FaaS You Like It!

@ewanslater

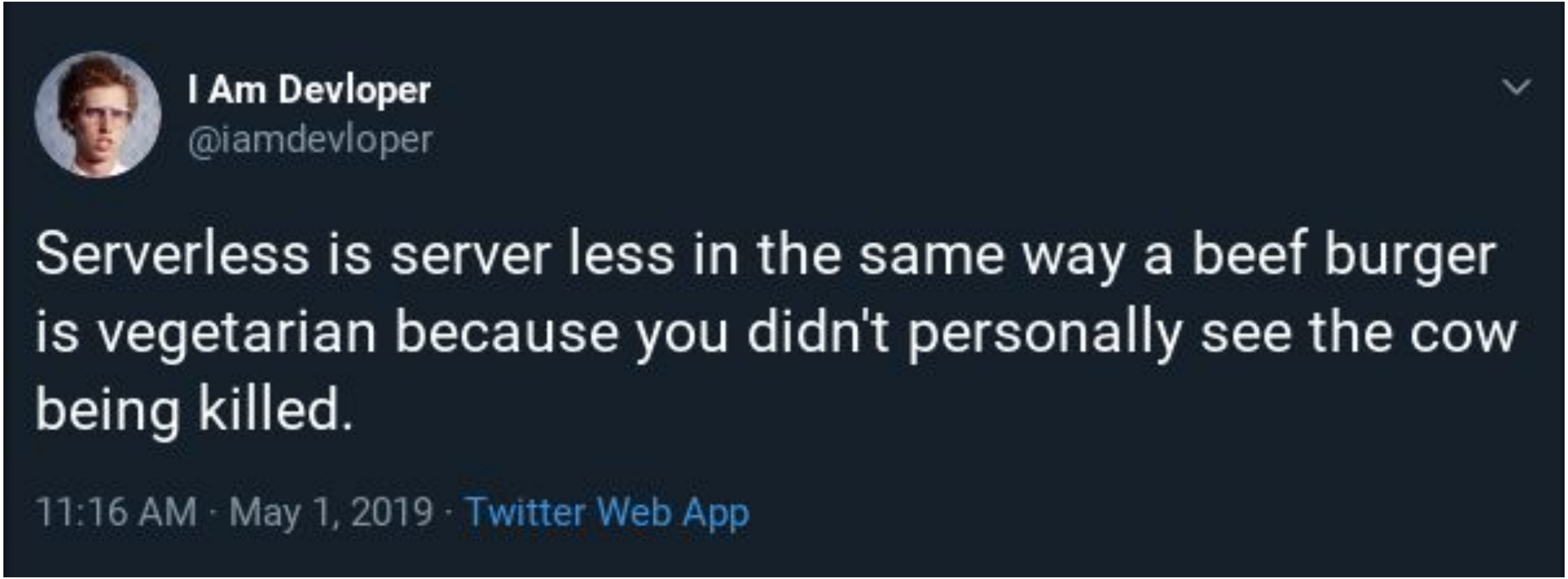
“Serverless”

CNCF Definition

“Serverless computing refers to the concept of building and running **applications** that do not **require server management**.

It describes a finer-grained deployment model where **applications**, bundled as one or more **functions**, are uploaded to a platform and then executed, scaled, and billed in response to the **exact demand needed at the moment**.”

- CNCF Serverless Whitepaper



I Am Devloper
@iamdevloper

Serverless is server less in the same way a beef burger is vegetarian because you didn't personally see the cow being killed.

11:16 AM · May 1, 2019 · Twitter Web App

Abstraction

TechAffect

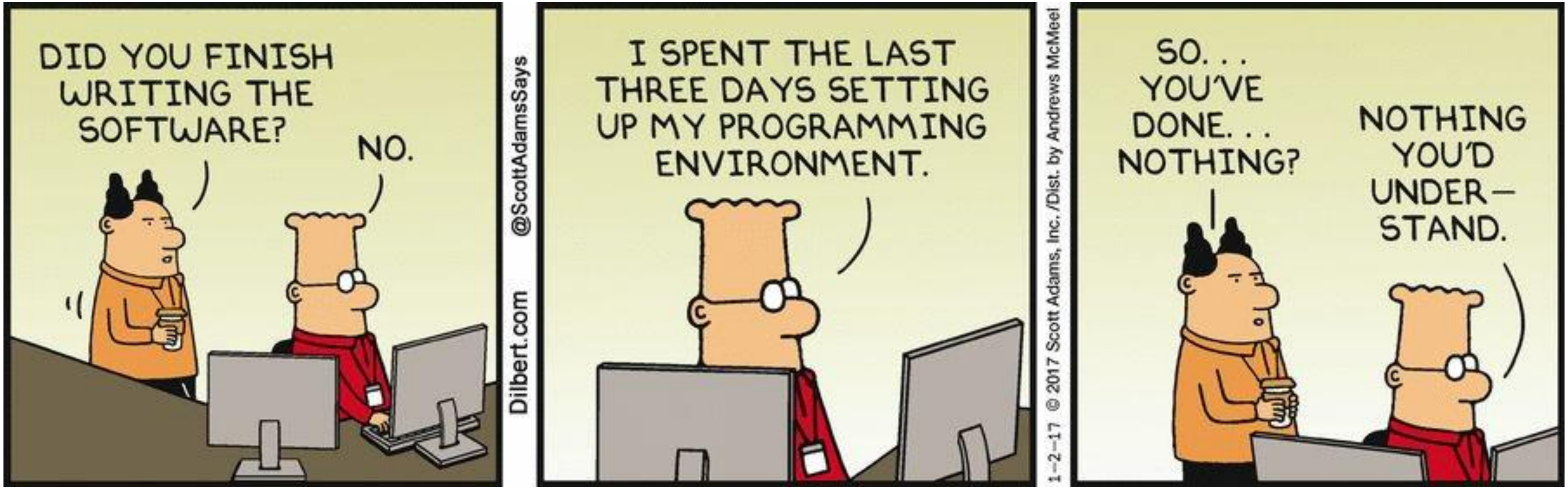
Functions

(as a service)

FaaS

- Small functions
- Do one thing well
- Easy to Understand
- Run on Serverless platform (i.e. abstracted infrastructure)

Avoid



DID YOU FINISH WRITING THE SOFTWARE?
NO.
@ScottAdamsSays
Dilbert.com
I SPENT THE LAST THREE DAYS SETTING UP MY PROGRAMMING ENVIRONMENT.
1-2-17 © 2017 Scott Adams, Inc. /Dist. by Andrews McMeel
SO... YOU'VE DONE... NOTHING?
NOTHING YOU'D UNDER-STAND.

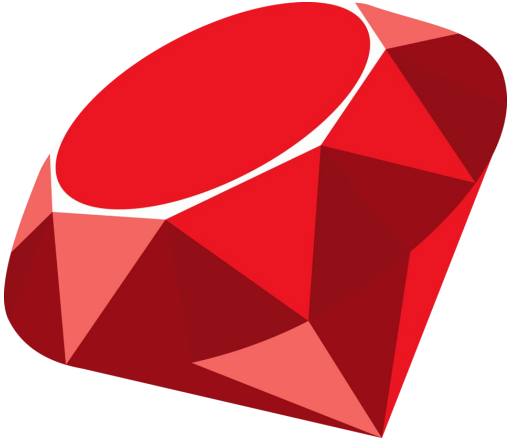
50
55
60
5
10
15
20
25
30
35
40
45
50
60
10
20
30
40
0.2 s

“Customers benefit from **increased programming productivity** and in many scenarios can enjoy **cost savings** as well.”

University of Berkeley,
Technical Report No. UCB/EECS-2019-3

Freedom(s)

- Portability
- Decentralisation
- Packaging
- Local testing
- Language



Fn Project

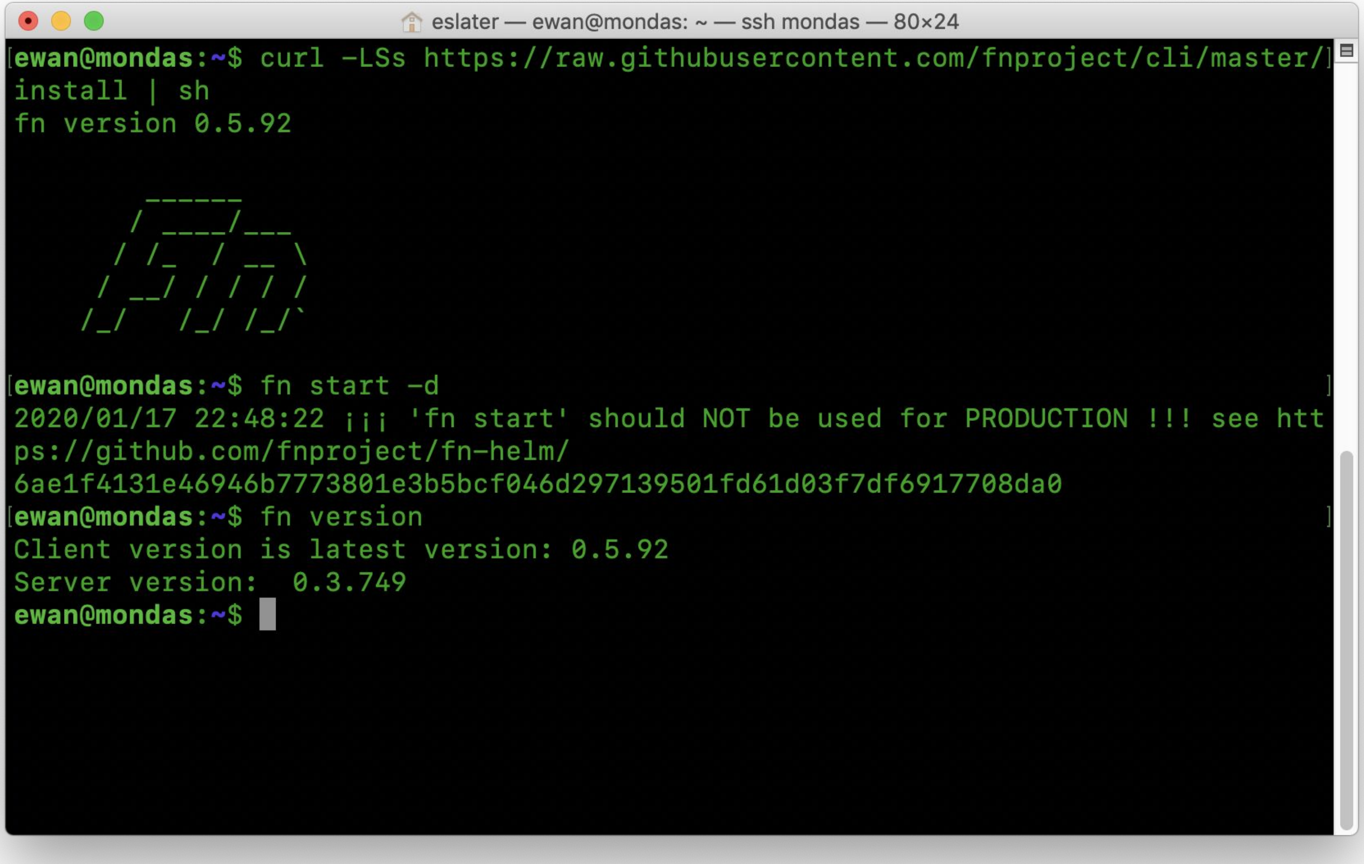
- Fnproject.io
- Serverless platform
- Container based
- Open Source (Apache 2.0)
- CNCF

