Custom AWS lambda runtimes

...and a crystal demo

Alexander Reelsen
@spinscale
alr@spinscale.de

There is no serverless architecture. It's just someone else's execution environment.

Alex R.

About me

- Working distributed at Elastic since 2013
- Elasticsearch developer
- Dislikes running own infrastructure
- Basketball & JVM fan, husband, dad

Agenda

- Custom AWS Lambda runtimes
- Introduction into crystal lang
- A crystal based runtime

AWS Lambda

FAAS

node.js

python

go

java

JVM

startup time

dependencies

JIT

2015: custom runtime?

node shim

https://aws.amazon.com/blogs/compute/running-executables-in-aws-lambda/

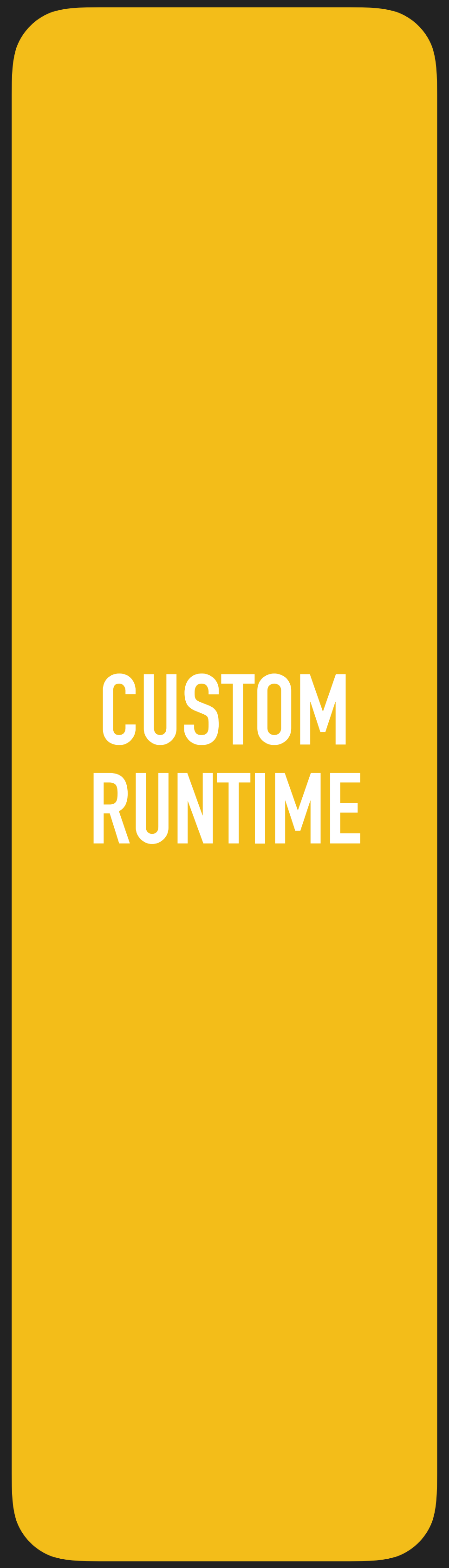
2018: custom runtime!

all the languages!

