What's new in OpenResty for 2016

☺ *agentzh@gmail.com* ☺

Yichun Zhang (agentzh)

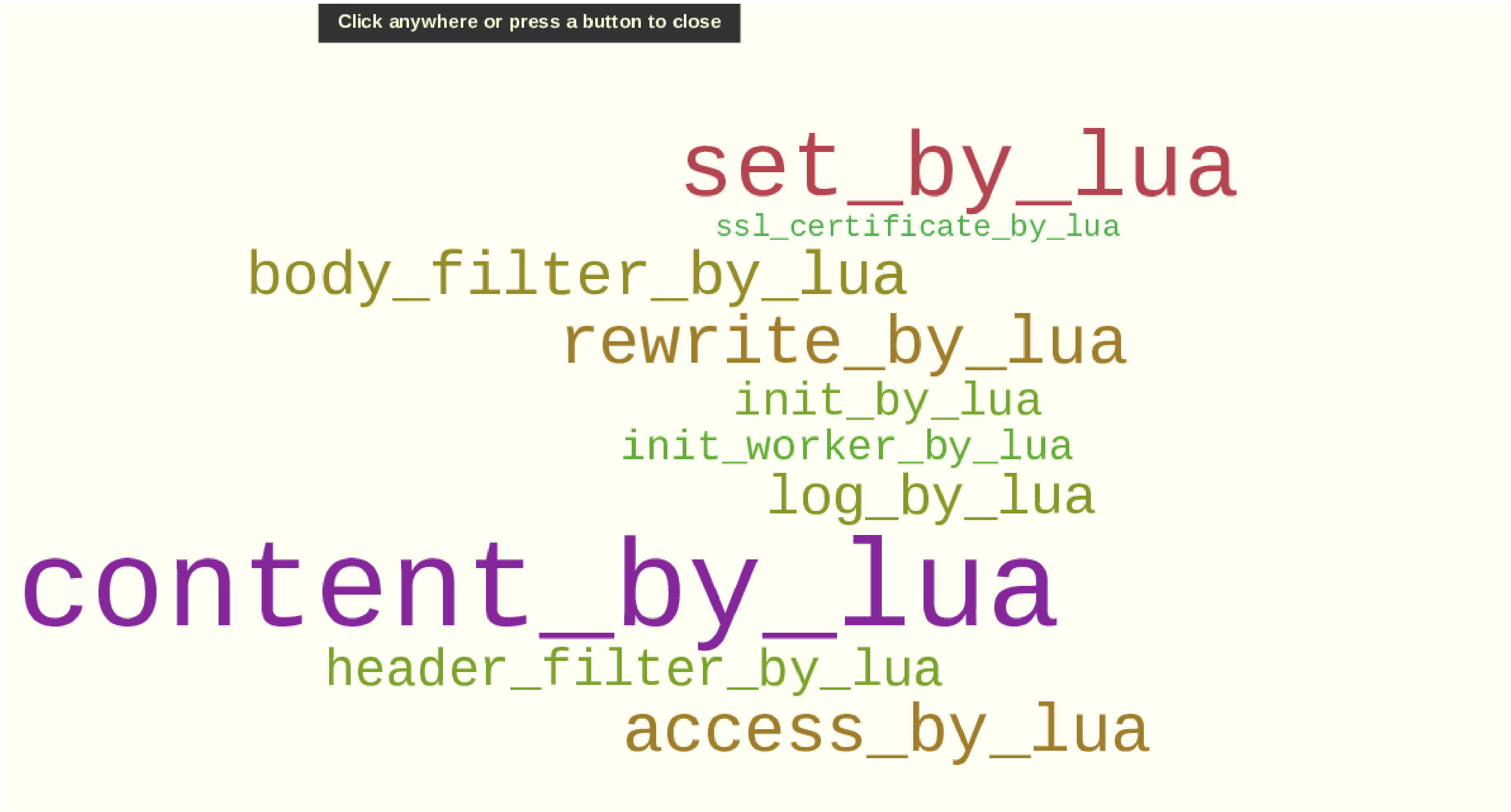
2016.3

☺ ngx_stream_lua_module

(vs ngx_http_lua_module)

```
stream {
    # define a TCP server listening on
    # the port 1234:
    server {
        listen 1234;

        content_by_lua_block {
            ngx.say("Hello, Lua!")
        }
    }
}
```

```
stream {
    server {
        listen 4343 ssl;

        ssl_certificate     /path/to/cert.pem;
        ssl_certificate_key /path/to/cert.key;

        content_by_lua_block {
            local sock = ngx.req.socket(true)

            -- read a line from client
            local data = sock:receive()
            if data == "thunder!" then
                ngx.say("flash!")  -- output data
            end
        }
    }
}
```