Functions as Containers

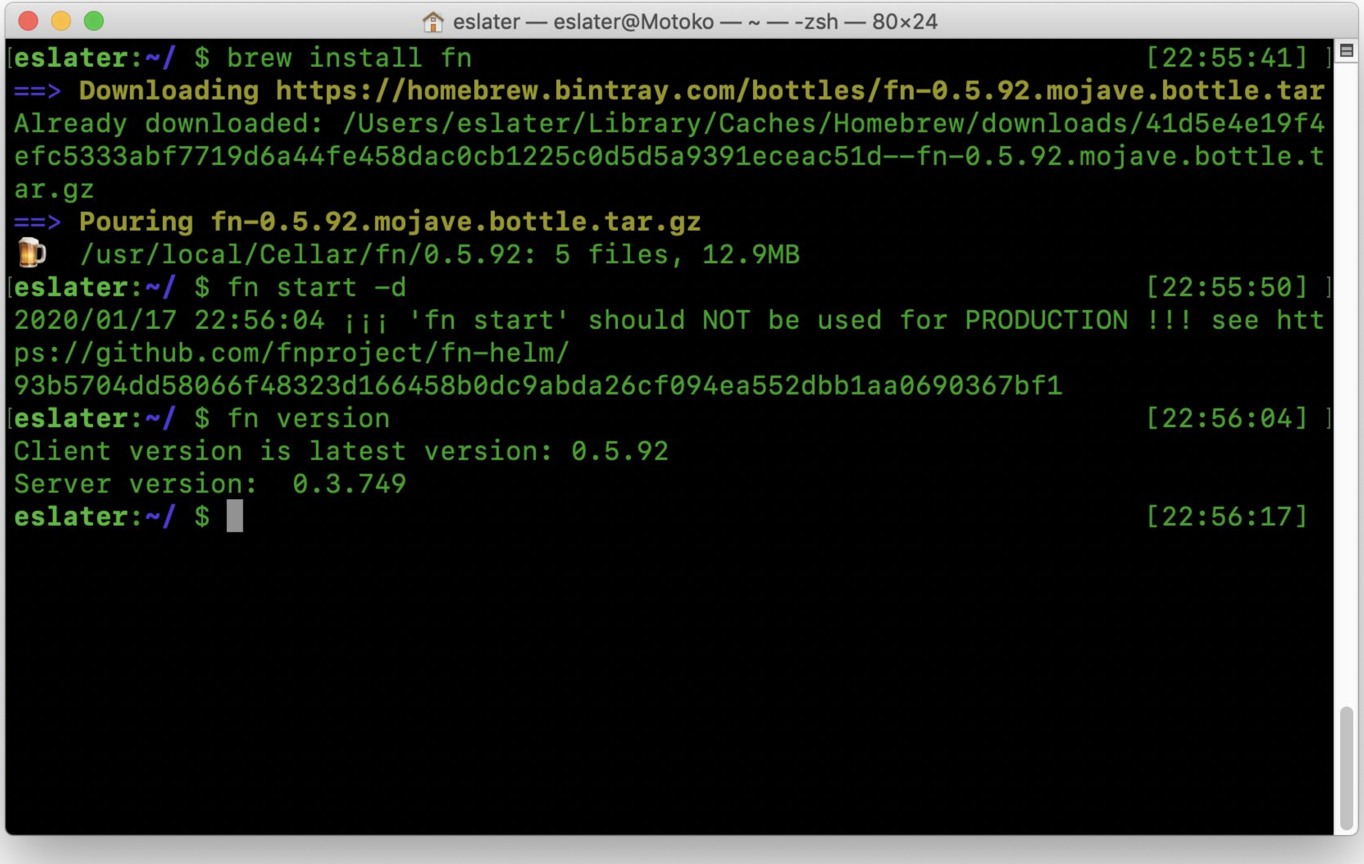
- Function
- Dependencies
- Self - Contained
- Stateless
- Ephemeral

Runs on

- Any Cloud
- On Premises
- Your laptop ;-)
- Anywhere you’ve installed Docker

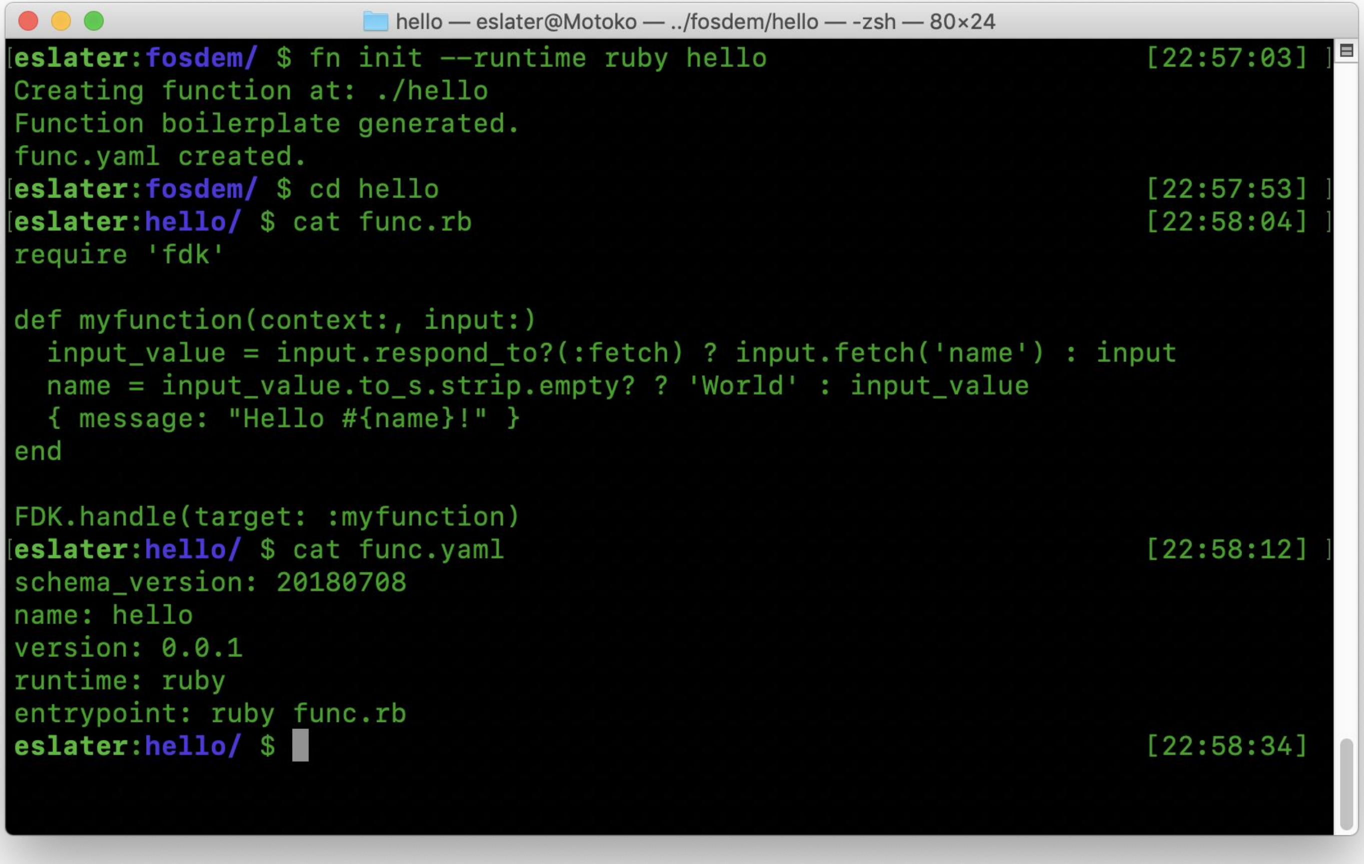


```
ewan@mondas:~$ curl -LSs https://raw.githubusercontent.com/fnproject/cli/master/
install | sh
fn version 0.5.92

        ______
       / ____/___
      / /_  / __ \
     / __/ / / / /
    /_/   /_/ /_/`

ewan@mondas:~$ fn start -d
2020/01/17 22:48:22 ¡¡¡ 'fn start' should NOT be used for PRODUCTION !!! see htt
ps://github.com/fnproject/fn-helm/
6ae1f4131e46946b7773801e3b5bcf046d297139501fd61d03f7df6917708da0
ewan@mondas:~$ fn version
Client version is latest version: 0.5.92
Server version:  0.3.749
ewan@mondas:~$ 
```

