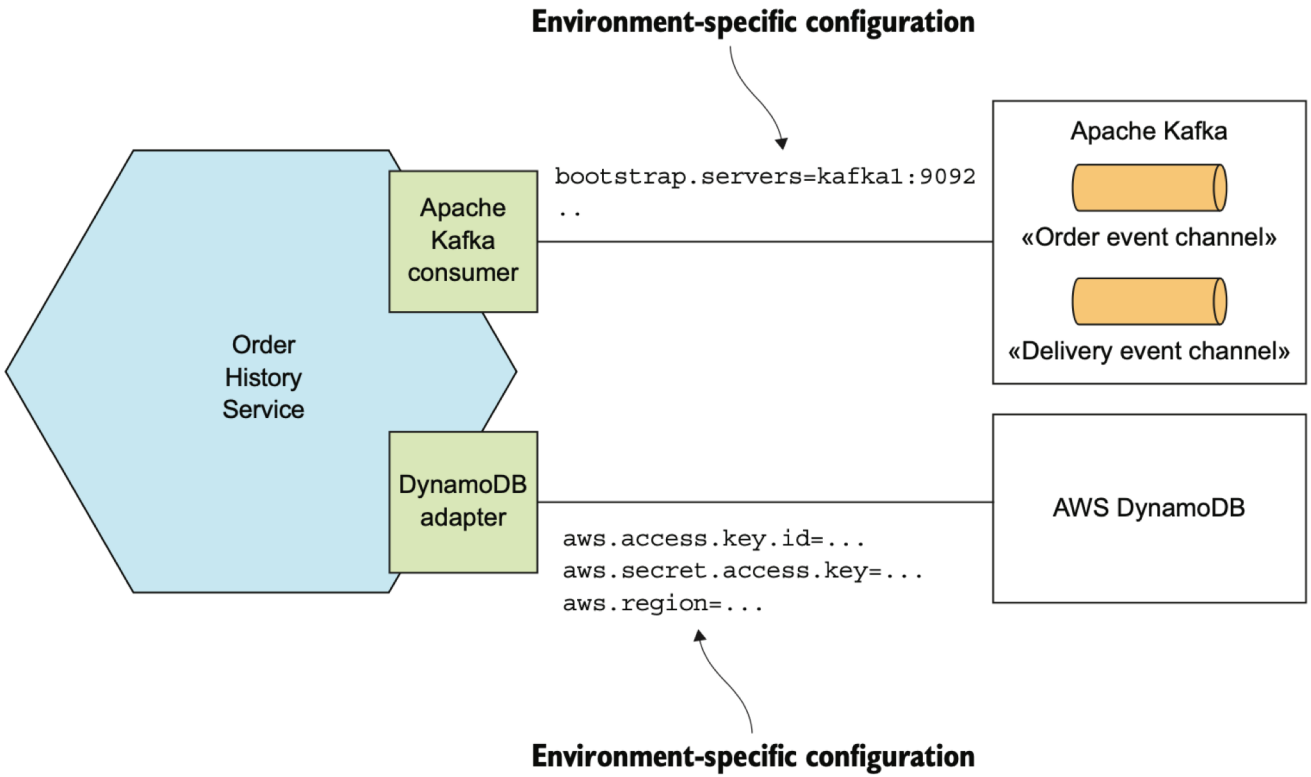
Span Name: Recv.helloworld.Greeter.SayHello

Finished Requests 10

```
When                                    Elapsed(s)
----------------------------------------
2018/02/02-12:55:03.215000      0.052822     TraceId: 038c6b3fb381ec5f3b548fe375f37ccf SpanId: dad61c20fc94e4b0 ParentSpanId: 0004e8caec703c82
          12:55:03.215126        .   126 ... Received message id=0 uncompressed_size=0 compressed_size=10832
          12:55:03.267723        . 52596 ... Sent message id=0 uncompressed_size=10838 compressed_size=10838
                                             Status{canonicalCode=OK, description=null}
                                             Attributes:{}
2018/02/02-12:55:02.177000      0.028152     TraceId: 3c55d9a2c4a0031fc6b16dbcf66d28d7 SpanId: b6534ce7c76c6795 ParentSpanId: 3f4effd04fdc2634
          12:55:02.177160        .   160 ... Received message id=0 uncompressed_size=0 compressed_size=1563
          12:55:02.205104        . 27943 ... Sent message id=0 uncompressed_size=1569 compressed_size=1569
```

Параметры окружения

Зачем параметризовать?