runtime flow

CUSTOM
RUNTIME

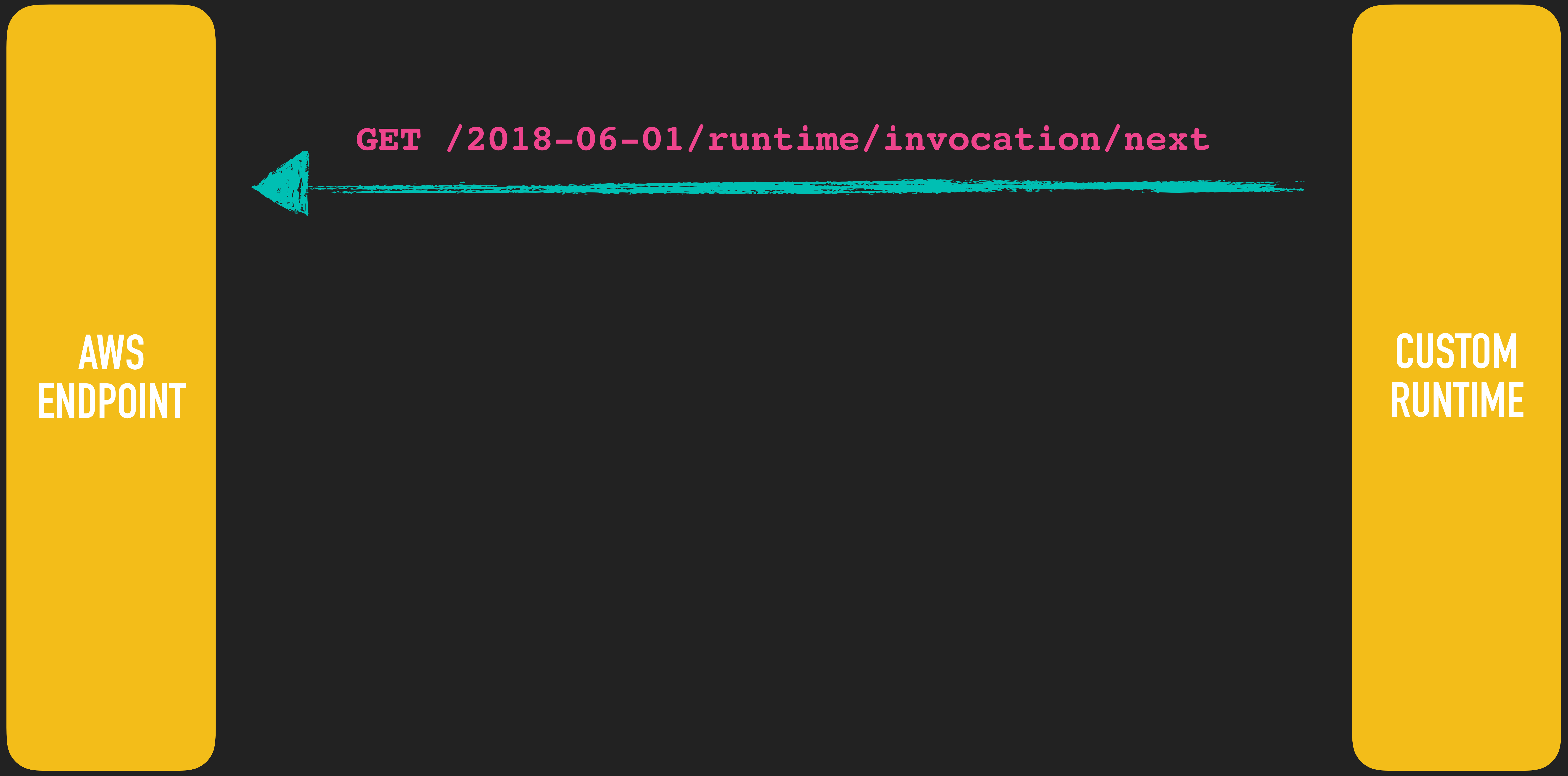
```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```

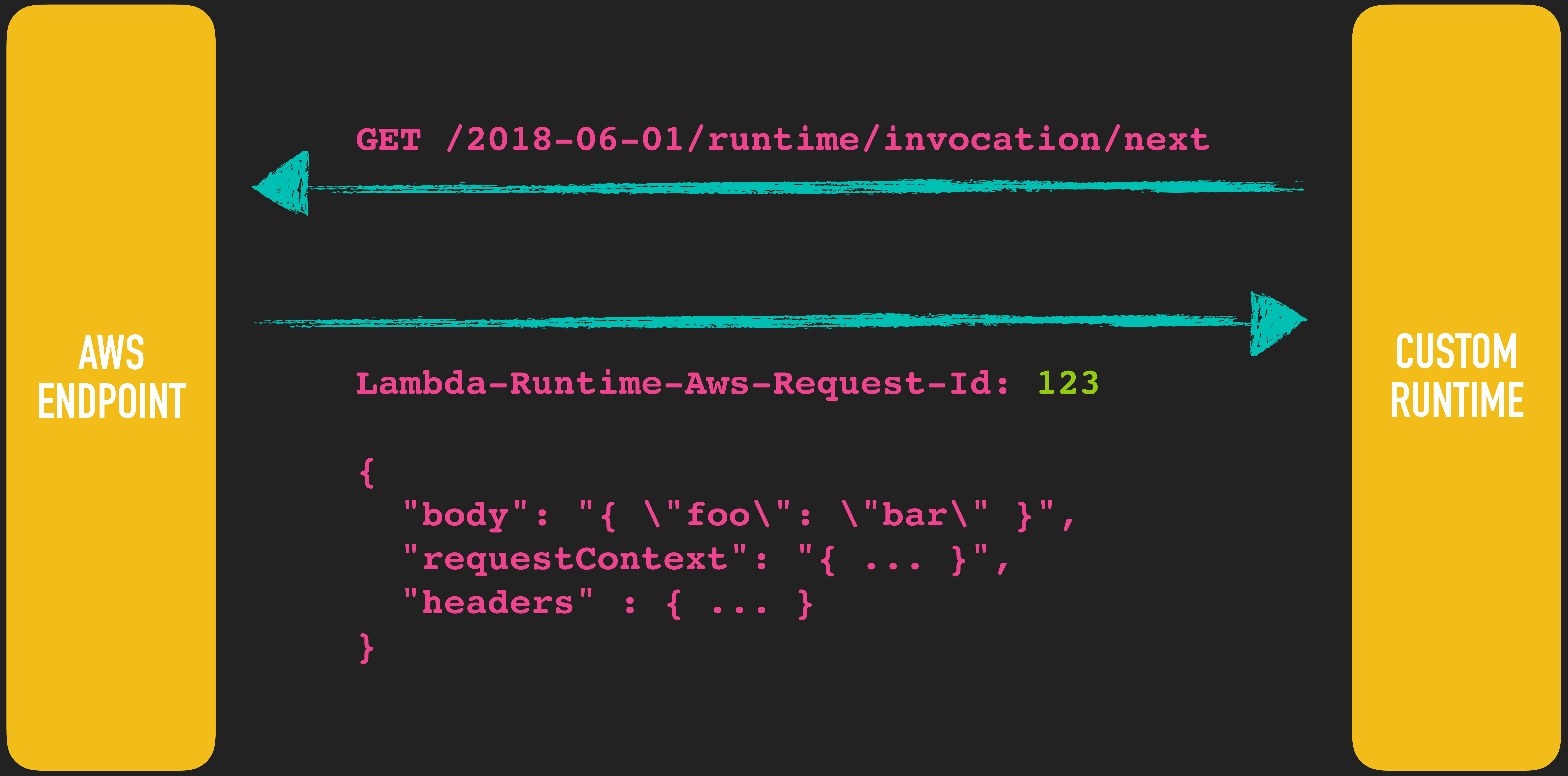
AWS ENDPOINT
CUSTOM RUNTIME