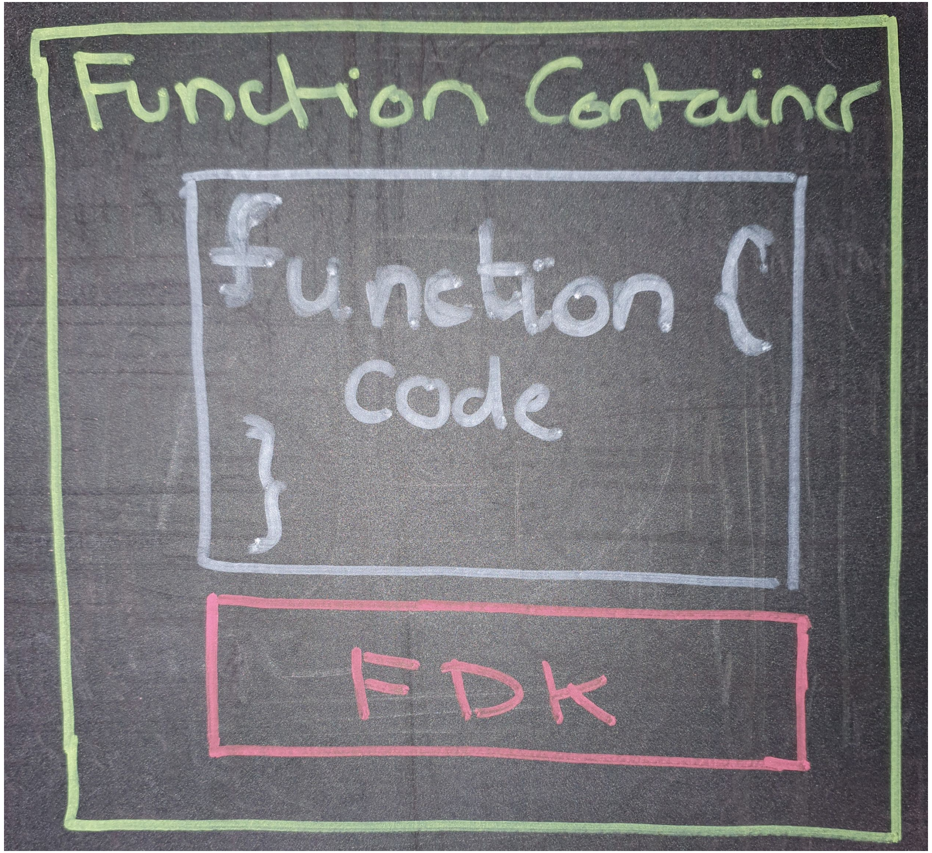


eslater — eslater@Motoko — ~ — -zsh — 80×24

```
eslater:~/ $ brew install fn                                        [22:55:41]
==> Downloading https://homebrew.bintray.com/bottles/fn-0.5.92.mojave.bottle.tar
Already downloaded: /Users/eslater/Library/Caches/Homebrew/downloads/41d5e4e19f4
efc5333abf7719d6a44fe458dac0cb1225c0d5d5a9391eceac51d--fn-0.5.92.mojave.bottle.t
ar.gz
==> Pouring fn-0.5.92.mojave.bottle.tar.gz
🍺  /usr/local/Cellar/fn/0.5.92: 5 files, 12.9MB
eslater:~/ $ fn start -d                                            [22:55:50]
2020/01/17 22:56:04 ¡¡¡ 'fn start' should NOT be used for PRODUCTION !!! see htt
ps://github.com/fnproject/fn-helm/
93b5704dd58066f48323d166458b0dc9abda26cf094ea552dbb1aa0690367bf1
eslater:~/ $ fn version                                             [22:56:04]
Client version is latest version: 0.5.92
Server version:  0.3.749
eslater:~/ $                                                        [22:56:17]
```

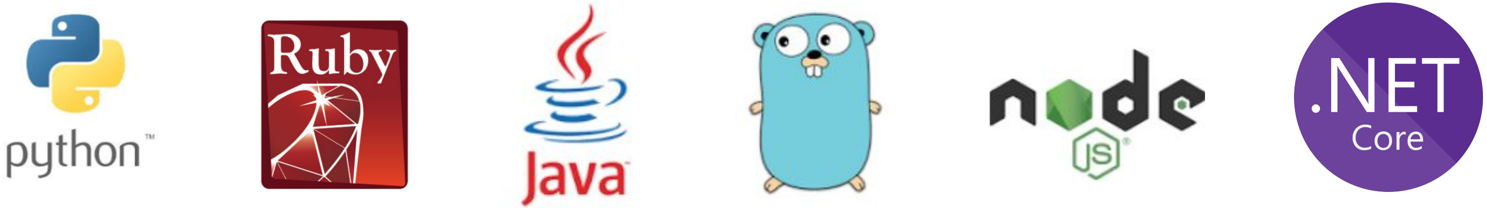


```
eslater:fosdem/ $ fn init --runtime ruby hello                     [22:57:03]
Creating function at: ./hello
Function boilerplate generated.
func.yaml created.
eslater:fosdem/ $ cd hello                                         [22:57:53]
eslater:hello/ $ cat func.rb                                       [22:58:04]
require 'fdk'

def myfunction(context:, input:)
  input_value = input.respond_to?(:fetch) ? input.fetch('name') : input
  name = input_value.to_s.strip.empty? ? 'World' : input_value
  { message: "Hello #{name}!" }
end

FDK.handle(target: :myfunction)
eslater:hello/ $ cat func.yaml                                     [22:58:12]
schema_version: 20180708
name: hello
version: 0.0.1
runtime: ruby
entrypoint: ruby func.rb
eslater:hello/ $                                                   [22:58:34]
```


Function Container
function {
code
}
FDK

Function Development Kits (FDKs)

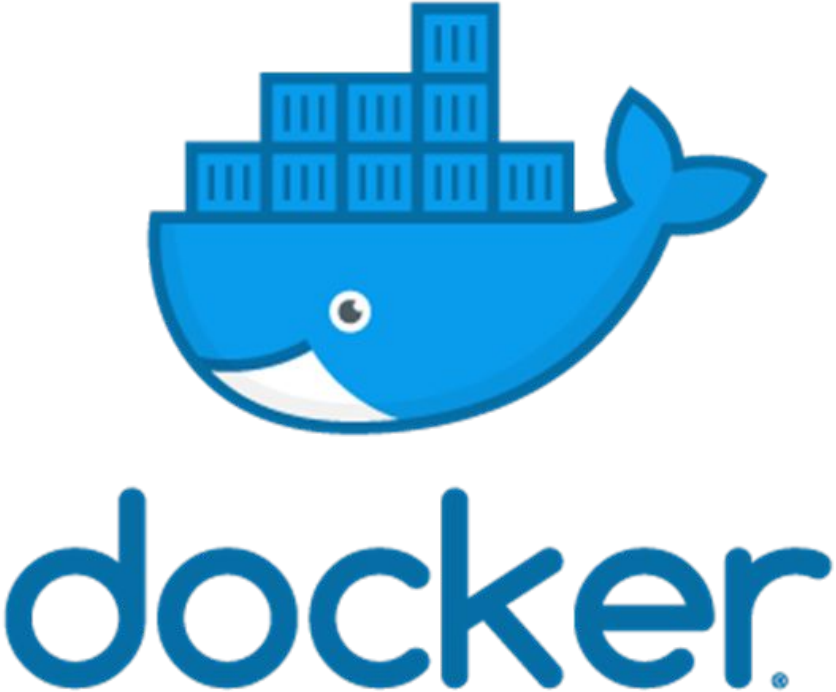
- Makes it a lot easier to write functions
- Developer includes FDK package / library / gem
- Developer writes function to the FDK’s interface
- FDK
    - Provides input data to function
    - Writes output & errors