Что выносим в параметры?

Все, что меняется на окружениях dev/test/stage/prod:

Например:

- подключение к postgresql/clickhouse/kafka/nat/rabbitmq…
- подключение к вызываемым сервисам (ip/port/timeout)

Все, что может меняться в процессе жизни приложения:

Например:

- Кэш (время жизни, размер)
- Троттлинг (кол-во запросов в минуту, кол-во одновременных запросов

Мониторинг

Метрики

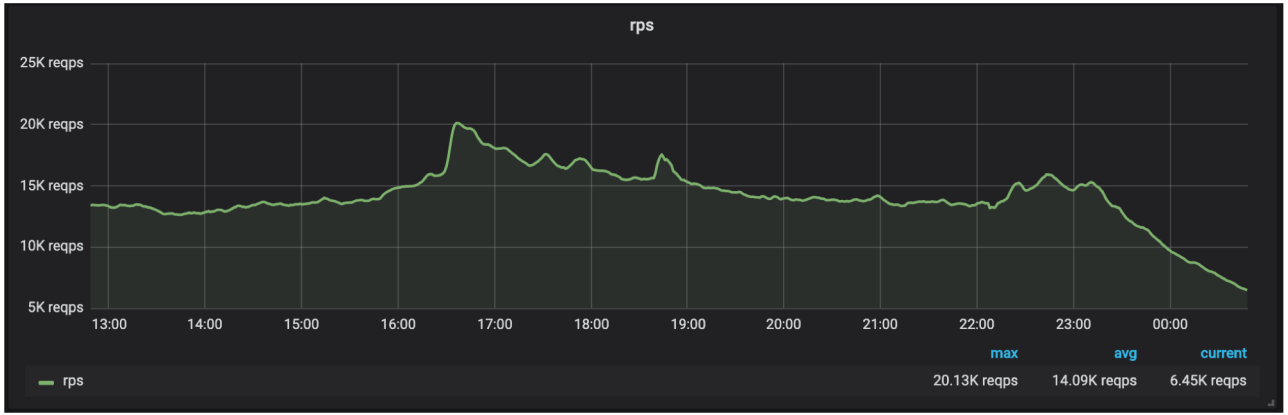
RED

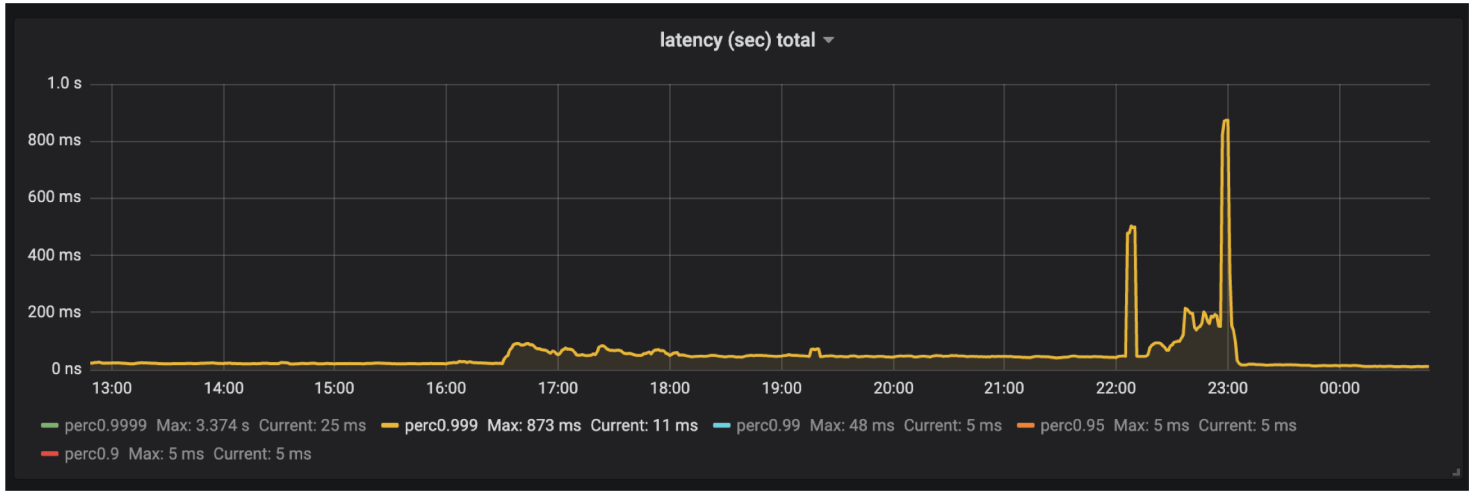
- Rate
- Errors
- Duration(Latency)

