

Ingress Nginx

Subtitle

2022/09/30



Table of Contents

- Ingress Nginx 1
 - tengine http_reqstat* 1
 - ssl_protocol per server* 1
 - Other* 1
- Nginx ingress controller closes chunked encoding requests 1
 - 更正* 1
 - 调试nginx coredump* 2

Ingress Nginx

tengine http_reqstat

- 2.1.2及之后的tengine版本中，httpreqstat模块不能获取\$proxyhost, \$upstreamaddr等变量，参考：<https://github.com/alibaba/tengine/issues/983> * 在Nginx中编译httpreqstat 参考：https://github.com/x-v8/nginx_http_reqstat_module
- 寻找替代品 http://nginx.org/en/docs/http/nginx_http_api_module.html

ssl_protocol per server

- <https://github.com/openssl/openssl/issues/4301>
- <http://mailman.nginx.org/pipermail/nginx/2018-April/056034.html>
- <http://mailman.nginx.org/pipermail/nginx/2018-April/056036.html>

Other

- [Nginx模块开发](#)

Nginx ingress controller closes chunked encoding requests

给一个响应为 chunked encoding的接口做代理，随机返回：

* transfer closed with outstanding read data remaining

或者

* Empty reply from server

- <https://github.com/kubernetes/kubernetes/issues/54505>
- <https://stackoverflow.com/questions/46909621/nginx-chunked-encoding-response-randomly-truncated>

更正

设置configmap之后仍然无效，发现nginx worker一直在重启：

```
{"log":"2018/10/23 17:10:39 [alert] 42#42: worker process 30459 exited on signal 11\n","stream":"stderr","time":"2018-10-23T09:10:39.38505128Z"}
{"log":"2018/10/23 17:10:39 [alert] 42#42: worker process 31582 exited on signal 11\n","stream":"stderr","time":"2018-10-23T09:10:39.430871919Z"}
{"log":"2018/10/23 17:10:39 [alert] 42#42: worker process 31549 exited on signal 11\n","stream":"stderr","time":"2018-10-23T09:10:39.512565818Z"}
{"log":"2018/10/23 17:10:39 [alert] 42#42: worker process 31615 exited on signal 11\n","stream":"stderr","time":"2018-10-23T09:10:39.51808842Z"}
{"log":"2018/10/23 17:10:39 [alert] 42#42: worker process 31450 exited on signal 11\n","stream":"stderr","time":"2018-10-23T09:10:39.52472621Z"}
```

...