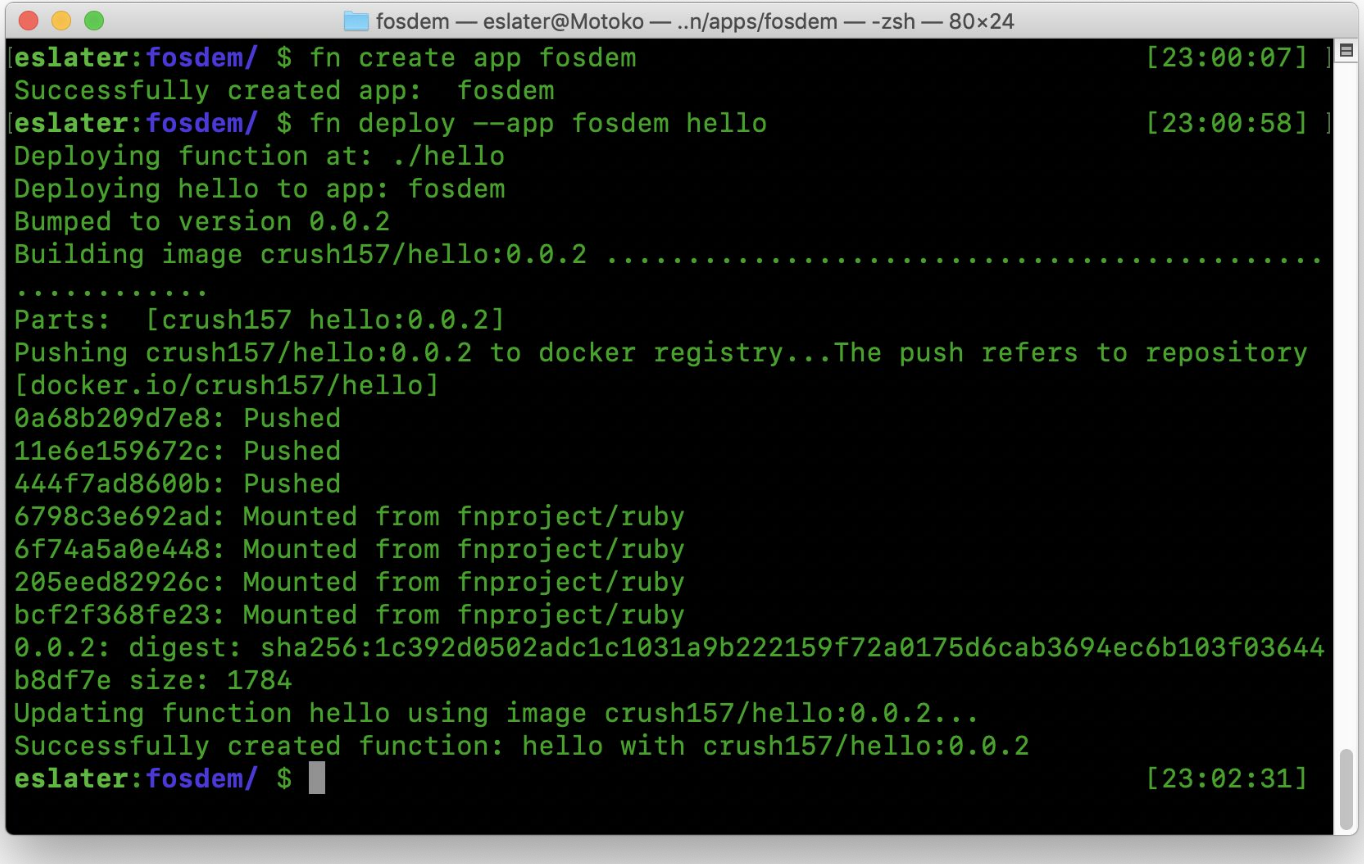


Ruby FDK

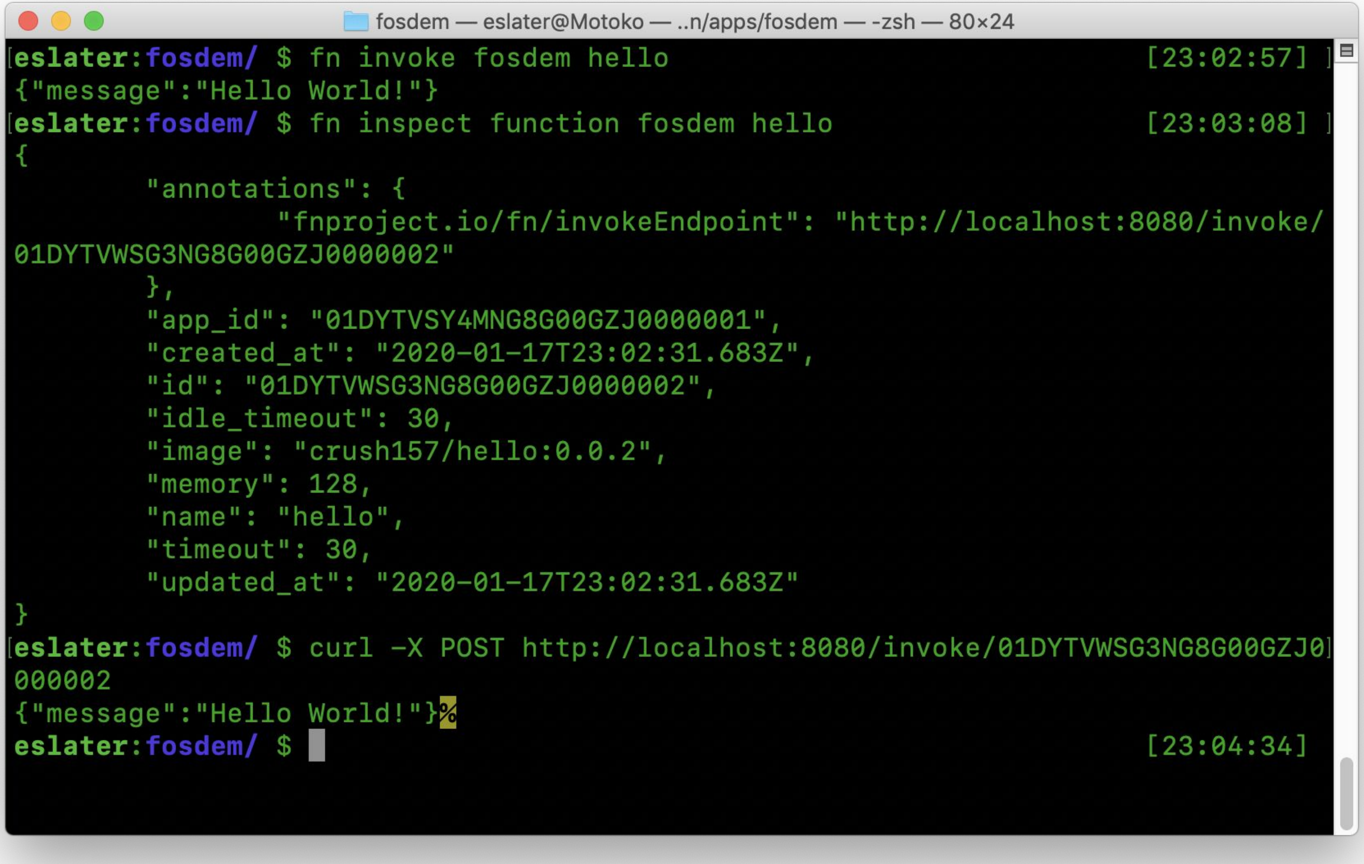
- Opens a socket in function container
  - (Fn Server connects to socket)
- Parses input from http-stream
- Executes function
  - Input
  - Context
- Sends back output on http-stream
- Writes errors to STDERR (→ syslog)

No FDK, no problem!