```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
GET /2018-06-01/runtime/invocation/next
```

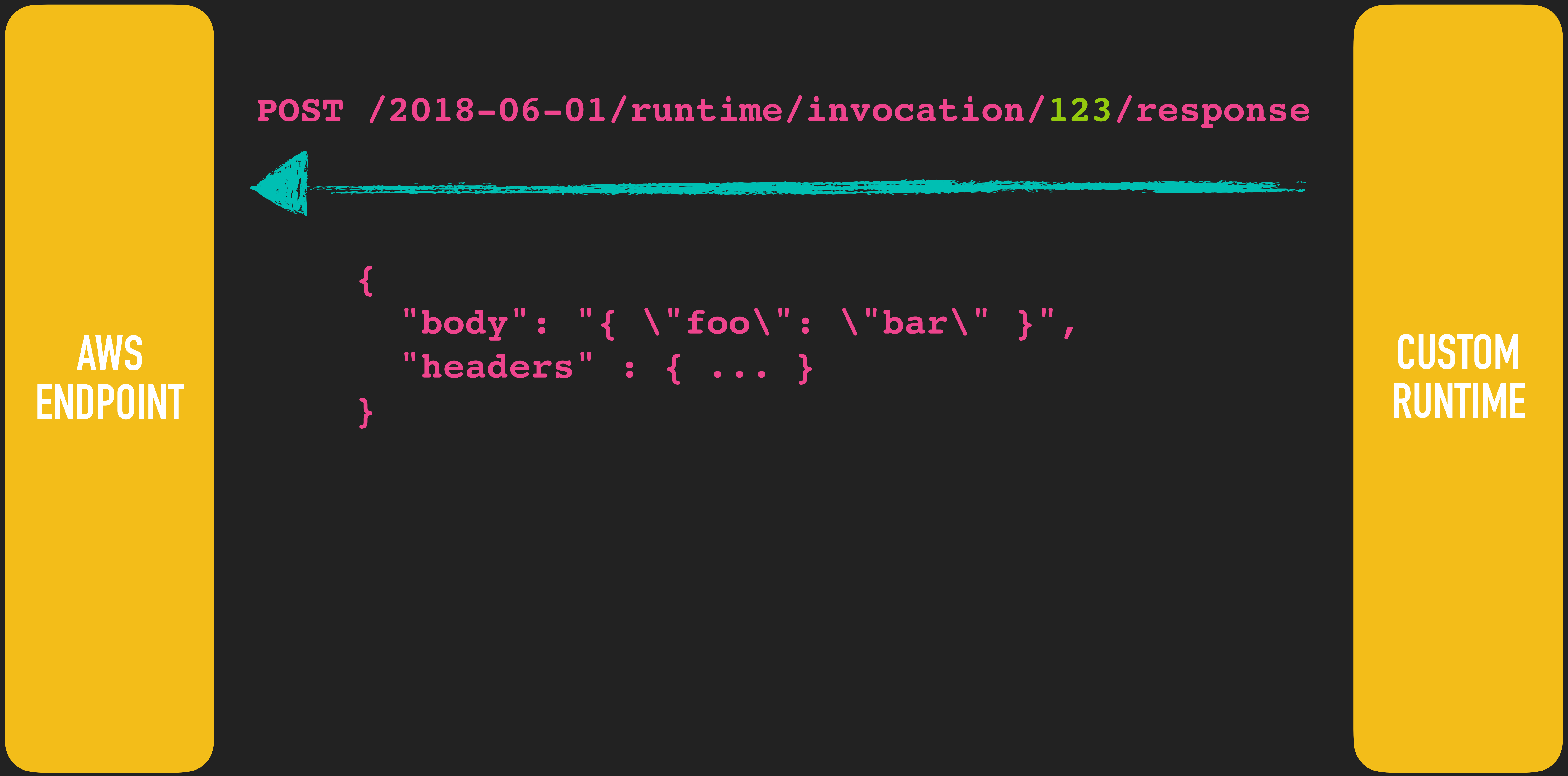
```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
GET /2018-06-01/runtime/invocation/next
```

```
Lambda-Runtime-Aws-Request-Id: 123

{
  "body": "{ \"foo\": \"bar\" }",
  "requestContext": "{ ... }",
  "headers" : { ... }
}
```

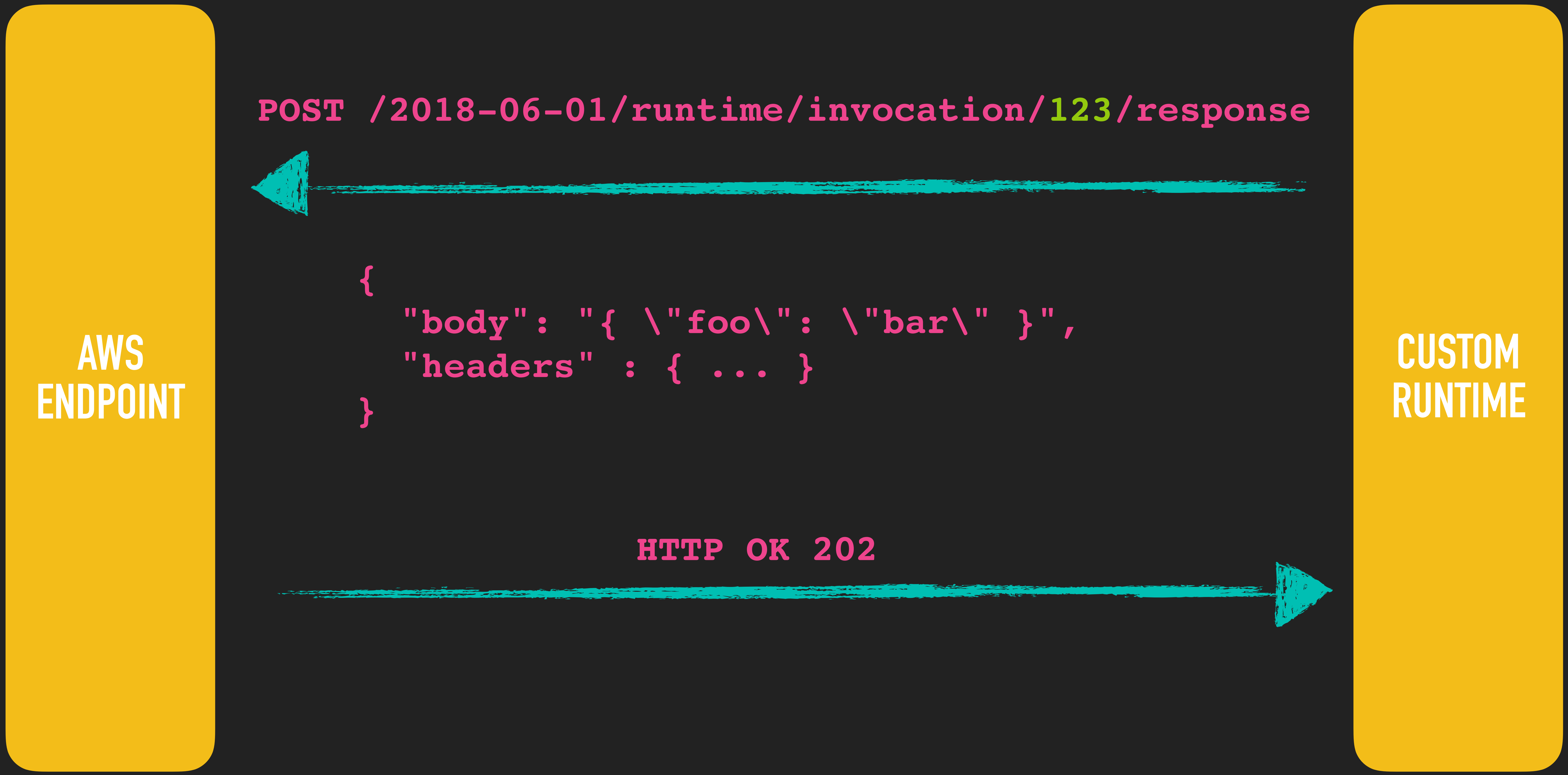
```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
POST /2018-06-01/runtime/invocation/123/response
```

