The LAMP Just Died: Follow the Light

nginx.conf 2014 – San Francisco

Speaker: Bernard Rosset
bernard@rosset.net

About me: https://rosset.net/

- Graduated in 2012
- Deeply interested by High-Availability challenges
  Nginx, Redis (NoSQL), OpenStack, …

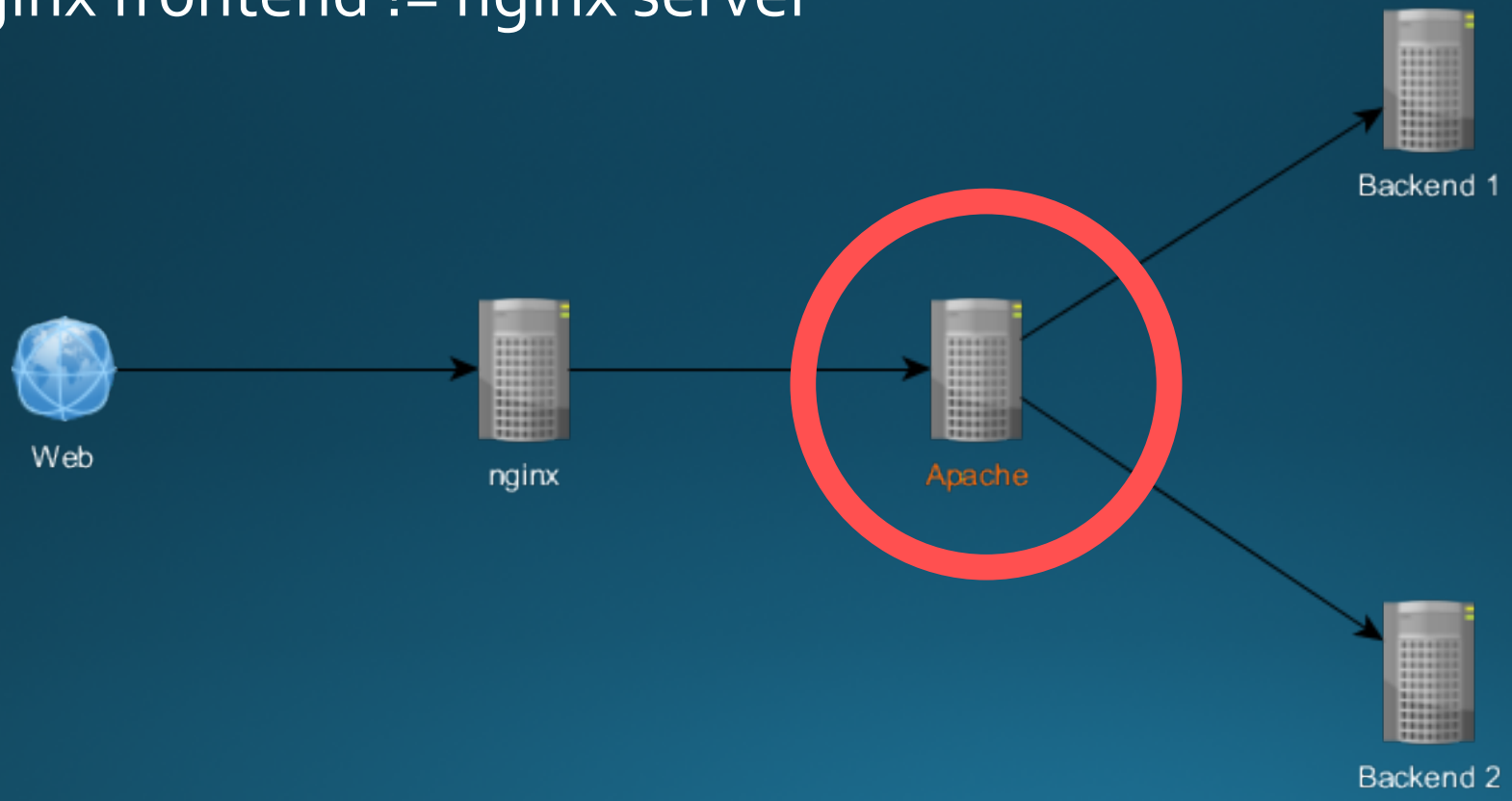
- Afficionado of nginx for 3 years
- Using PHP for 10+ years despite its inherent flaws

Step-by-step basic nginx + PHP-FPM setup

Basics

From Apache to nginx (1/2)

- nginx frontend != nginx server



From Apache to nginx (2/2)

- Apache's C10k (non-static data) nightmare:
  - mod_php (and mod_fcgid) must die: Apache spawned processes
  - mod_fastcgi… 3rd-party… not nominal

Some performance benchmarks:

https://coolpandaca.wordpress.com/2012/03/20/apache-mpm-worker-prefork-mod_php-mod_fcgid-mod_fastcgi-php-fpm-and-nginx/

- PHP + FastCGI = PHP-FPM: in PHP's core since v5.3.3!