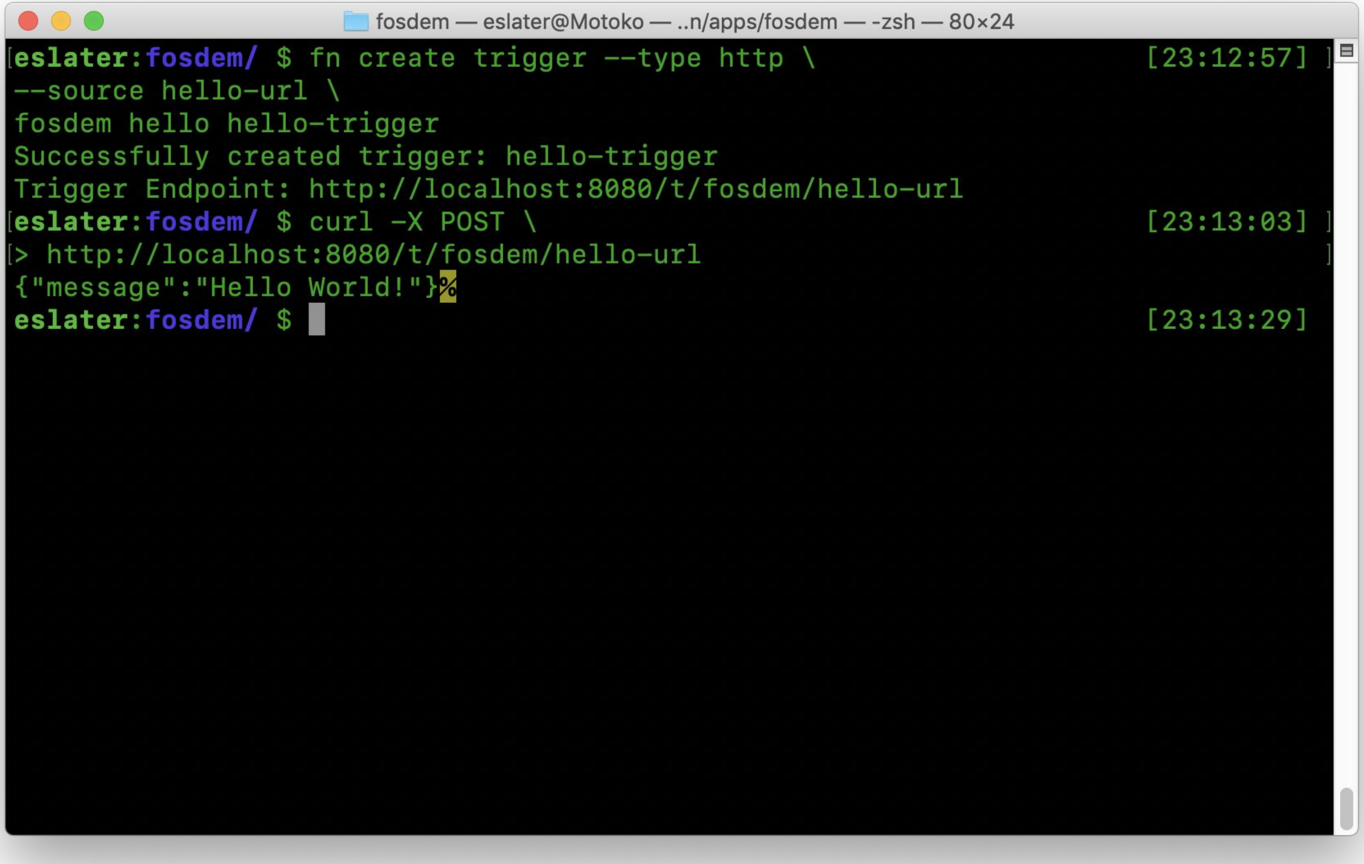




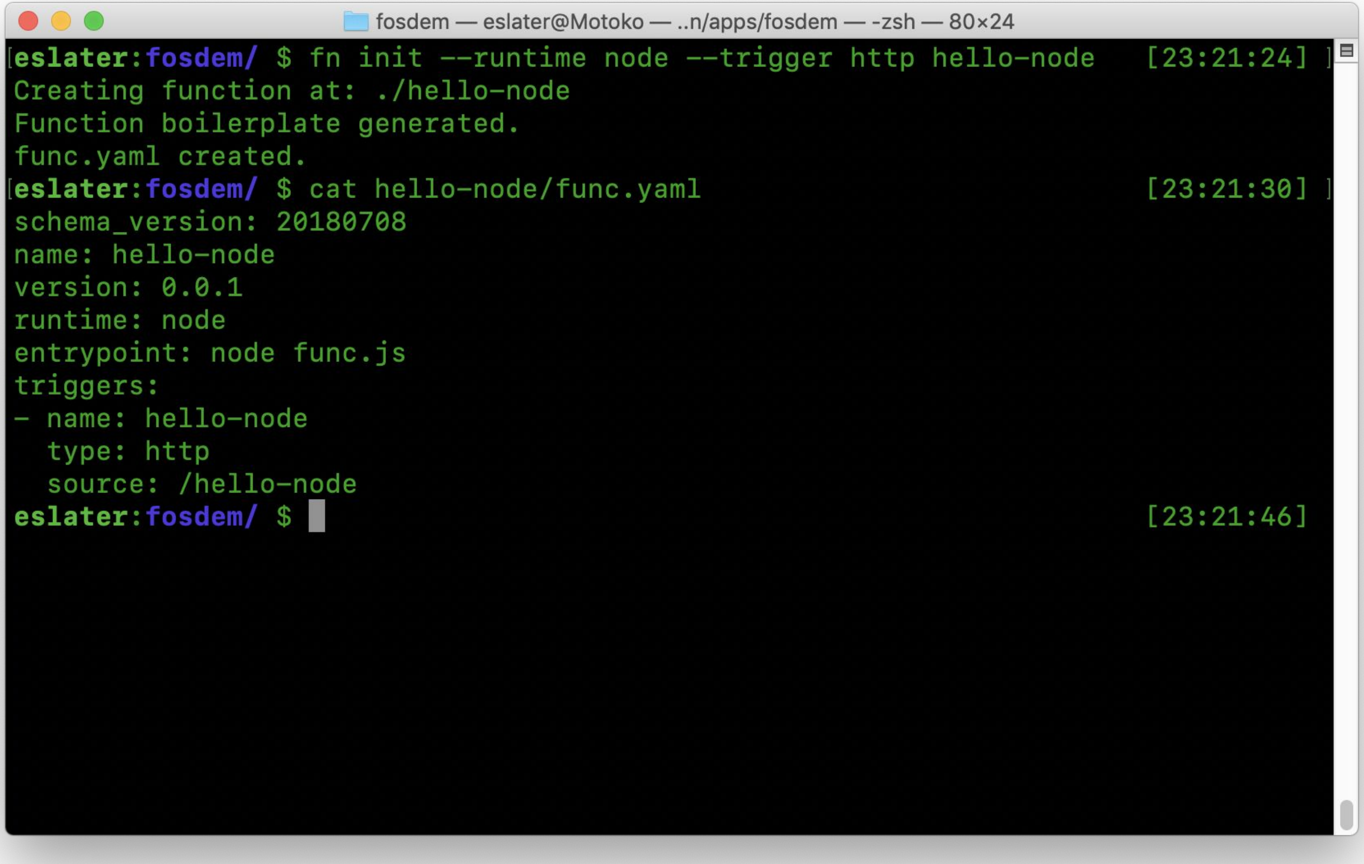
```
eslater:fosdem/ $ fn create app fosdem                              [23:00:07]
Successfully created app:  fosdem
eslater:fosdem/ $ fn deploy --app fosdem hello                      [23:00:58]
Deploying function at: ./hello
Deploying hello to app: fosdem
Bumped to version 0.0.2
Building image crush157/hello:0.0.2 ............................................
............
Parts:  [crush157 hello:0.0.2]
Pushing crush157/hello:0.0.2 to docker registry...The push refers to repository
[docker.io/crush157/hello]
0a68b209d7e8: Pushed
11e6e159672c: Pushed
444f7ad8600b: Pushed
6798c3e692ad: Mounted from fnproject/ruby
6f74a5a0e448: Mounted from fnproject/ruby
205eed82926c: Mounted from fnproject/ruby
bcf2f368fe23: Mounted from fnproject/ruby
0.0.2: digest: sha256:1c392d0502adc1c1031a9b222159f72a0175d6cab3694ec6b103f03644
b8df7e size: 1784
Updating function hello using image crush157/hello:0.0.2...
Successfully created function: hello with crush157/hello:0.0.2
eslater:fosdem/ $                                                   [23:02:31]
```



```
eslater:fosdem/ $ fn invoke fosdem hello                                [23:02:57]
{"message":"Hello World!"}
eslater:fosdem/ $ fn inspect function fosdem hello                      [23:03:08]
{
        "annotations": {
                "fnproject.io/fn/invokeEndpoint": "http://localhost:8080/invoke/
01DYTVWSG3NG8G00GZJ0000002"
        },
        "app_id": "01DYTVSY4MNG8G00GZJ0000001",
        "created_at": "2020-01-17T23:02:31.683Z",
        "id": "01DYTVWSG3NG8G00GZJ0000002",
        "idle_timeout": 30,
        "image": "crush157/hello:0.0.2",
        "memory": 128,
        "name": "hello",
        "timeout": 30,
        "updated_at": "2020-01-17T23:02:31.683Z"
}
eslater:fosdem/ $ curl -X POST http://localhost:8080/invoke/01DYTVWSG3NG8G00GZJ0
000002
{"message":"Hello World!"}%
eslater:fosdem/ $                                                       [23:04:34]
```

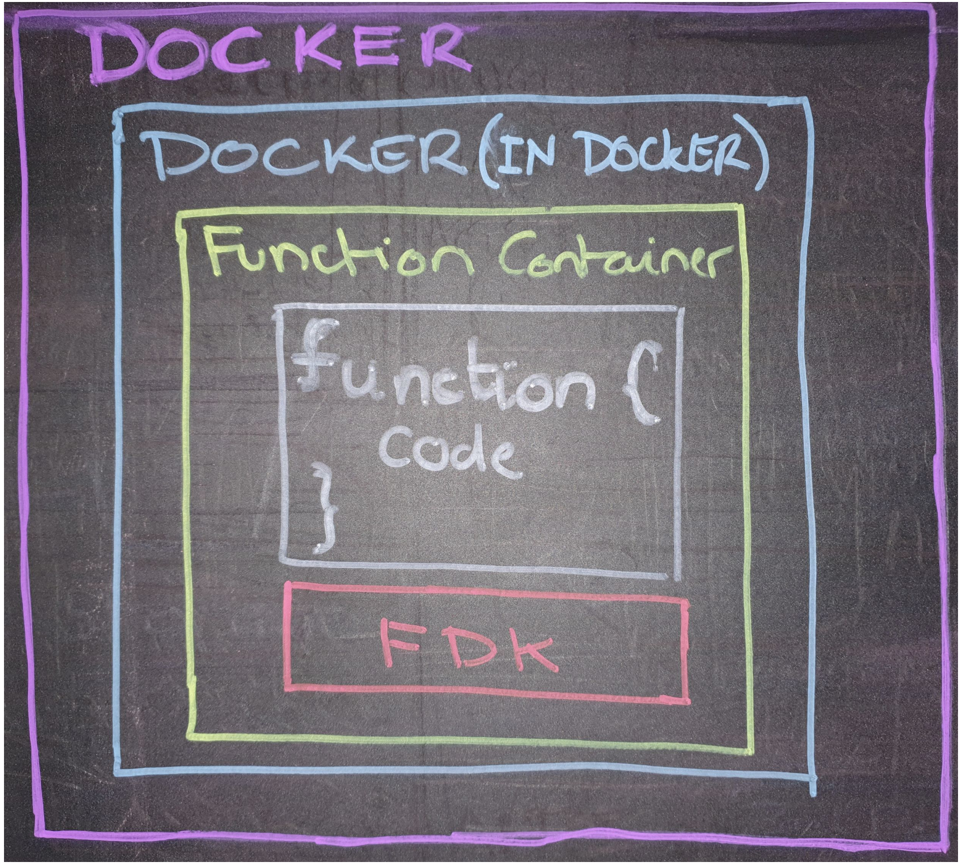


```
[eslater:fosdem/ $ fn create trigger --type http \                  [23:12:57] ]
--source hello-url \
fosdem hello hello-trigger
Successfully created trigger: hello-trigger
Trigger Endpoint: http://localhost:8080/t/fosdem/hello-url
[eslater:fosdem/ $ curl -X POST \                                   [23:13:03] ]
[> http://localhost:8080/t/fosdem/hello-url                                    ]
{"message":"Hello World!"}%
eslater:fosdem/ $                                                   [23:13:29]
```