```
{
  "body": "{ \"foo\": \"bar\" }",
  "headers" : { ... }
}
```

```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```

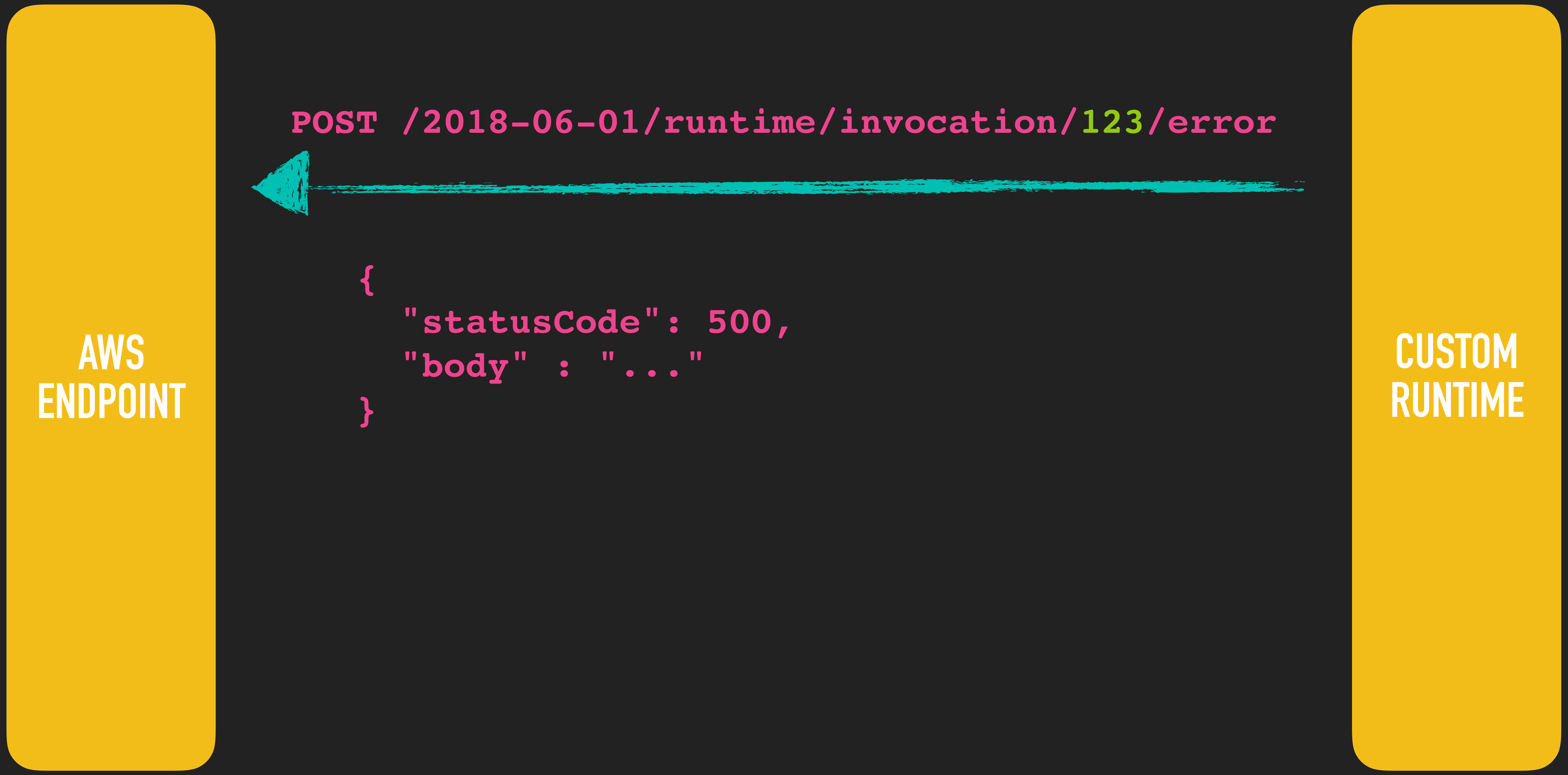


```
POST /2018-06-01/runtime/invocation/123/response
```

```
{
  "body": "{ \"foo\": \"bar\" }",
  "headers" : { ... }
}
```

```
HTTP OK 202
```

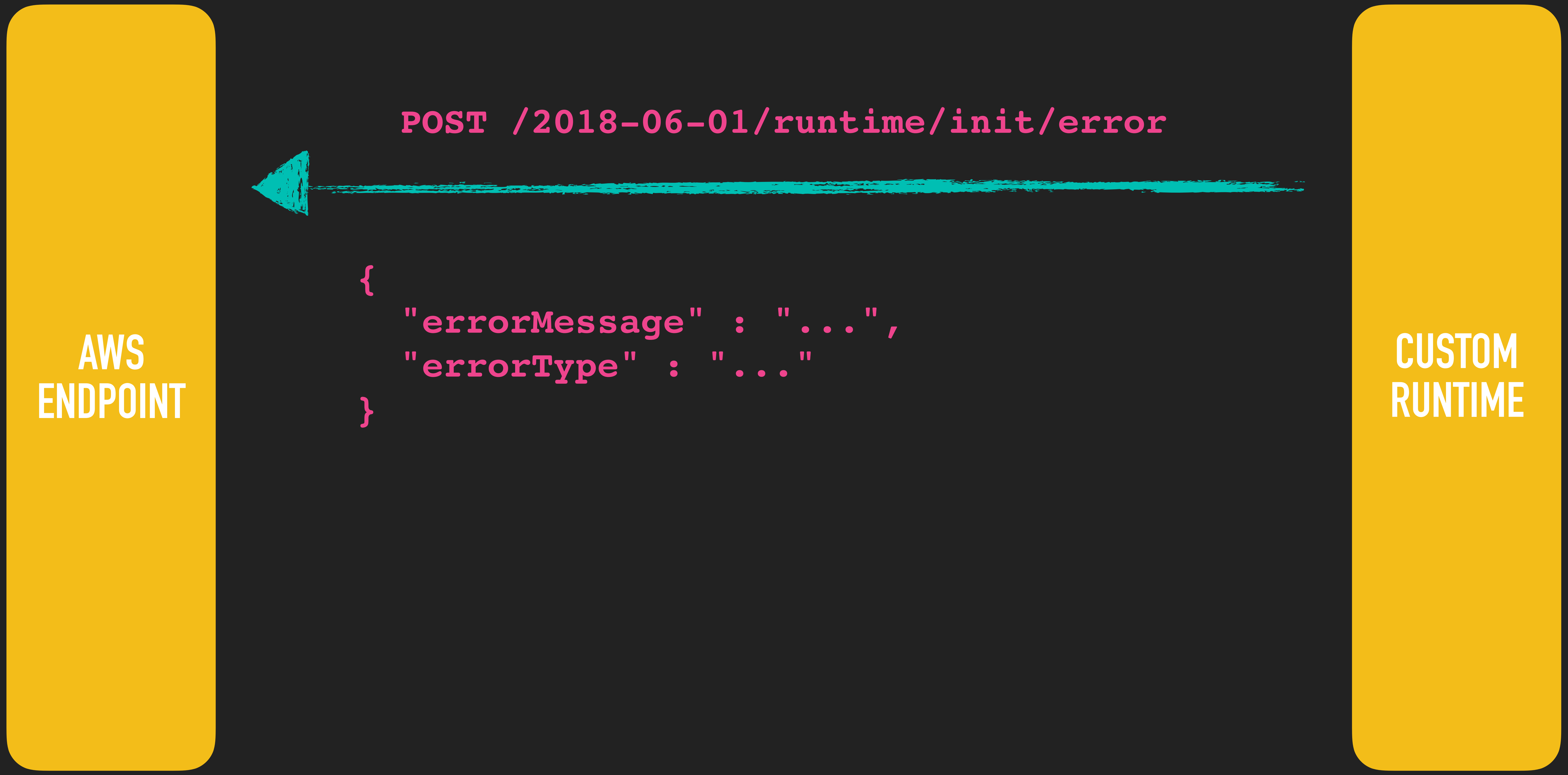
```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
POST /2018-06-01/runtime/invocation/123/error
```

```
{
  "statusCode": 500,
  "body" : "..."
}
```

```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"

/bin/bootstrap
```



```
POST /2018-06-01/runtime/init/error
```

```
{
  "errorMessage" : "...",
  "errorType" : "..."
}
```

