NGINX Conf 2018

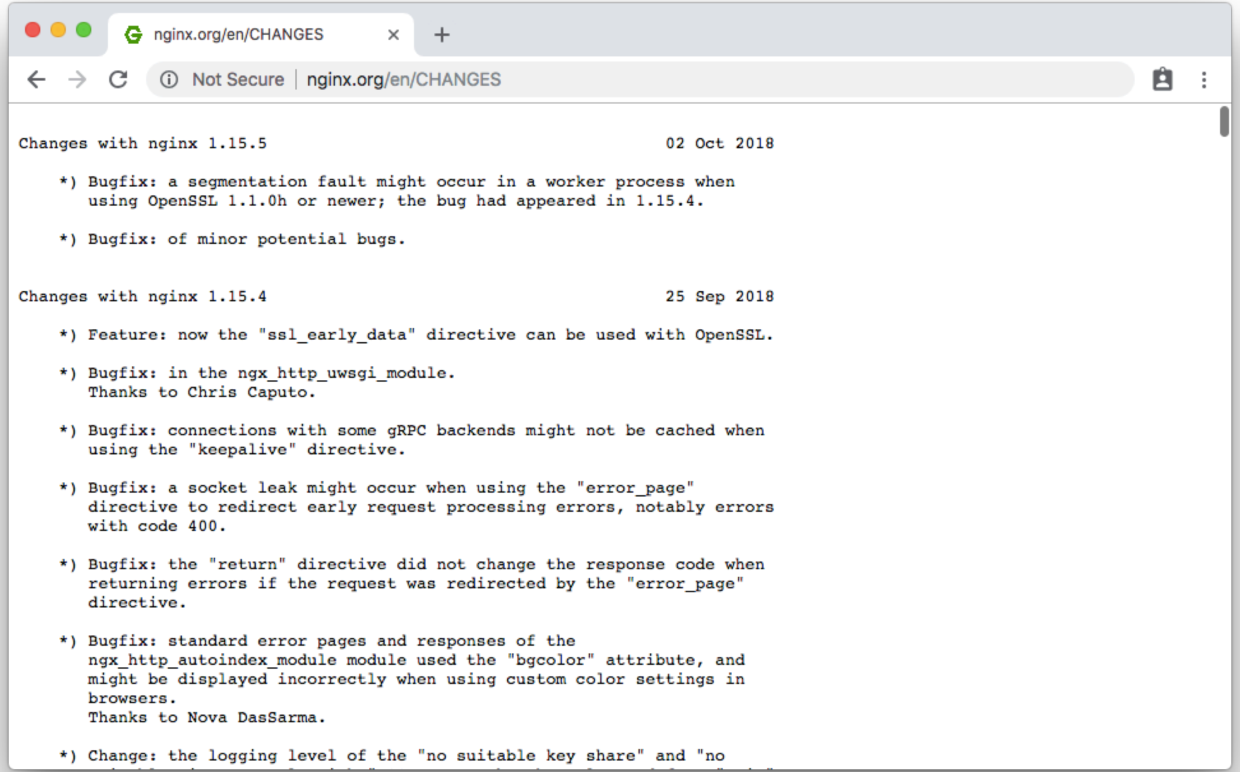
The official event for all things NGINX

Reading nginx CHANGES together

Maxim Dounin

NGINX

CHANGES



nginx.org/en/CHANGES

Not Secure | nginx.org/en/CHANGES

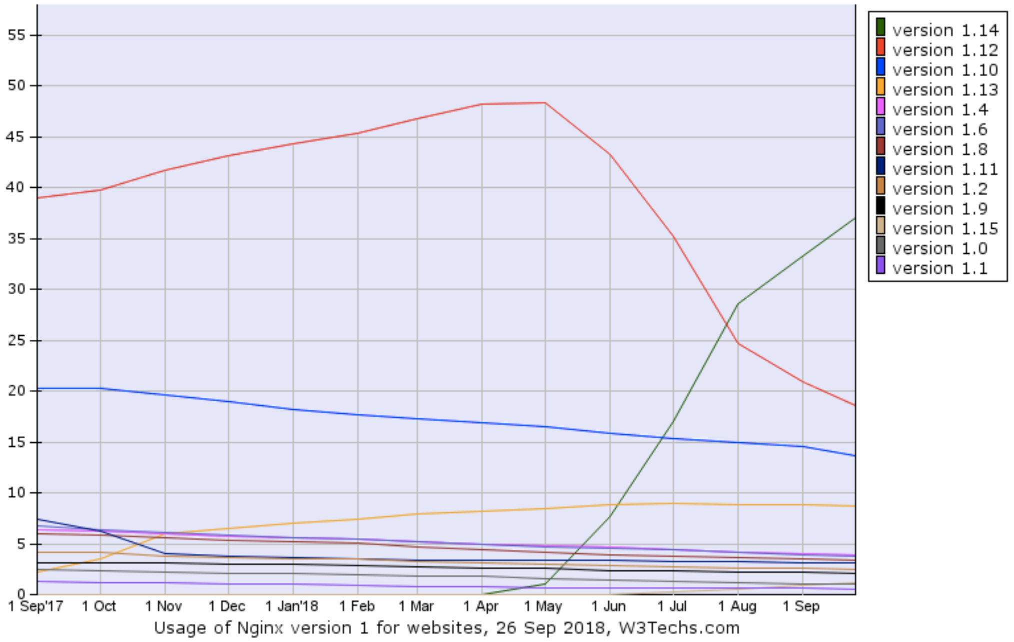
```
Changes with nginx 1.15.5                                        02 Oct 2018

    *) Bugfix: a segmentation fault might occur in a worker process when
       using OpenSSL 1.1.0h or newer; the bug had appeared in 1.15.4.

    *) Bugfix: of minor potential bugs.


Changes with nginx 1.15.4                                        25 Sep 2018

    *) Feature: now the "ssl_early_data" directive can be used with OpenSSL.

    *) Bugfix: in the ngx_http_uwsgi_module.
       Thanks to Chris Caputo.

    *) Bugfix: connections with some gRPC backends might not be cached when
       using the "keepalive" directive.

    *) Bugfix: a socket leak might occur when using the "error_page"
       directive to redirect early request processing errors, notably errors
       with code 400.

    *) Bugfix: the "return" directive did not change the response code when
       returning errors if the request was redirected by the "error_page"
       directive.

    *) Bugfix: standard error pages and responses of the
       ngx_http_autoindex_module module used the "bgcolor" attribute, and
       might be displayed incorrectly when using custom color settings in
       browsers.
       Thanks to Nova DasSarma.

    *) Change: the logging level of the "no suitable key share" and "no
```

nginx versions

- 1.11.x, 1.13.x, 1.15.x - mainline
  - Odd numbers
  - New features are developed here
  - Current version - **1.15.5**
- 1.12.x, 1.14.x - stable
  - Even numbers
  - New stable branch every year
  - Only critical fixes, stable API
  - Current stable version - **1.14.0**

Lies, damned lies, and statistics

```
1.14.x - 37.0%
1.12.x - 18.6%
1.10.x - 13.6%
1.13.x -  8.8%
...
1.15.x -  1.2%
```



- Changes in 1.13.x
  - Available in 1.14.0, latest stable version
- Changes in 1.15.x
  - Available in 1.15.5, latest mainline version

1.13.x

