Nginx ❤ PHP

match made in heaven

Helgi Þormar Þorbjörnsson

FOSDEM 2013

Helgi

Co-founded **Orchestra.io**

Work at **EngineYard**

PEAR Developer

From **Iceland**

@h on Twitter

Nginx

Just a web server?

No! It's so much more!

- ✓ Web Server
- ✓ Proxy
- ✓ Cache
- ✓ Mail Proxy
- ✓ And more!

Important for tweaking

Always run **configtest** before doing anything!

Reload (HUP Signal)

Reload (HUP Signal)

- Reloads config
- Starts up new workers
- Old workers stop listening
- Finish up any work they have

Upgrade (USR2 Signal)

Upgrade (USR2 Signal)

- Live upgrade of Nginx executable
- Starts up a new Master
- Run in parallel
- Old Workers gracefully shutdown
- Old Master can be brought back

Location Blocks

Foundation of most things we will do

Most Common Block

```
location / {
    try_files $uri
              $uri/
              /index.php$is_args$args;
}
```

```
location /helgi/is/awesome {
    return 202;
}
```

Accepts Regex

Done by adding ~ in front of the regular expression

Accepts Regex

```
# deny access to all .dot-files
location ~ /\. {
    access_log off;
    log_not_found off;
    deny all;
}
```

Accepts Regex

```
# deny access to all backups
location ~ ~$ {
    access_log off;
    log_not_found off;
    deny all;
}
```

PHP + Nginx configs

```
[global]
pid = /var/run/php/fpm.pid
error_log = /var/log/php-fpm/error.log
log_level = error
daemonize = yes

[www]
listen = /var/run/php/www1.sock
user = www-data
group = www-data

pm = static
pm.max_children = 10
pm.max_requests = 500

request_slowlog_timeout = 30
slowlog = /var/log/php-fpm/slow.log
```

```
upstream web_workers {
    server unix:/var/run/php/www1.sock;
}

location / {
    try_files $uri $uri/ /index.php$is_args$args;
}

location ~ \.php$ {
    if (!-f $request_filename) {
        return 404;
    }

    fastcgi_pass  web_workers;
    fastcgi_index index.php;
    fastcgi_intercept_errors off;

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param SERVER_NAME $host;
}
```

Health Checks

Nginx Check

```
location = /health/ping {
    return 200 “ok”;
}
```

Nginx + FPM

PHP FPM Config

```
[healthcheck]
listen = /var/run/php/healthcheck.sock
user = www-data

pm = static
pm.max_children = 1
pm.max_requests = 10000

ping.path = /health/ping
ping.response = "pong"
```

Nginx + FPM

Nginx Config

```
upstream healthcheck {
    server unix:/var/run/php/healthcheck.sock;
}

location = /health/ping {
    fastcgi_pass healthcheck;
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME
                  $document_root$fastcgi_script_name;
}
```

Error Pages

Basic Custom 404

```
error_page 404 /errors/404.html;
```

Requires /errors/404.html to be
in the document root

Basic Custom 404

```
root /var/www/public;

error_page 404 /errors/404.html;
location /errors/404.html {
  internal;
  root /var/www;
}
```

Errors via PHP

```
error_page 404 @four_oh_four;
location @four_oh_four {
    if (!-f "/var/www/errors/404.php") {
        return 404;
    }

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME /var/www/errors/404.php;
    fastcgi_pass  web_workers;
}
```

50x and others

```
error_page 500 501 502 503 504 @five_oh_ex;
location @five_oh_ex {
    <Same as 404>
}
```

Most HTTP codes can be bunched together

Facebook

Official Solution

```
error_page 405 =200 $uri;
```

My Hack

```
error_page 405 =200 @four_oh_five;
location @four_oh_five {
    if (!-f $request_filename) {
        return 404;
    }

    proxy_method GET;
    proxy_pass http://localhost:80;
}
```

Rewrite Module

Regex (PCRE)

Responsible for all if statements, file exists checks, returns and more.

Can work with most Nginx variables such as `$http_cookie`, `$user_agent`, `$uri` and countless others.

last	Finish rewrite and evaluates all rewrites again
break	Finish rewrite does no further rewrite processing
redirect	Returns a 302 on the rewrite
permanent	Returns a 301 on the rewrite

Forward Domains

How to send www.helgi.ws to helgi.ws

Forward Domains

The Correct Way™

```
server {
    server_name www.helgi.ws;
    return 301 $scheme://helgi.ws$request_uri;
}
```

Headers

add_header

```
add_header Set-Cookie "_orchestra=1;
                       Max-Age=2;
                       Path=/";
```

expires

```
location ~* ^.+\.(jpg|js|jpeg|png)$ {
    expires 1h;
}
```

Introducing

wiki.nginx.org/NginxHttpHeadersMoreModule

```
more_set_headers 'Server: My-Temple';

# set and clear output headers
location /bar {
    more_set_headers 'X-MyHeader: blah' 'X-MyHeader2: foo';
    more_set_headers -t 'text/plain text/css'
                       'Content-Type: text/foo';

    more_set_headers -s '400 404 500 503' -s 413 'Foo: Bar';
    more_clear_headers 'Transfer-Encoding' 'Content-Type';
}

# set output headers
location /type {
    more_set_headers 'Content-Type: text/plain';
}
```

```
# set input headers
location /foo {
    more_set_input_headers 'Host: Brussels';
    more_set_input_headers -t 'text/plain' 'X-Fosdem: bah';
}

# replace input header X-Fosdem *only* if it already exists
more_set_input_headers -r 'X-Fosdem: howdy';
```

Load Balancing

Simple Round Robin

```
upstream web_workers {
   server    www1.example.com;
   server    www2.example.com;
   server    www3.example.com;
   server    www4.example.com;
}
```

Consistent IP Routing

```
upstream web_workers {
   ip_hash;
   server    www1.example.com;
   server    www2.example.com;
   server    www3.example.com;
   server    www4.example.com;
}
```

Different Weights

```
upstream web_workers {
   server    www1.example.com;
   server    www2.example.com
             weight=2 max_fails=2 fail_timeout=15;
   server    www3.example.com
             weight=4 max_fails=3;
   server    www4.example.com
             weight=4 max_fails=4 fail_timeout=20;
   keepalive 8;
}
```

weight and ip_hash can work
together in Nginx **1.3.1+**

Cache

```
http {
  proxy_set_header  X-Real-IP  $remote_addr;
  proxy_set_header  X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_set_header  Host $http_host;
  proxy_cache_path  /dev/shm/nginx levels=1:2
                                   keys_zone=my-cache:8m
                                   max_size=2g
                                   inactive=600m;
  proxy_temp_path /dev/shm/nginx/tmp;
  proxy_cache_use_stale updating;

  server {
    location / {
      proxy_pass http://example.net;
      proxy_cache my-cache;
      proxy_cache_valid  200 302  60m;
      proxy_cache_valid  404      1m;
    }
  }
}
```

Questions?

@h

helgi@engineyard.com

Please rate at joind.in/7083