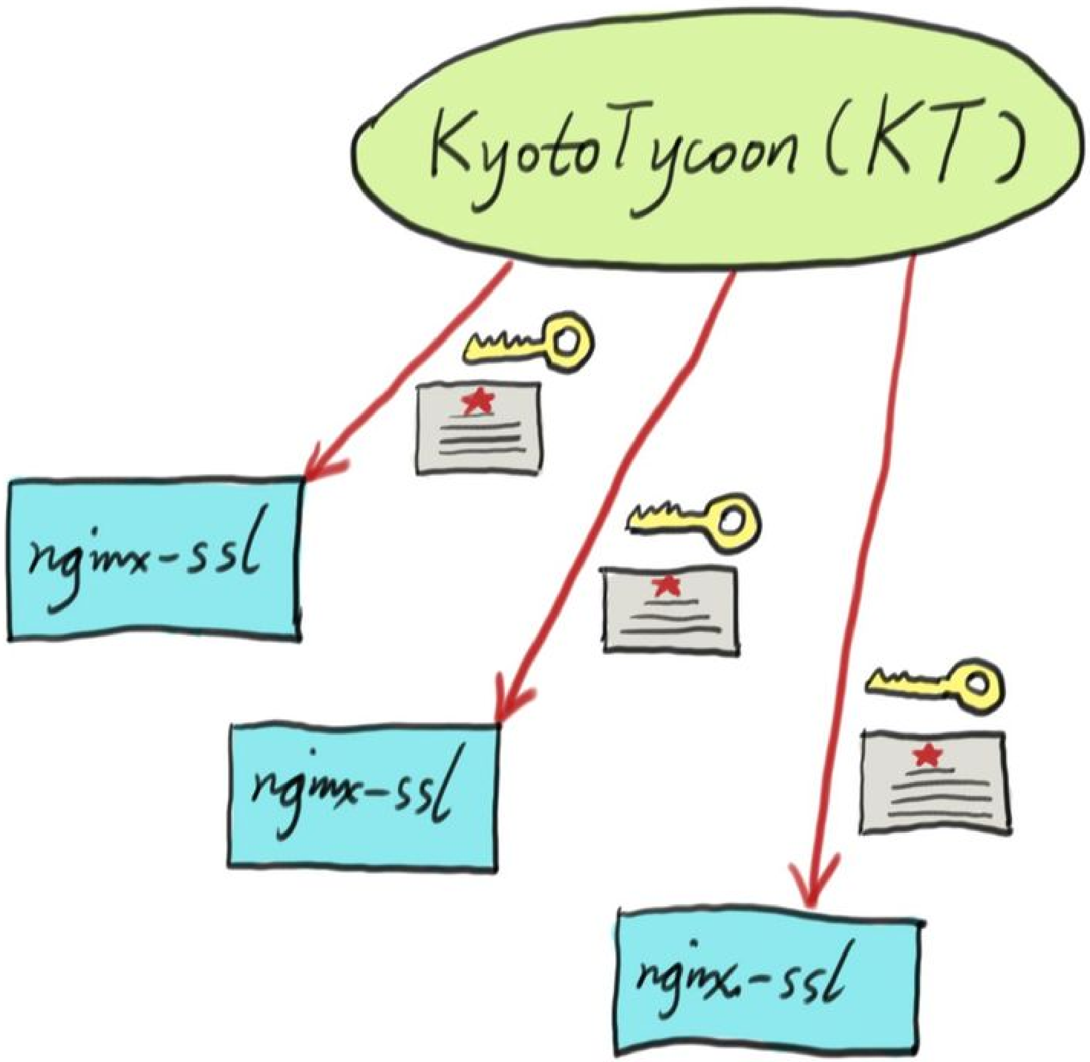

Click anywhere or press a button to close
set_by_lua
ssl_certificate_by_lua
body_filter_by_lua
rewrite_by_lua
init_by_lua
init_worker_by_lua
log_by_lua
content_by_lua
header_filter_by_lua
access_by_lua