Basic TLS 1.3 support, gRPC proxy module, mirror module to enable traffic investigation, HTTP/2 push, monotonic timers, PROXY protocol version 2, and more.

Available in 1.14.0 stable.

TLS 1.3

- RFC 8446
    - Published in August 2018
- 1 RTT full handshake
    - Not guaranteed, but usually
    - Instead of 2 RTT in previous versions
- 0 RTT / early data
    - No reply protection
    - Needs special support - not yet in 1.13.x (but in 1.15.x)

TLS 1.3 basic support

```
server {
    listen 443 ssl;

    ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;

    ssl_certificate test.crt;
    ssl_certificate test.key;
}
```

- Not enabled by default
- Works with OpenSSL 1.1.1
- Only basic support (no early data in 1.13.x)

TLS 1.3 caveats

- Might not work with your browser
  - OpenSSL 1.1.1 implements RFC 8446
  - Chrome 69 - draft 28 or draft 23
  - Firefox 62 - draft 28
  - Safari on macOS High Sierra - draft 18, disabled by default
- Can be easily broken by incorrect configuration
  - ssl_ecdh_curve secp384r1;

Other SSL improvements

- Renegotiation with backend servers
  - Disabled due to CVE-2009-3555 - no longer relevant
  - Some backends require renegotiation
- The $ssl_client_escaped_cert variable
  - Simplifies passing the certificate to backends
- Now tcp_nodelay activated before SSL handshake
  - For TLS 1.3, triggers "Nagle vs. Delayed Ack" problem

Mirror

```
location / {
    mirror /mirror;
    proxy_pass http://real-backend;
}

location /mirror {
    proxy_pass http://mirror-backend;
    proxy_set_header X-Original-URI $request_uri;
}
```

Mirror: details

- Uses background subrequests
  - Introduced for proxy_cache_background_update, rewritten for mirror
- Subrequests are executed in parallel with main request
  - Slow subrequest can delay main request
- The request body is read by default
  - mirror_request_body off;

Mirror: development details

- Fixed an old problem with proxying subreqests with bodies
  - An optimization: request body file closed when response header is received
  - Caused problems with SSI and POST requests
  - Now switched off with subrequests
- New request processing phase: precontent
  - Used by try_files and mirror
  - Can be used for your own modules