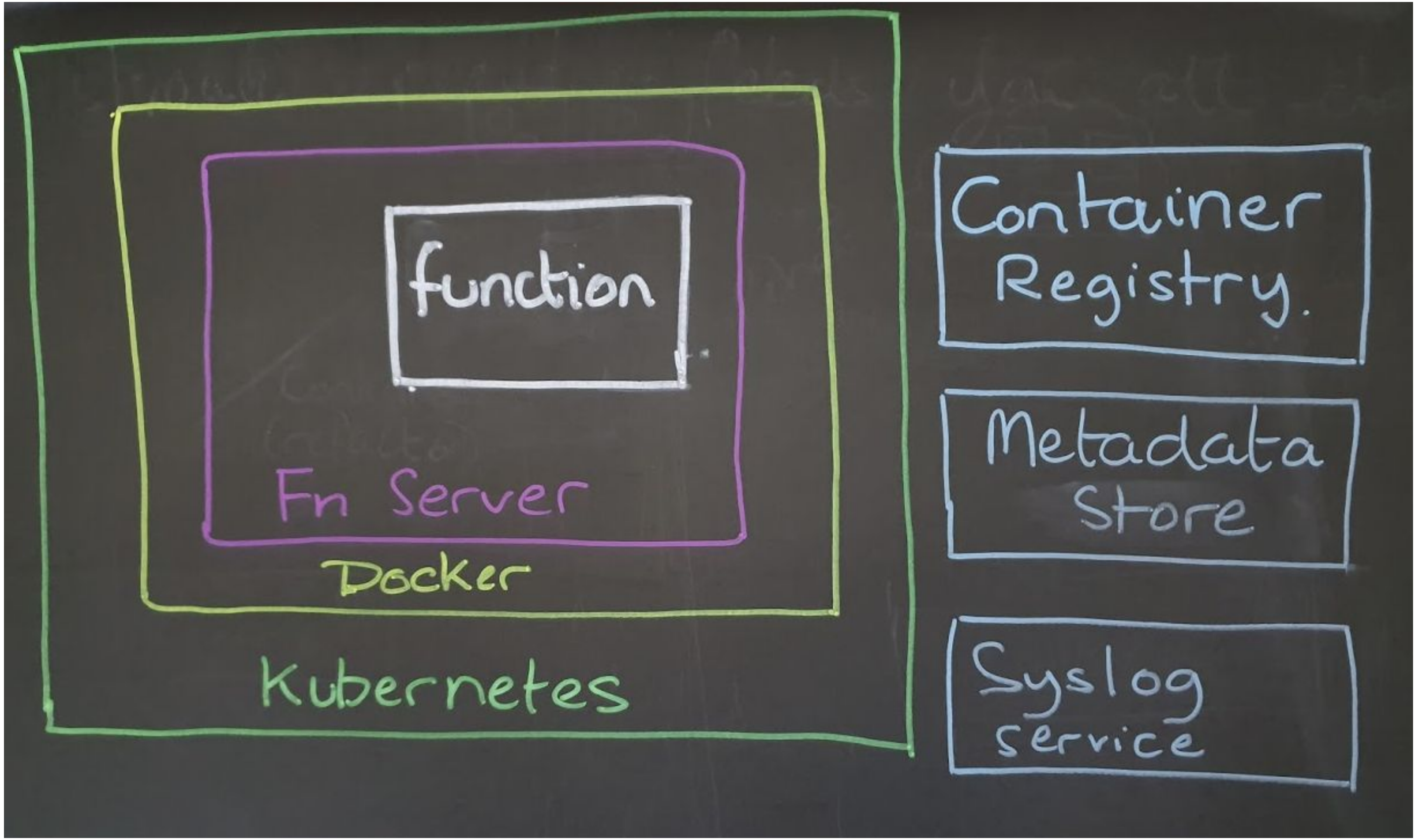


```
eslater:fosdem/ $ fn init --runtime node --trigger http hello-node   [23:21:24]
Creating function at: ./hello-node
Function boilerplate generated.
func.yaml created.
eslater:fosdem/ $ cat hello-node/func.yaml                           [23:21:30]
schema_version: 20180708
name: hello-node
version: 0.0.1
runtime: node
entrypoint: node func.js
triggers:
- name: hello-node
  type: http
  source: /hello-node
eslater:fosdem/ $                                                    [23:21:46]
```