☺ ssl_certificate_by_lua*

ngx.ssl

ngx.ocsp

```
ssl_certificate_by_lua_block {
    local ssl = require "ngx.ssl"

    -- clear default certificates and private keys
    ssl.clear_certs()

    -- set certificate chain in DER format for the
    -- current SSL session
    ssl.set_der_cert(my_cert_chain)

    -- set private key in DER format for the current
    -- SSL session
    ssl.set_der_priv_key(my_private_key)
}
```


KyotoTycoon (KT)
nginx-ssl
nginx-ssl
nginx-ssl

☺ ssl_session_fetch_by_lua*
ssl_session_store_by_lua*

☺ balancer_by_lua*

ngx.balancer

```
upstream my_backend {
    server 0.0.0.1;   # just a place holder

    balancer_by_lua_block {
        local balancer = require "ngx.balancer"

        local host = "10.26.4.2"
        local port = 8080

        balancer.set_current_peer(host, port)

        -- Note: you can control retries here as well.
    }

    keepalive 10;  # configure connection pool
}
```

```
server {
    location / {
        proxy_pass http://my_backend;
    }
}
```

☺ ngx.semaphore

```lua
local semaphore = require "ngx.semaphore"
local sema = semaphore.new()
```

```
-- producer
sema:post(2)   -- post 2 resources
```

```
-- consumer, maybe in another request handler
-- wait for a resource, 3 sec timeout
local ok, err =  sema:wait(3)
```

☺ Lists or *queues* in lua_shared_dict

```
local animals = ngx.shared.animals

-- apend an item to the end of the list
under key "dogs"
local cnt, err = animals:rpush("dogs", "Tom")

-- append another item
cnt, err = animals:rpush("dogs", "Bob")

-- pop out an item from the beginning
of the list under key "dogs"
cnt, err = animals:lpop("dogs")
```

☺ C2000K

☺ Lemplate

TT2 templates ⇒ Lua code

```
<!-- file a.tt2 -->
<html><body>
<ul>
 [% FOREACH v IN list -%]
   <li>[% v | html %]</li>
 [% END -%]
</ul>
</body></html>
```

```
$ lemplate --compile a.tt2 > my_templates.lua
```

```
location = /animals {
    content_by_lua_block {
        local templates = require "my_templates"
        local animals = { "cats", "dogs", "birds" }

        local html = templates.process(
                "a.tt2",  -- only as a key
                { list = animals })

        ngx.print(html)
    }
}
```