同时相同配置的intra集群和dev, beta集群nginx worker不会频繁重启。需要解决此问题

调试nginx coredump

- 设置working_directory为/tmp[][参考链接](#)
- gdb使用 <https://blog.csdn.net/sunxiaopengsun/article/details/72974548>

```
[New LWP 20370]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by `nginx: worker process'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00007f74814f015b in ngx_http_reqstat_log_handler (r=0x7f747ebc3050)
    at /tmp/build/nginx_http_reqstat-1.0/nginx_http_reqstat_module.c:879
879          ngx_http_request_received(r)
[Current thread is 1 (Thread 0x7f74813edbc0 (LWP 20342))]
(gdb)

(gdb) backtrace full
#0 0x00007f74814f015b in ngx_http_reqstat_log_handler (r=0x7f747ebc3050)
    at /tmp/build/nginx_http_reqstat-1.0/nginx_http_reqstat_module.c:879
    p = <optimized out>
    val = {len = 140731689803648, data = 0x7f747ebc3050 "HTTP"}
    value = 0x7f747ebc4990
    indicator = <optimized out>
    iv = <optimized out>
    i = 0
    j = <optimized out>
    k = <optimized out>
    status = <optimized out>
    utries = <optimized out>
    tp = <optimized out>
    ms = <optimized out>
    total_ms = <optimized out>
    shm_zone = 0x7f747ebc7d08
    z = 0x7f747ea46378
    ctx = 0x7f747ebc7b88
    rcf = 0x7f747ea5b2b8
    store = 0x7f747ebc4918
    fnode = 0x7f747aff3020
    fnode_store = 0x7f747ebc4988
    state = <optimized out>
    v = <optimized out>
#1 0x00007f74814731e0 in ngx_http_log_request (r=r@entry=0x7f747ebc3050) at
src/http/nginx_http_request.c:3566
    i = 2
    n = 4
---Type <return> to continue, or q <return> to quit---
    log_handler = 0x7f747ea82b58
```

```
    cmcf = <optimized out>
#2 0x00007f7481474eda in ngx_http_free_request (r=0x7f747ebc3050, rc=rc@entry=0) at
src/http/nginx_http_request.c:3513
    log = 0x7f747ebfc2b0
    pool = <optimized out>
    linger = {l_onoff = -1503584384, l_linger = 32766}
    cln = 0x0
    ctx = <optimized out>
    clcf = <optimized out>
#3 0x00007f74814ae0bf in ngx_http_v2_close_stream (stream=0x7f747ebc3f88, rc=rc@entry=0)
at src/http/v2/nginx_http_v2.c:4328
    pool = 0x7f747e837400
    push = 0
    ev = <optimized out>
    fc = 0x7f747ebfc100
    node = <optimized out>
    h2c = 0x7f747eaf8e20
#4 0x00007f7481475081 in ngx_http_close_request (r=r@entry=0x7f747ebc3050, rc=rc@entry=0)
at src/http/nginx_http_request.c:3455
    c = 0x7f747ebfc100
#5 0x00007f7481476492 in ngx_http_finalize_connection (r=r@entry=0x7f747ebc3050) at
src/http/nginx_http_request.c:2583
    clcf = 0x7f747ebfc100
#6 0x00007f7481476d71 in ngx_http_finalize_request (r=r@entry=0x7f747ebc3050,
rc=rc@entry=0) at src/http/nginx_http_request.c:2510
    c = 0x7f747ebfc100
    pr = <optimized out>
    clcf = <optimized out>
#7 0x00007f7481487aa3 in ngx_http_upstream_finalize_request (r=r@entry=0x7f747ebc3050,
u=u@entry=0x7f747ebc4aa8, rc=rc@entry=0)
at src/http/nginx_http_upstream.c:4404
    flush = <optimized out>
---Type <return> to continue, or q <return> to quit---
#8 0x00007f748148a97c in ngx_http_upstream_process_non_buffered_request
(r=r@entry=0x7f747ebc3050, do_write=<optimized out>,
do_write@entry=1) at src/http/nginx_http_upstream.c:3552
    size = <optimized out>
    n = <optimized out>
    b = 0x7f747ebc4de0
    rc = <optimized out>
    downstream = 0x7f747ebfc100
    upstream = 0x7f747b807950
    u = 0x7f747ebc4aa8
    clcf = <optimized out>
#9 0x00007f748148ae40 in ngx_http_upstream_process_non_buffered_downstream
(r=r@entry=0x7f747ebc3050)
at src/http/nginx_http_upstream.c:3484
    wev = 0x7f747ebfc250
    c = 0x7f747ebfc100
    u = 0x7f747ebc4aa8
#10 0x00007f748148bc74 in ngx_http_upstream_send_response (r=r@entry=0x7f747ebc3050,
```

```
u=u@entry=0x7f747ebc4aa8)
  at src/http/nginx_http_upstream.c:2956
    n = <optimized out>
    rc = <optimized out>
    p = <optimized out>
    c = 0x7f747ebfc100
    clcf = 0x7f747ea5c668