USE

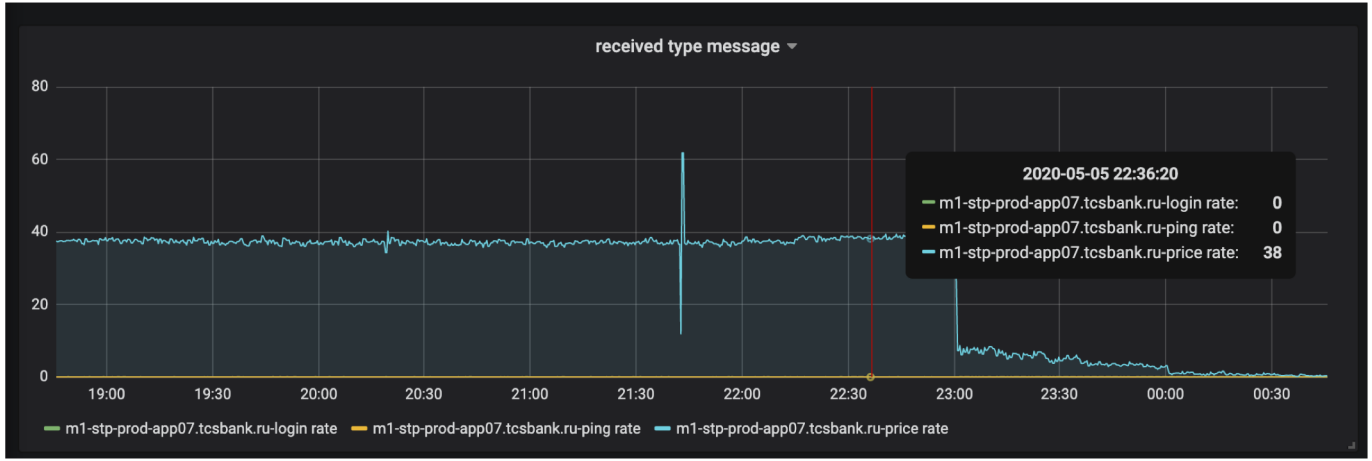
- Utilization
- Saturation
- Errors

Пример RED


rps
25K reqps
20K reqps
15K reqps
10K reqps
5K reqps
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
00:00
max
avg
current
rps
20.13K reqps
14.09K reqps
6.45K reqps


latency (sec) total
1.0 s
800 ms
600 ms
400 ms
200 ms
0 ns
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
00:00
perc0.9999 Max: 3.374 s Current: 25 ms
perc0.999 Max: 873 ms Current: 11 ms
perc0.99 Max: 48 ms Current: 5 ms
perc0.95 Max: 5 ms Current: 5 ms
perc0.9 Max: 5 ms Current: 5 ms

Пример USE


received type message
80
60
40
20
0
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30
00:00
00:30
2020-05-05 22:36:20
m1-stp-prod-app07.tcsbank.ru-login rate: 0
m1-stp-prod-app07.tcsbank.ru-ping rate: 0
m1-stp-prod-app07.tcsbank.ru-price rate: 38
m1-stp-prod-app07.tcsbank.ru-login rate
m1-stp-prod-app07.tcsbank.ru-ping rate
m1-stp-prod-app07.tcsbank.ru-price rate


errors type
0.040
0.030
0.020
0.010
0
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30
00:00
00:30
m1-stp-prod-app07.tcsbank.ru-read_WS

Prometheus + Grafana

https://github.com/prometheus/client_golang

```
// nolint: interfacer
func makeTechRoutes(logger log.Logger, moexPrivateHandler *PrivateMOEXHandler)
handler.Option {
   return func(r *handler.Handler) {
      silent := r.Group(
         handler.WithRecover(),
      )

      silent.Mount("/debug", middleware.Profiler())
      silent.Mount("/metrics", promhttp.HandlerFor(prometheus.DefaultGatherer,
promhttp.HandlerOpts{ErrorLog: logger}))
      silent.Mount("/readiness", health.Routes())

      silent.Mount("/private", moexPrivateHandler.Routes())
   }
}
```