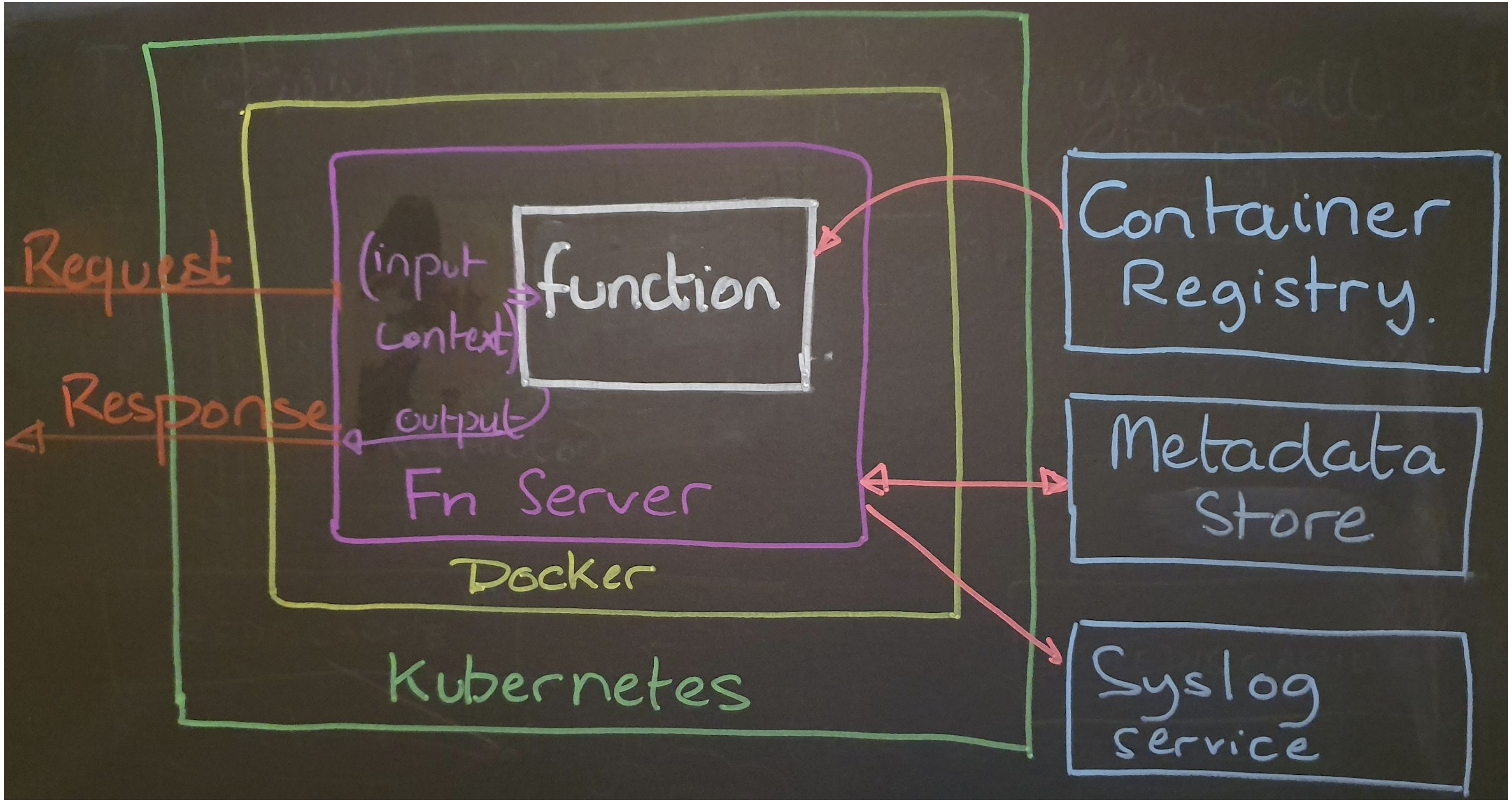
Fn Server

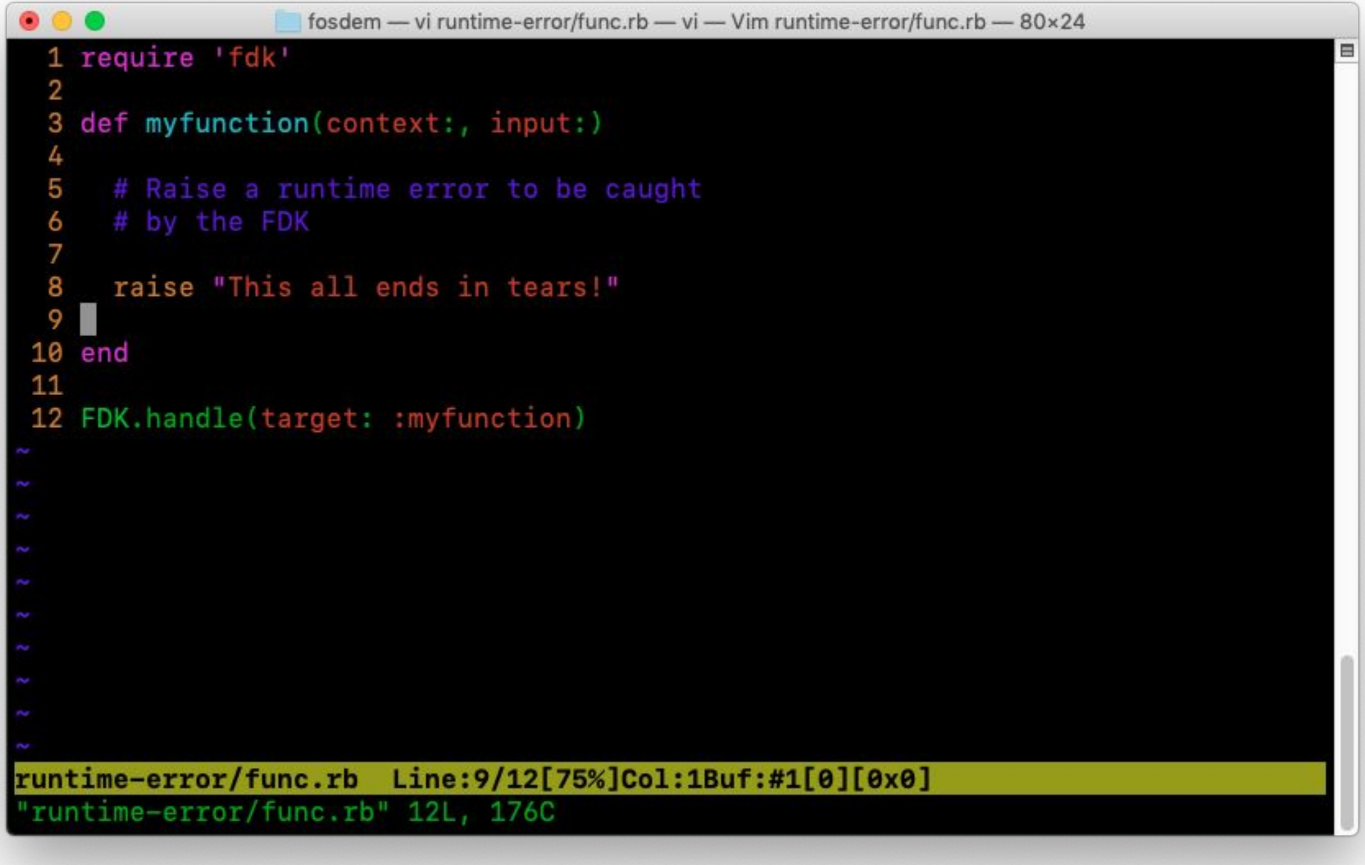

DOCKER
DOCKER (IN DOCKER)
Function Container
function {
code
}
FDK

Architecture (K8’s for production)

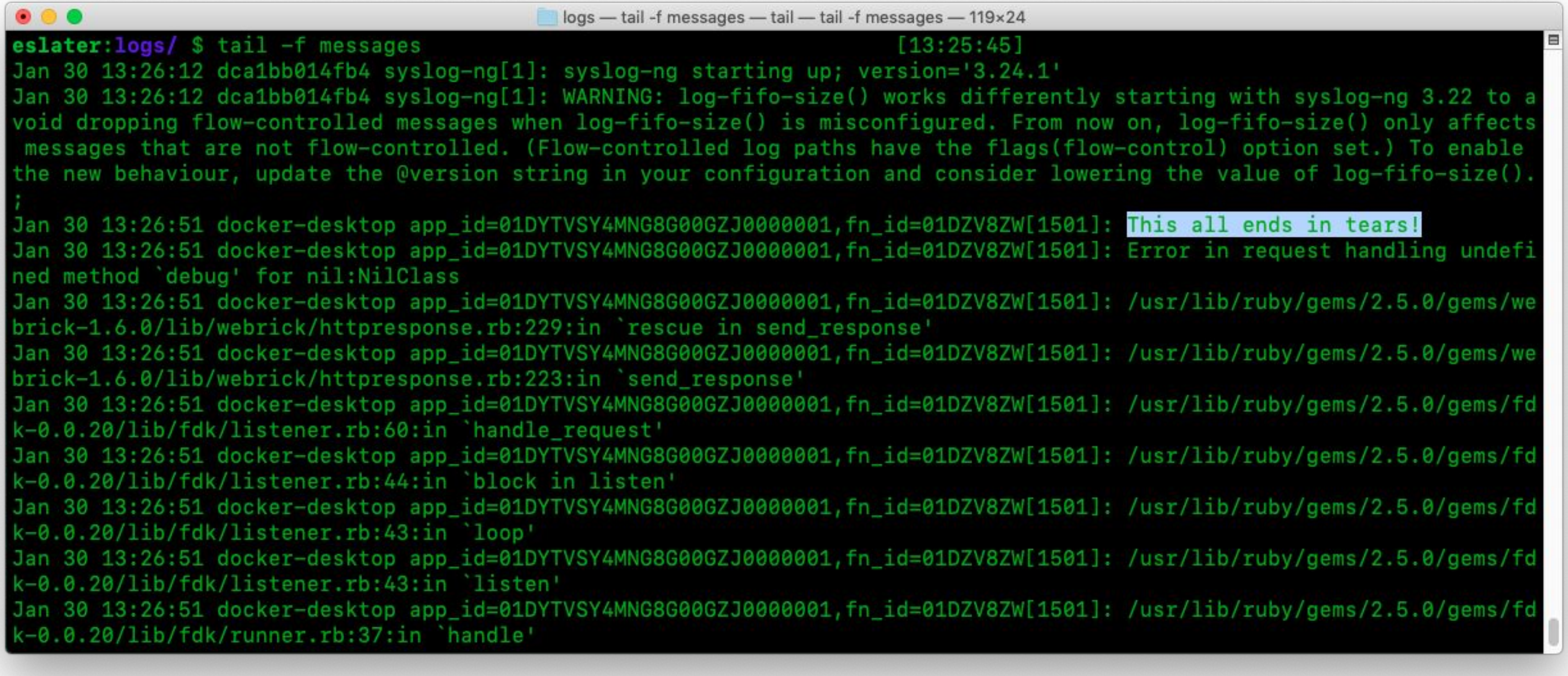


Request Process

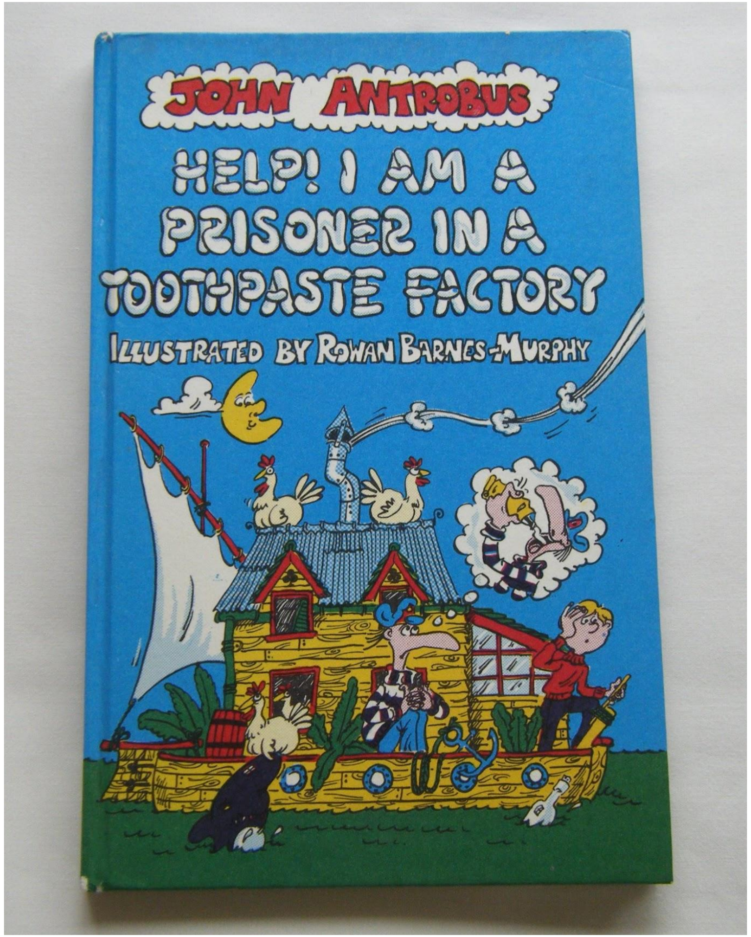



fosdem — vi runtime-error/func.rb — vi — Vim runtime-error/func.rb — 80×24

```ruby
require 'fdk'

def myfunction(context:, input:)

  # Raise a runtime error to be caught
  # by the FDK

  raise "This all ends in tears!"

end

FDK.handle(target: :myfunction)
~
~
~
~
~
~
~
~
~
~
runtime-error/func.rb  Line:9/12[75%]Col:1Buf:#1[0][0x0]
"runtime-error/func.rb" 12L, 176C
```