crystal

readable

CRYSTAL

fast

CRYSTAL

statically typed

batteries included

CRYSTAL

```
/tmp  cat server.cr

File: server.cr

# A very basic HTTP server
require "http/server"

server = HTTP::Server.new do |context|
  context.response.content_type = "text/plain"
  context.response.print "Hello world, got #{context.request.path}!"
end

puts "Listening on http://127.0.0.1:8080"
server.listen(8080)
```

testing

CRYSTAL

benchmarks

CRYSTAL

metaprogramming

CRYSTAL

fibers

channels

CRYSTAL

boehm gc

c bindings

CRYSTAL

dependency mgmt

CRYSTAL

web frameworks

binaries

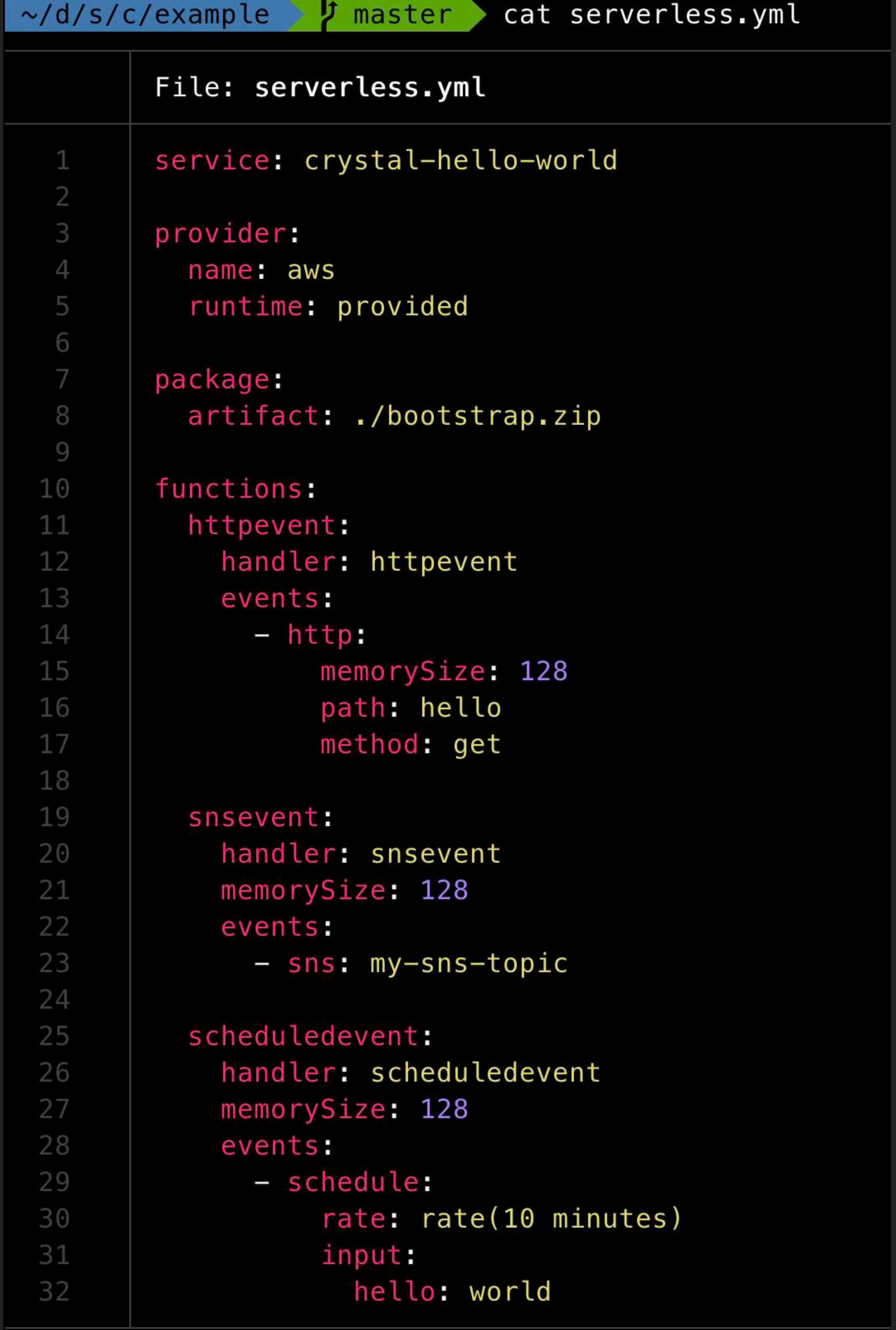
CRYSTAL

static binaries

CRYSTAL

```crystal
def _process_request(proc : Proc(JSON::Any, JSON::Any))
  client = HTTP::Client.new(host: @host, port: @port)

  begin
    response = client.get "/2018-06-01/runtime/invocation/next"
    ENV["_X_AMZN_TRACE_ID"] = response.headers["Lambda-Runtime-Trace-Id"] || ""
    aws_request_id = response.headers["Lambda-Runtime-Aws-Request-Id"]
    base_url = "/2018-06-01/runtime/invocation/#{aws_request_id}"
    input = JSON.parse response.body
    body = proc.call input

    @logger.info("preparing body #{body}")
    response = client.post("#{base_url}/response", body: body.to_json)
    @logger.debug("response invocation response #{response.status_code} #{response.body}")
  rescue ex
    body = %Q({ "statusCode": 500, "body" : "#{ex.message}" })
    response = client.post("#{base_url}/error", body: body)
    @logger.error("response error invocation response from exception #{ex.message} #{response.status_code} #{response.body}")
  ensure
    client.close
  end
end
```

```crystal
require "lambda_builder"

runtime = LambdaBuilder::Runtime.new

runtime.register_handler("httpevent") do |input|
  req = LambdaBuilder::Util::LambdaHttpRequest.new(input)
  user = req.query_params.fetch("hello", "World")
  response = LambdaBuilder::Util::LambdaHttpResponse.new(200, "Hello #{user} from Crystal")
  # not super efficient, serializing to JSON string and then parsing, simplify this
  JSON.parse response.to_json
end

runtime.register_handler("scheduledevent") do |input|
  runtime.logger.debug("Hello from scheduled event, input: #{input}")
  JSON.parse "{}"
end

runtime.register_handler("snsevent") do |input|
  runtime.logger.info("SNSEvent input: #{input}")
  JSON.parse "{}"
end

runtime.run
```



```
~/d/s/c/example  master  cat serverless.yml
```

```
File: serverless.yml
```

```yaml
service: crystal-hello-world

provider:
  name: aws
  runtime: provided

package:
  artifact: ./bootstrap.zip

functions:
  httpevent:
    handler: httpevent
    events:
      - http:
          memorySize: 128
          path: hello
          method: get

  snsevent:
    handler: snsevent
    memorySize: 128
    events:
      - sns: my-sns-topic

  scheduledevent:
    handler: scheduledevent
    memorySize: 128
    events:
      - schedule:
          rate: rate(10 minutes)
          input:
            hello: world
```