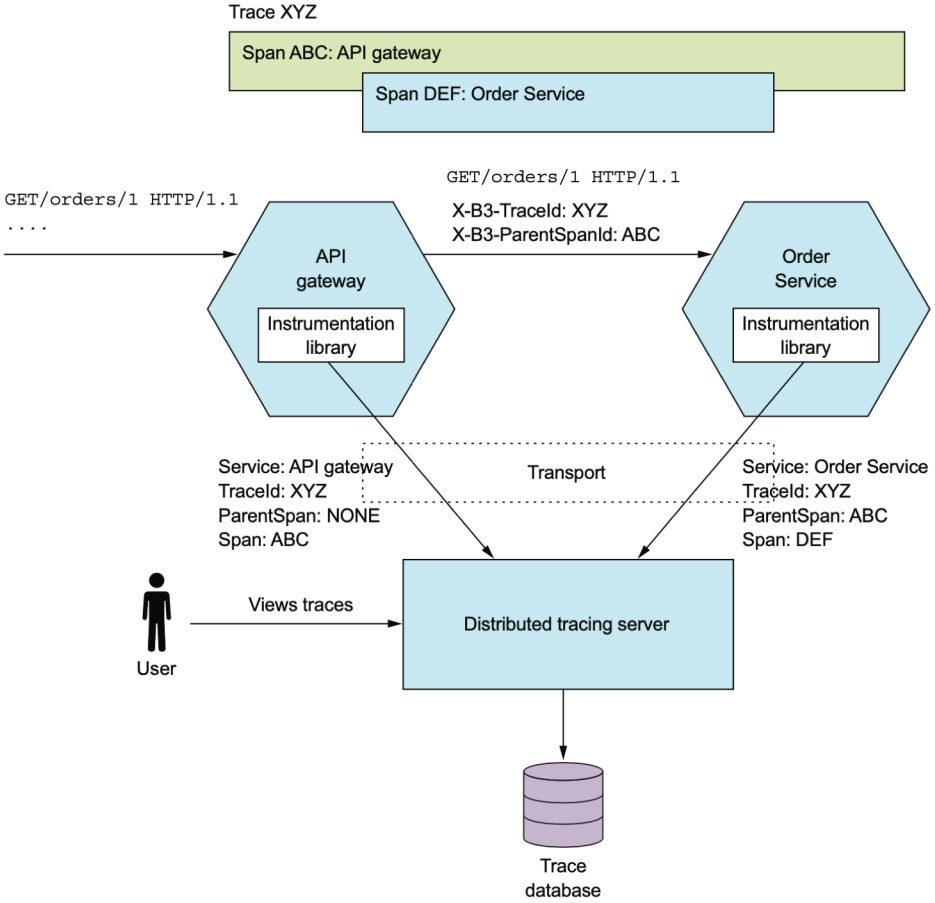
Output пример

```
# HELP fix_bytes_in total bytes received
# TYPE fix_bytes_in counter
fix_bytes_in 771
# HELP fix_bytes_out total bytes sent
# TYPE fix_bytes_out counter
fix_bytes_out 769
# HELP fix_errors total fix errors occurred
# TYPE fix_errors counter
fix_errors{error_type="internal_error"} 1
fix_errors{error_type="receive_error"} 1
# HELP fix_messages_in total fix messages received
# TYPE fix_messages_in counter
fix_messages_in{message_type="heartbeat"} 6
fix_messages_in{message_type="logon"} 1
fix_messages_in{message_type="logout"} 1
# HELP fix_messages_out total fix messages sent
# TYPE fix_messages_out counter
fix_messages_out{message_type="heartbeat"} 6
fix_messages_out{message_type="logon"} 1
fix_messages_out{message_type="logout"} 1
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
```