#11 0x00007f748148c8af in ngx_http_upstream_process_header (r=0x7f747ebc3050,
u=0x7f747ebc4aa8) at src/http/nginx_http_upstream.c:2391
  n = 164
  rc = <optimized out>
  c = 0x7f747b807950
#12 0x00007f7481487f75 in ngx_http_upstream_handler (ev=0x7f7478806a20) at
src/http/nginx_http_upstream.c:1280
---Type <return> to continue, or q <return> to quit---
  c = 0x7f747ebfc100
  r = 0x7f747ebc3050
  u = 0x7f747ebc4aa8
#13 0x00007f748145dafd in ngx_epoll_process_events (cycle=0x7f747ea46050, timer=<optimized
out>, flags=<optimized out>)
  at src/event/modules/nginx_epoll_module.c:902
    events = 1
    revents = <optimized out>
    instance = 0
    i = 0
    level = <optimized out>
    err = <optimized out>
    rev = 0x7f7478806a20
    wev = <optimized out>
    queue = <optimized out>
    c = 0x7f747b807950
#14 0x00007f7481451715 in ngx_process_events_and_timers
(cycle=cycle@entry=0x7f747ea46050) at src/event/nginx_event.c:242
  flags = 1
  timer = 2805
  delta = 2437952131
#15 0x00007f748145b239 in ngx_worker_process_cycle (cycle=cycle@entry=0x7f747ea46050,
data=data@entry=0xe)
  at src/os/unix/nginx_process_cycle.c:750
    worker = 14
#16 0x00007f7481458f54 in ngx_spawn_process (cycle=cycle@entry=0x7f747ea46050,
proc=0x7f748145b130 <ngx_worker_process_cycle>,
  data=0xe, name=0x7f7481592d0c "worker process", respawn=respawn@entry=56) at
src/os/unix/nginx_process.c:199
  on = 1
  pid = 0
  s = 56
---Type <return> to continue, or q <return> to quit---
#17 0x00007f748145a933 in ngx_reap_children (cycle=cycle@entry=0x7f747ea46050) at
src/os/unix/nginx_process_cycle.c:622
  i = 56
```



```

n = <optimized out>
live = 1
ch = {command = 2, pid = 19207, slot = 56, fd = -1}
ccf = <optimized out>
#18 0x00007f748145be2a in ngx_master_process_cycle (cycle=0x7f747ea46050) at
src/os/unix/nginx_process_cycle.c:175
title = <optimized out>
p = <optimized out>
size = <optimized out>
i = <optimized out>
n = <optimized out>
sigio = 0
set = {__val = {0 <repeats 16 times>}}
itv = {it_interval = {tv_sec = 0, tv_usec = 0}, it_value = {tv_sec = 0, tv_usec = 0}}
live = 1
delay = 0
ls = <optimized out>
ccf = 0x7f747ea47918
#19 0x00007f748143053f in main (argc=<optimized out>, argv=<optimized out>) at
src/core/nginx.c:382
b = <optimized out>
log = 0x7f748188bee0 <ngx_log>
i = <optimized out>
cycle = 0x7f747e8b7050
init_cycle = {conf_ctx = 0x0, pool = 0x7f747e833800, log = 0x7f748188bee0 <ngx_log>,
new_log = {log_level = 0, file = 0x0,
connection = 0, disk_full_time = 0, handler = 0x0, data = 0x0, writer = 0x0, wdata = 0x0,
action = 0x0, next = 0x0},
log_use_stderr = 0, files = 0x0, free_connections = 0x0, free_connection_n = 0, modules =
0x0, modules_n = 0,
---Type <return> to continue, or q <return> to quit---
modules_used = 0, reusable_connections_queue = {prev = 0x0, next = 0x0},
reusable_connections_n = 0, listening = {
elts = 0x0, nelts = 0, size = 0, nalloc = 0, pool = 0x0}, paths = {elts = 0x0, nelts = 0, size =
0, nalloc = 0,
pool = 0x0}, config_dump = {elts = 0x0, nelts = 0, size = 0, nalloc = 0, pool = 0x0},
config_dump_rbtrees = {root = 0x0,
sentinel = 0x0, insert = 0x0}, config_dump_sentinel = {key = 0, left = 0x0, right = 0x0,
parent = 0x0, color = 0 '\000',
data = 0 '\000'}, open_files = {last = 0x0, part = {elts = 0x0, nelts = 0, next = 0x0}, size =
0, nalloc = 0,
pool = 0x0}, shared_memory = {last = 0x0, part = {elts = 0x0, nelts = 0, next = 0x0}, size
= 0, nalloc = 0, pool = 0x0},
connection_n = 0, files_n = 0, connections = 0x0, read_events = 0x0, write_events = 0x0,
old_cycle = 0x0, conf_file = {
len = 21, data = 0x7ffea661397b "ss"}, conf_param = {len = 0, data = 0x0}, conf_prefix =
{len = 11,
data = 0x7ffea661397b "ss"}, prefix = {len = 17, data = 0x7f748158d5b6
"/usr/share/nginx/"}, lock_file = {len = 0,
data = 0x0}, hostname = {len = 0, data = 0x0}}
cd = <optimized out>