demo

summary

easy

$$$

static binaries

aws library

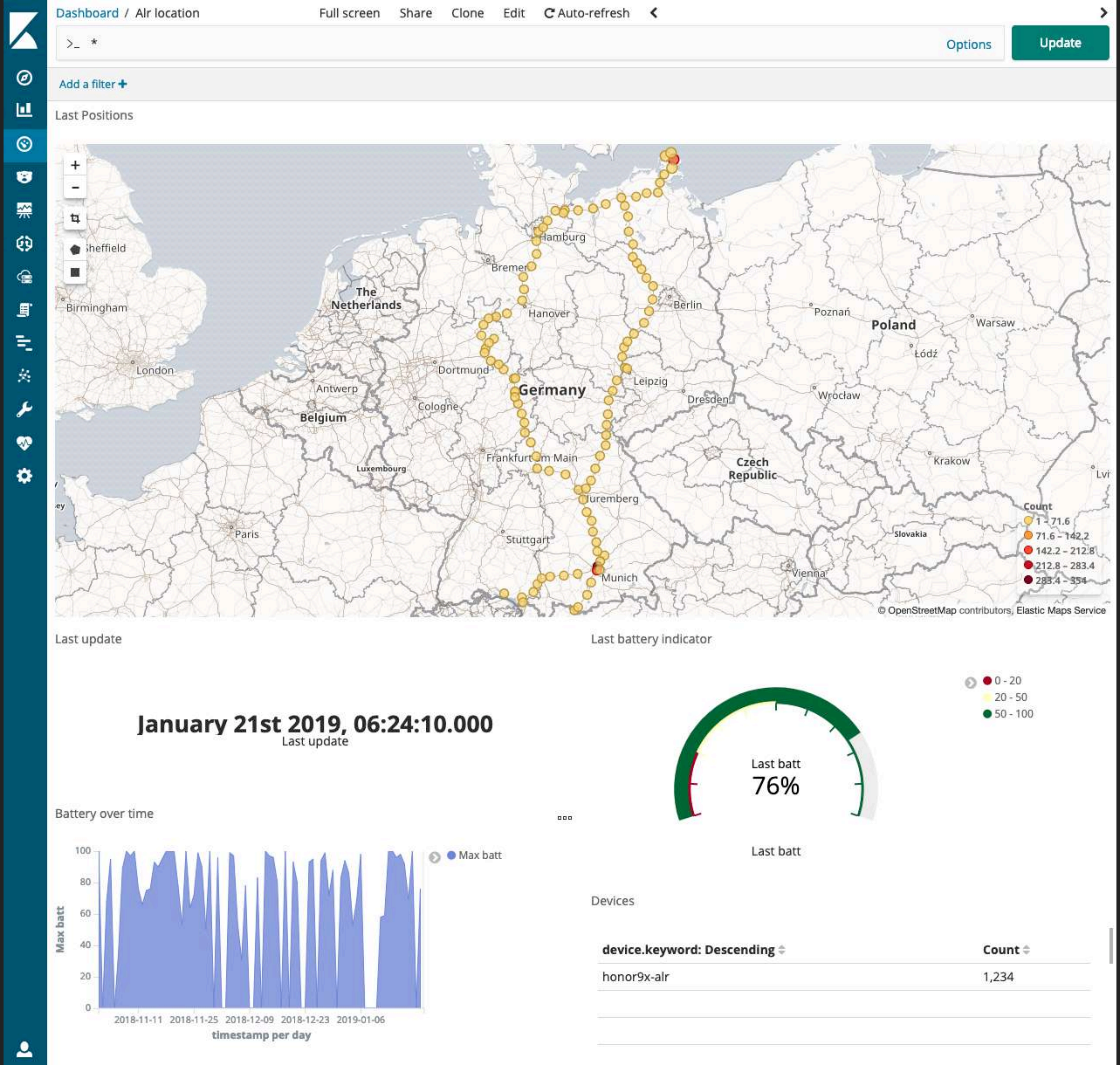
examples

static sites, dynamic

slack hook

github hook

location tracking



Last Positions

Last update

January 21st 2019, 06:24:10.000
Last update

Last battery indicator

Last batt
76%
Last batt

Battery over time

Devices

device.keyword: Descending	Count
honor9x-alr	1,234

scheduled jobs

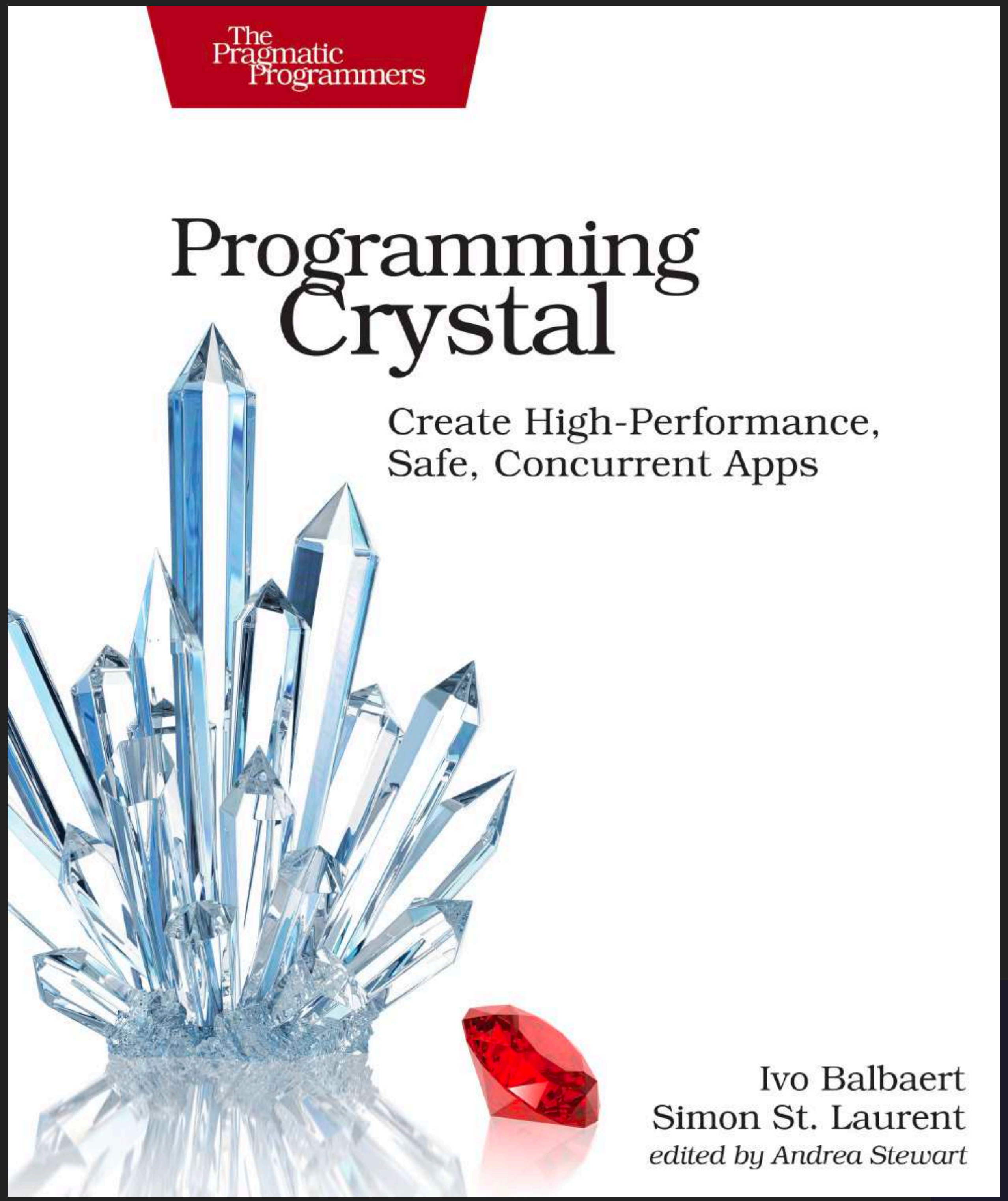
emails

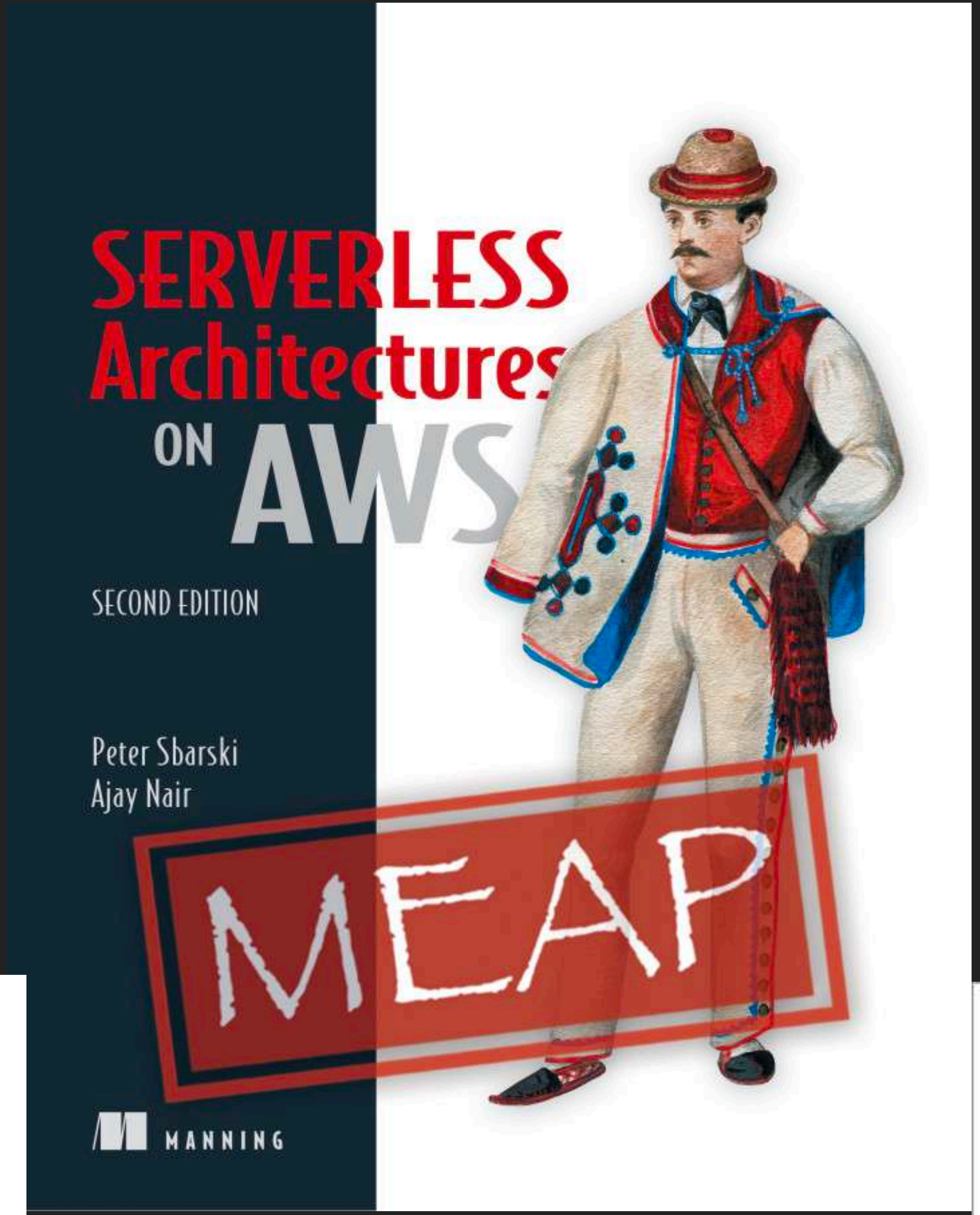
reverse geocoder

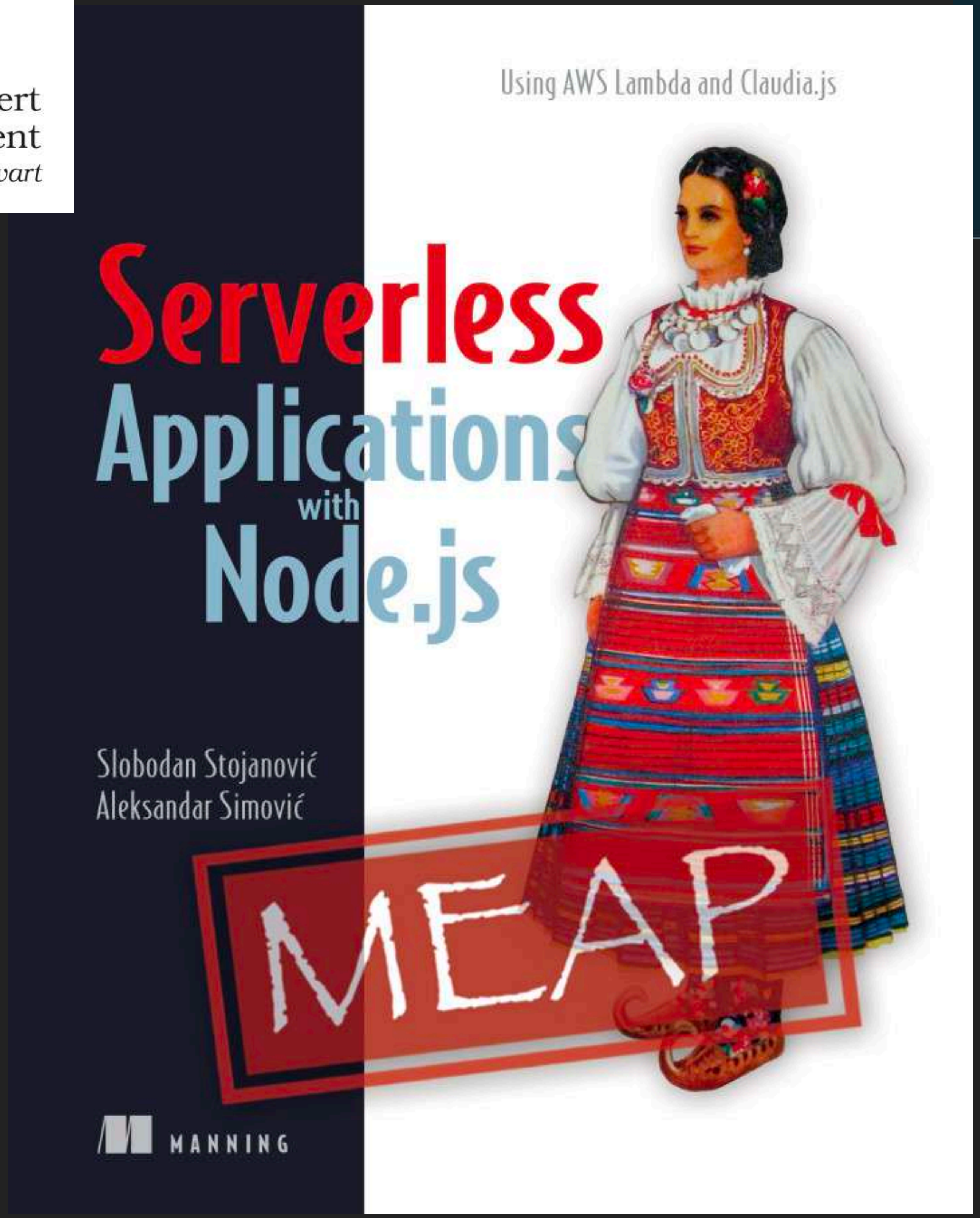
thank you