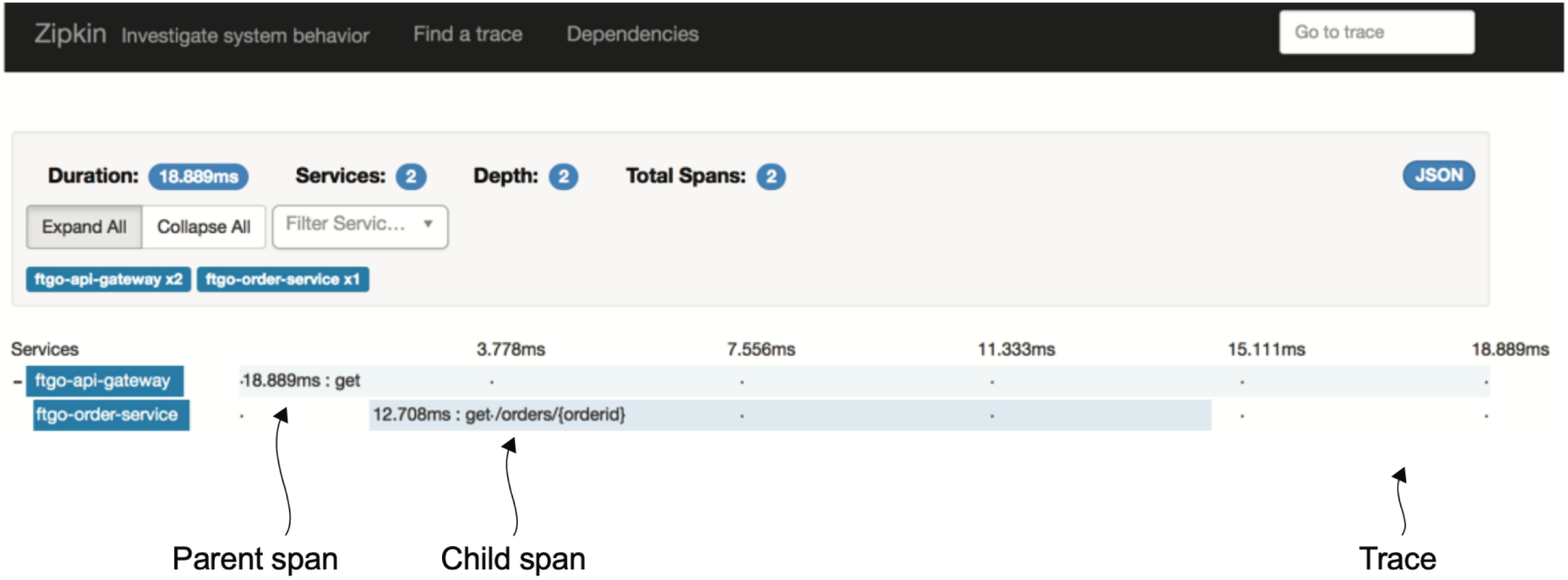
Диагностика

Distributed tracing

https://github.com/openzipkin/b3-propagation



Zipkin


Zipkin
Investigate system behavior
Find a trace
Dependencies
Go to trace
Duration: 18.889ms
Services: 2
Depth: 2
Total Spans: 2
JSON
Expand All
Collapse All
Filter Servic...
ftgo-api-gateway x2
ftgo-order-service x1
Services
3.778ms
7.556ms
11.333ms
15.111ms
18.889ms
ftgo-api-gateway
18.889ms : get
ftgo-order-service
12.708ms : get /orders/{orderid}
Parent span
Child span
Trace

Пример


32.934ms
65.868ms
98.801ms
131.735ms
164.669ms
164.669ms : get /timeline
36.362ms : auth.accesstoken
128.204ms : inbox.timeline.messages
5.162ms : cache.get
72.078ms : mysql.createsession
21.417ms : mysql.query
8.925ms : cache.put
8.838ms : user.profile
5.191ms : cache.get
8.817ms : user.profile
5.220ms : cache.get
8.809ms : user.profile
5.221ms : cache.get
8.782ms : user.profile
5.208ms : cache.get
8.791ms : user.profile
5.162ms : cache.get
8.733ms : user.profile
5.208ms : cache.get

Go library

https://github.com/openzipkin/zipkin-go

Доступность

Кэши

https://github.com/patrickmn/go-cache

Дедупликация

https://godoc.org/golang.org/x/sync/singleflight

Троттлинг

https://github.com/throttled/throttled

Переповторы

https://github.com/hashicorp/go-retryablehttp

Что-то еще?

Circuit breaking, fallback, etc.

Обратная связь

Спасибо за внимание

Tinkoff.ru