```

```
ccf = <optimized out>
```

是因为编译了 `ngx_http_reqstat` 模块导致的

查看堆栈

```
(gdb) bt
#0 0x00007f74814f015b in ngx_http_reqstat_log_handler (r=0x7f747ebc3050)
    at /tmp/build/nginx_http_reqstat-1.0/nginx_http_reqstat_module.c:879
#1 0x00007f74814731e0 in ngx_http_log_request (r=r@entry=0x7f747ebc3050) at
src/http/nginx_http_request.c:3566
#2 0x00007f7481474eda in ngx_http_free_request (r=0x7f747ebc3050, rc=rc@entry=0) at
src/http/nginx_http_request.c:3513
#3 0x00007f74814ae0bf in ngx_http_v2_close_stream (stream=0x7f747ebc3f88, rc=rc@entry=0)
at src/http/v2/nginx_http_v2.c:4328
#4 0x00007f7481475081 in ngx_http_close_request (r=r@entry=0x7f747ebc3050, rc=rc@entry=0)
at src/http/nginx_http_request.c:3455
#5 0x00007f7481476492 in ngx_http_finalize_connection (r=r@entry=0x7f747ebc3050) at
src/http/nginx_http_request.c:2583
#6 0x00007f7481476d71 in ngx_http_finalize_request (r=r@entry=0x7f747ebc3050,
rc=rc@entry=0) at src/http/nginx_http_request.c:2510
#7 0x00007f7481487aa3 in ngx_http_upstream_finalize_request (r=r@entry=0x7f747ebc3050,
u=u@entry=0x7f747ebc4aa8, rc=rc@entry=0)
    at src/http/nginx_http_upstream.c:4404
#8 0x00007f748148a97c in ngx_http_upstream_process_non_buffered_request
(r=r@entry=0x7f747ebc3050, do_write=<optimized out>,
do_write@entry=1) at src/http/nginx_http_upstream.c:3552
#9 0x00007f748148ae40 in ngx_http_upstream_process_non_buffered_downstream
(r=r@entry=0x7f747ebc3050)
    at src/http/nginx_http_upstream.c:3484
#10 0x00007f748148bc74 in ngx_http_upstream_send_response (r=r@entry=0x7f747ebc3050,
u=u@entry=0x7f747ebc4aa8)
    at src/http/nginx_http_upstream.c:2956
#11 0x00007f748148c8af in ngx_http_upstream_process_header (r=0x7f747ebc3050,
u=0x7f747ebc4aa8) at src/http/nginx_http_upstream.c:2391
#12 0x00007f7481487f75 in ngx_http_upstream_handler (ev=0x7f7478806a20) at
src/http/nginx_http_upstream.c:1280
#13 0x00007f748145dafd in ngx_epoll_process_events (cycle=0x7f747ea46050, timer=<optimized
out>, flags=<optimized out>)
    at src/event/modules/nginx_epoll_module.c:902
#14 0x00007f7481451715 in ngx_process_events_and_timers
(cycle=cycle@entry=0x7f747ea46050) at src/event/nginx_event.c:242
#15 0x00007f748145b239 in ngx_worker_process_cycle (cycle=cycle@entry=0x7f747ea46050,
data=data@entry=0xe)
    at src/os/unix/nginx_process_cycle.c:750
#16 0x00007f7481458f54 in ngx_spawn_process (cycle=cycle@entry=0x7f747ea46050,
proc=0x7f748145b130 <ngx_worker_process_cycle>,
data=0xe, name=0x7f7481592d0c "worker process", respawn=respawn@entry=56) at
src/os/unix/nginx_process.c:199
#17 0x00007f748145a933 in ngx_reap_children (cycle=cycle@entry=0x7f747ea46050) at
src/os/unix/nginx_process_cycle.c:622
```

```
#18 0x00007f748145be2a in ngx_master_process_cycle (cycle=0x7f747ea46050) at
src/os/unix/ngx_process_cycle.c:175
---Type <return> to continue, or q <return> to quit---
#19 0x00007f748143053f in main (argc=<optimized out>, argv=<optimized out>) at
src/core/nginx.c:382
```

Printed on: 2022/09/30 01:16

Convert to img Failed!