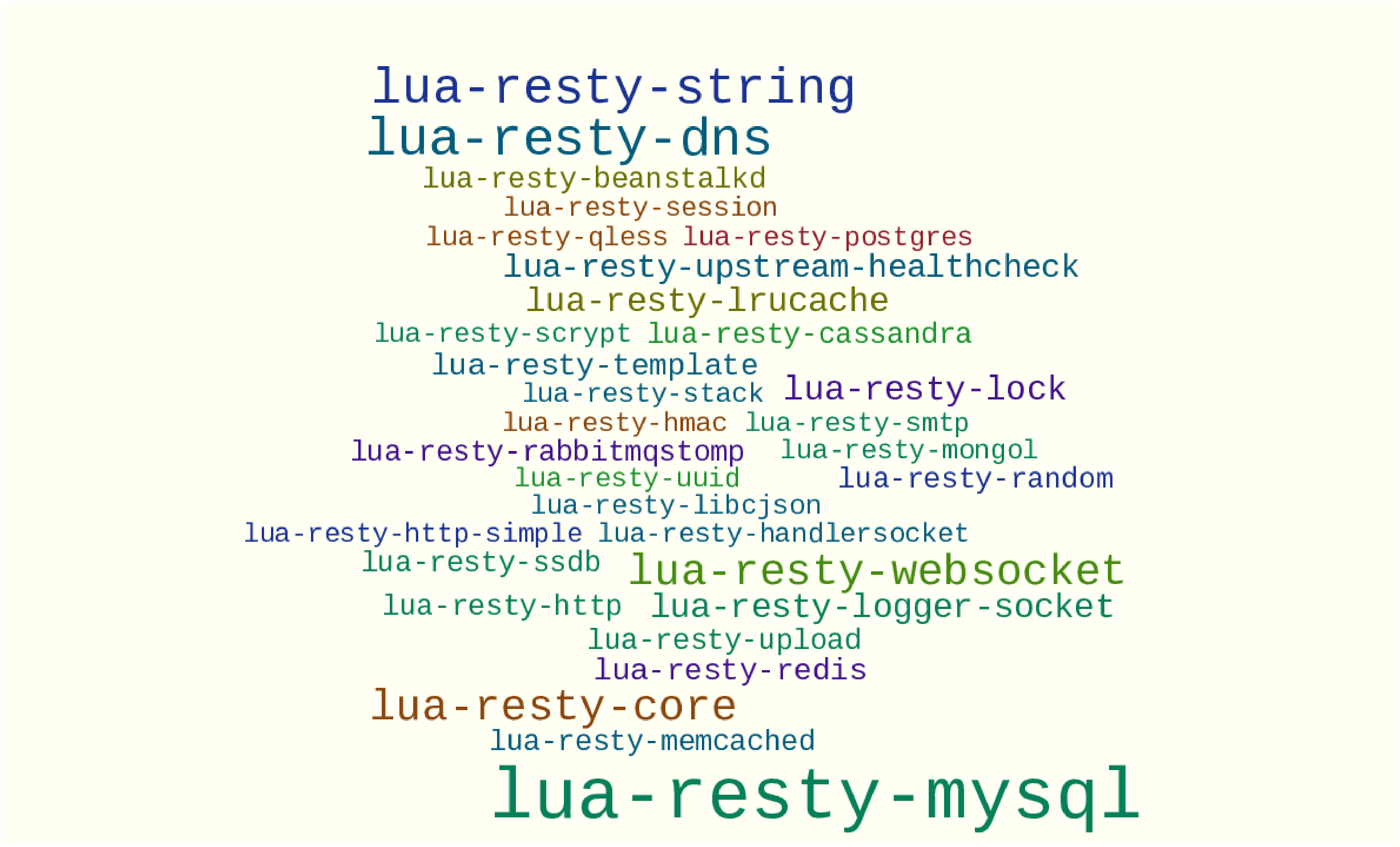
```
$ curl http://localhost/animals
<!-- file a.tt2 -->
<html><body>
<ul>
  <li>dogs</li>
  <li>cats</li>
  <li>birds</li>
</ul>
</body></html>
```

☺ Jemplate

TT2 templates ⇒ Browser JavaScript code

☺ Official Windows binary builds

Linux binary packages coming...


lua-resty-string
lua-resty-dns
lua-resty-beanstalkd
lua-resty-session
lua-resty-qless
lua-resty-postgres
lua-resty-upstream-healthcheck
lua-resty-lrucache
lua-resty-scrypt
lua-resty-cassandra
lua-resty-template
lua-resty-stack
lua-resty-lock
lua-resty-hmac
lua-resty-smtp
lua-resty-rabbitmqstomp
lua-resty-mongol
lua-resty-uuid
lua-resty-random
lua-resty-libcjson
lua-resty-http-simple
lua-resty-handlersocket
lua-resty-ssdb
lua-resty-websocket
lua-resty-http
lua-resty-logger-socket
lua-resty-upload
lua-resty-redis
lua-resty-core
lua-resty-memcached
lua-resty-mysql

- ☺ Official Package Management

resty.luarocks.org

```
    $ opm install agentzh/lua-resty-
scream
```

☺ OpenResty Edge Platform

```
server agentzh.org;

uri-prefix("/foo", "/bar/baz")
=>
    set-uri-arg(channel: 3),
    done;

uri(rx{ / ([0-9a-f]+) }), uri-arg("name") as $name
=>
    redirect(uri: "/en/$name/$1", code: 301);
```

☺ The Edge *optimizing* compiler targeting Lua

- ✓ PCRE JIT
- ✓ Sregex DFA
- ✓ Intel HyperScan
- ✓ Google RE2

☺ OpenResty WAF Platform

ModSecurity rule translator targeting OpenResty Edge

☺ The ORSQL language

Relational-relational mapping (RRM)

```
date        $day;
symbol      $database, $table, $column;
location    $db_node;

@res :=
    select $column, count(id)
    from $database.$table
    where day = $day
    group by $column
    at $db_node;
```

☺ The Y language

Y source code

- ⇛ SystemTap scripts
- ⇛ GDB Python scripts
- ⇛ LLDB Python scripts
- ⇛ Linux eBPF bytecode?

☺ *Any questions*? ☺