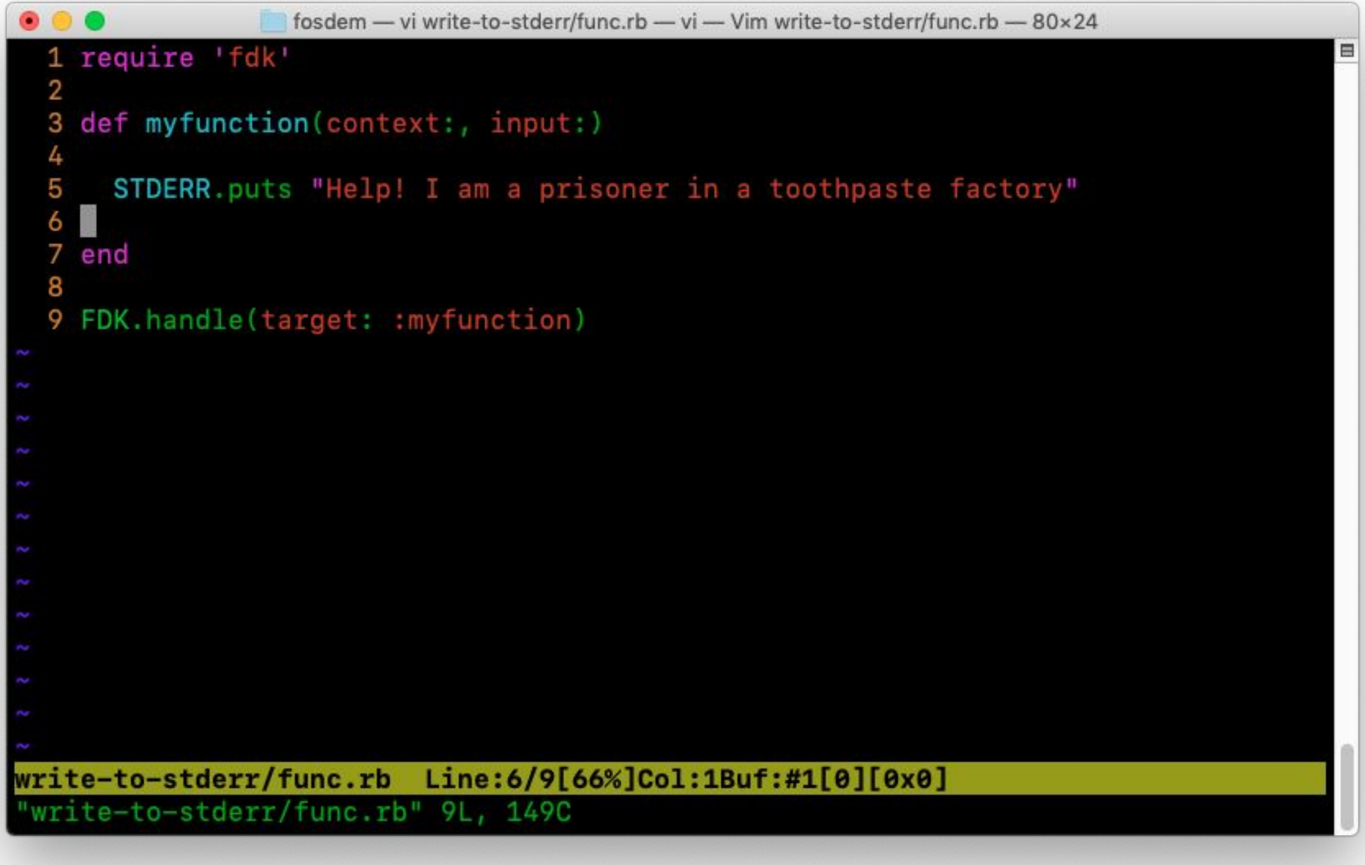
```
eslater:logs/ $ tail -f messages                                        [13:25:45]
Jan 30 13:26:12 dca1bb014fb4 syslog-ng[1]: syslog-ng starting up; version='3.24.1'
Jan 30 13:26:12 dca1bb014fb4 syslog-ng[1]: WARNING: log-fifo-size() works differently starting with syslog-ng 3.22 to a
void dropping flow-controlled messages when log-fifo-size() is misconfigured. From now on, log-fifo-size() only affects
 messages that are not flow-controlled. (Flow-controlled log paths have the flags(flow-control) option set.) To enable
the new behaviour, update the @version string in your configuration and consider lowering the value of log-fifo-size().
;
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: This all ends in tears!
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: Error in request handling undefi
ned method `debug' for nil:NilClass
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/we
brick-1.6.0/lib/webrick/httpresponse.rb:229:in `rescue in send_response'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/we
brick-1.6.0/lib/webrick/httpresponse.rb:223:in `send_response'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/fd
k-0.0.20/lib/fdk/listener.rb:60:in `handle_request'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/fd
k-0.0.20/lib/fdk/listener.rb:44:in `block in listen'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/fd
k-0.0.20/lib/fdk/listener.rb:43:in `loop'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/fd
k-0.0.20/lib/fdk/listener.rb:43:in `listen'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]: /usr/lib/ruby/gems/2.5.0/gems/fd
k-0.0.20/lib/fdk/runner.rb:37:in `handle'
```



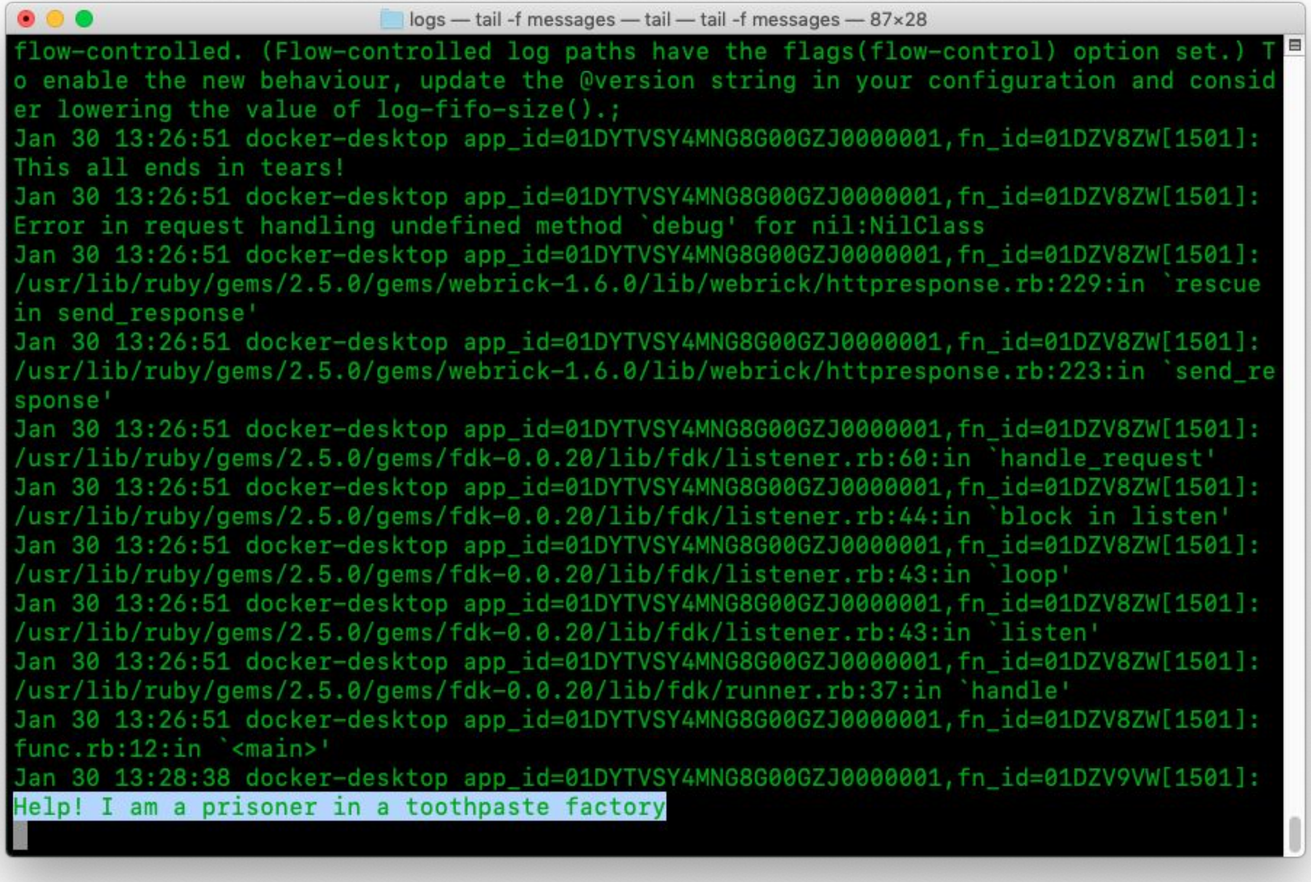
JOHN ANTROBUS

HELP! I AM A PRISONER IN A TOOTHPASTE FACTORY

ILLUSTRATED BY ROWAN BARNES-MURPHY


fosdem — vi write-to-stderr/func.rb — vi — Vim write-to-stderr/func.rb — 80×24

```
require 'fdk'

def myfunction(context:, input:)

  STDERR.puts "Help! I am a prisoner in a toothpaste factory"

end

FDK.handle(target: :myfunction)
~
~
~
~
~
~
~
~
~
~
~
~
~
write-to-stderr/func.rb  Line:6/9[66%]Col:1Buf:#1[0][0x0]
"write-to-stderr/func.rb" 9L, 149C
```



```
flow-controlled. (Flow-controlled log paths have the flags(flow-control) option set.) T
o enable the new behaviour, update the @version string in your configuration and consid
er lowering the value of log-fifo-size().;
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
This all ends in tears!
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
Error in request handling undefined method `debug' for nil:NilClass
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/webrick-1.6.0/lib/webrick/httpresponse.rb:229:in `rescue
in send_response'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/webrick-1.6.0/lib/webrick/httpresponse.rb:223:in `send_re
sponse'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/fdk-0.0.20/lib/fdk/listener.rb:60:in `handle_request'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/fdk-0.0.20/lib/fdk/listener.rb:44:in `block in listen'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/fdk-0.0.20/lib/fdk/listener.rb:43:in `loop'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/fdk-0.0.20/lib/fdk/listener.rb:43:in `listen'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
/usr/lib/ruby/gems/2.5.0/gems/fdk-0.0.20/lib/fdk/runner.rb:37:in `handle'
Jan 30 13:26:51 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV8ZW[1501]:
func.rb:12:in `<main>'
Jan 30 13:28:38 docker-desktop app_id=01DYTVSY4MNG8G00GZJ0000001,fn_id=01DZV9VW[1501]:
Help! I am a prisoner in a toothpaste factory
```

Hot functions

- Run *at least* once

- Wait before terminating
  - Default timeout 30 seconds
  - Configurable

- Potentially reusable
  - FDK handles multiple invocations
  - Stateless remember!?

Orchestration

Flow

- Promises Based Orchestration
  - No new “special language”
  - Composable
  - Flexible

- A Flow is…
  - ...written in code
  - ...a function
  - ...deployed within an app

FaaS You Like It*

(*with apologies to Shakespeare)

Serverless

- Abstracted Infrastructure
- Ephemeral Functions
- Higher Productivity
- Lower Costs
- Beware of Lock In

Fn Project

- Open Source Serverless Platform
- Container Based
- Any language
- Any Cloud
- On Premises
- Your Laptop ;-)

Questions?

(grab me afterwards || @ewanslater)