HTTP/2 server push

- An HTTP/2 protocol feature
- May improve website latency when used properly
- But can make you site slower
- And it will in most cases
    - “Chrome's view on Push" by Brad Lassey, https://github.com/httpwg/wg-materials/blob/gh-pages/ietf102/chrome_push.pdf

HTTP/2 server push

How to:

```
http2_push /css/main.css;
```

Push "Link: rel=preload" on proxying:

```
http2_push_preload on;
```

Use with care

gRPC proxy

- Proxying and balancing gRPC backends
- Uses HTTP/2 but there are nuances
  - gRPC requires trailers support
- Designed specially for gRPC
  - No request buffering, no response buffering
- No multiplexing
- Persistent connections with upstream keepalive

gRPC proxy: example

```
server {
    listen 50051 http2;

    location / {
        grpc_pass 127.0.0.2:50051;
    }
}
```

gRPC proxy: keepalive

```
upstream backend {
    server 127.0.0.2:50051;
    server 127.0.0.3:50051;
    keepalive 10;
}

server {
    listen 50051 http2;

    location / {
        grpc_pass backend;
    }
}
```

Misc

- CPU affinity on DragonFly BSD
- Improved CPU cache line size detection
    - `sysconf(_SC_LEVEL1_DCACHE_LINESIZE)`
- Better compatibility with optimized zlib variants
- Socket buffers tuning in mail and stream modules

Misc 2

- Hostnames in set_real_ip_from
- Logging of PID of the process which sent the signal
- Support for 308 redirections in "return" and "error_page"
- Now nginx preserves CAP_NET_RAW on Linux
    - root not needed with "proxy_bind ... transparent;"
- $ssl_preread_alpn_protocols in the stream module

Misc 3

- Escaping can be disabled in access logs
  - `log_format … escape=none …`
- Arbitrary subrequests in memory
  - `<!–#include virtual="/file" set="one" -->`
  - Previously proxying only, now static files too

Misc 4

- Range requests from an empty file now return 200
    - Previously 416, but 200 is also valid and better for the slice module
- Monotonic timers
    - `clock_gettime(CLOCK_MONOTONIC)`
    - No more timeouts on system time changes
- PROXY protocol version 2
    - Amazon NLB

All these features where developed in 1.13.x branch.

Available in 1.14.x stable.

1.15.x

TLS 1.3, UDP sessions, random balancer, and more.
Things we are working on.

TLS 1.3

- Fixed backend session reuse
- Now works with BoringSSL
- Early data support

TLS 1.3 early data

How to use early data:

```
ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
ssl_early_data on;
```

- No replay protection
    - Not at all in BoringSSL
    - The one in OpenSSL breaks session reuse, so disabled
- The $ssl_early_data variable
    - Early-Data header, RFC 8470

SSL: better configuration checking

- Missing certificates for "listen ... ssl" now detected

```
server {
        listen 443 ssl default;

        # no ssl_certificate here
}
```

SSL: better configuration checking

- The "ssl" directive deprecated in favor of "listen ... ssl"

```
server {
    listen 80;
    listen 443;

    ssl on;

    ...
}
```

Stream: UDP sessions

- UDP proxying assumed only 1 packet from client
  - Did not work for complex UDP-based protocols, such as DTLS
- Now tries to lookup an existing session
  - Can handle DTLS
  - Much better speed when there are many packets
- Only works within a worker
  - Single worker or "listen ... reuseport"
- Now "listen ... reuseport" works on FreeBSD 12
  - SO_REUSEPORT_LB

Stream: $ssl_preread_protocol

```
stream {
    map $ssl_preread_protocol $u {
        ""                      127.0.0.1:8443;
        default                 127.0.0.1:22;
    }

    server {
        listen 443;
        proxy_pass $u;
        ssl_preread on;
    }
}
```

New balancer: random

```
upstream {
    random;
    server 192.0.2.1;
    server 192.0.2.2;
    server 192.0.2.3;
}
```

- Faster than round-robin with many backends
- The same quality with many frontends

New balancer: random two

```
upstream {
    random two;
    server 192.0.2.1;
    server 192.0.2.2;
    server 192.0.2.3;
}
```

- Two random choices, best of the two is used
- Almost least_conn, but faster

Misc

- Now "reset_timedout_connection" applies to "return 444"
    - Saves kernel memory and sockets
- Upstream keepalive limits
    - “keepalive_timeout” - prevents a race with connection close by a backend
    - “keepalive_requests" - ensures connection-specific allocations will be freed

All these features where introduced in 1.15.x branch.

More are being worked on now.

NGINX

Thank you!
Questions?

Maxim Dounin
mdounin@mdounin.ru