Links

- https://docs.aws.amazon.com/lambda/latest/dg/runtimes-custom.html
- https://docs.aws.amazon.com/lambda/latest/dg/runtimes-walkthrough.html
- https://crystal-lang.org
- https://github.com/spinscale/crystal-aws-lambda
- AWS Lambda Under the Hood: https://www.youtube.com/watch?v=QdzV04T_kec
- Firecracker: https://firecracker-microvm.github.io/
- Boehm GC: http://www.hboehm.info/gc/
- FaastRuby: https://faastruby.io/getting-started-crystal/
- Google Cloud Functions: https://github.com/sam0x17/gcf.cr
- Crystal on OpenFaas: https://github.com/TPei/crystal_openfaas


The Pragmatic Programmers
Programming Crystal
Create High-Performance, Safe, Concurrent Apps
Ivo Balbaert
Simon St. Laurent
edited by Andrea Stewart


SERVERLESS Architectures ON AWS
SECOND EDITION
Peter Sbarski
Ajay Nair
MEAP
MANNING


Using AWS Lambda and Claudia.js
Serverless Applications with Node.js
Slobodan Stojanović
Aleksandar Simović
MEAP
MANNING

The Pragmatic Programmers
Serverless Single Page Apps
Fast, Scalable, and Available
Ben Rady
edited by Jacquelyn Carter

@spinscale

alr@spinscale.de

questions?