Time to remove the extra layer!

Demo setup

- Linux Debian stable (Wheezy) v7.6
- Nginx FOSS v1.6.2 (nginx's official Debian package)
- PHP-FPM v5.6.1 (Dotdeb package)
  - v5.4 branch available through official Debian packages
  - v5.5 branch also available through Dotdeb

- Server content directory structure (Filesystem Hierarchy Standard)

```
/srv/
├── web   ← Document root (PHP scripts)
└── logs  ← Custom logs (later)
```

Installing nginx (official repository)

http://nginx.org/en/linux_packages.html

1. Add nginx repository

   ```
   deb http://nginx.org/packages/debian/ wheezy nginx
   ```

2. Download nginx repository PGP key

   ```
   wget http://nginx.org/keys/nginx_signing.key
   ```

3. Authenticate repository PGP key & add it

   http://nginx.org/en/pgp_keys.html

   Keys all stored at the same location = equal amount of trust… You just need to trust one!

   ```
   sudo apt-ket add nginx_signing.key
   ```

4. Update/Install

   ```
   /etc/nginx/
   ├── conf.d          ← conf.d/*.conf included from nginx.conf
   │   ├── default.conf          ← Server configuration example
   │   ├── example_ssl.conf      ← SSL example (commented out)
   ├── fastcgi_params  ← Standard FastCGI arguments for FastCGI client module
   ├── nginx.conf      ← Main configuration file
   ```

Configuring nginx

1. Change user/group (nginx/www-data)
   - `user` directive in `nginx.conf` OK
   - `nginx.conf` untouched + nginx user secondary group OK
   - `nginx.conf` untouched + nginx user primary group KO

     http://trac.nginx.org/nginx/ticket/165
2. Change workers number: `1` → `auto`
3. Remove server tokens
4. Activate GZip? Adjust logs?
5. Change server_name & clean-up default configuration
6. Adjust `root`

Installing PHP-FPM (Dotdeb)

http://www.dotdeb.org/instructions/

1. Add Dotdeb repository

```
deb http://packages.dotdeb.org wheezy-php56 all
```

2. Download Dotdeb repository PGP key

```
wget http://www.dotdeb.org/dotdeb.gpg
```

3. Add PGP key

```
sudo apt-key add dotdeb.gpg
```

4. Update/Install

```
/etc/php5/
├── conf.d
│   ├── …
├── fpm
│   ├── conf.d -> ../conf.d      ← Modules configuration, conf.d/*.conf included from php-fpm.conf
│   ├── php-fpm.conf                        ← Main FPM configuration file
│   ├── php.ini                  ← PHP interpreter configuration
│   └── pool.d                   ← Pool (listeners/environments) configuration, pool.d/*.conf included from php-
fpm.conf
│       └── www.conf
```

Configuring PHP-FPM (& nginx)

PHP

1. Add `date.timezone`

nginx

2. Add `index.php` to `index`
3. Add PHP `location`
    1. Include `fastcgi_params`
    2. Add `fastcgi_param: SCRIPT_FILENAME`
    3. Add `fastcgi_pass`
4. Basic nginx frontend + PHP backend working!

Full-fledged PHP environment

You want PATH_INFO set!

Easy!

1. Add `fastcgi_split_path_info ^(.+?\.php)(/.*)?$;`
2. Add `fastcgi_param` for `PATH_INFO`

Security hardening

http://example.org/php-logo-virus.jpg/non-existent.php

1. Why not 404?
2. Who denied?

Problem

Arbitrary code execution

PHP < v5.3.9 does not have the `security.limit_extensions` feature

https://nealpoole.com/blog/2011/04/setting-up-php-fastcgi-and-nginx-dont-trust-the-tutorials-check-your-configuration/ had it almost right:

Right problem… some wrong solutions!

Security hardening: the Dark side

Wrong solution

1. Use try_files

```
location ~ \.php {
        fastcgi_split_path_info ^(.+?\.php)(/.*)?$;

        try_files $fastcgi_scriptname =404;

        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
}
```

2. … use a workaround ← This is an early sign design is wrong

http://trac.nginx.org/nginx/ticket/321

Security hardening: the Light side

Right solution

1. Obvious: Exclude upload dirs from PHP processing. If under root:
   a) Add location matching the uploads sub-tree
   b) Ensure it has priority (location modifier precedence: http://nginx.org/r/location)

2. Set `cgi.fix_pathinfo=0` in `php.ini`
   https://php.net/cgi.fix_pathinfo

Special needs for special use cases

Advanced

Simple caching

- nginx fastcgi module integrates a caching system
- Dynamic + Cache = Static → High Availability

1. `fastcgi_cache_path` must be defined at http level
   Configures disk path for cached content + memory zone for index
2. `fastcgi_cache` activates the use of a cache zone in a block
3. `fastcgi_cache_valid` sets validity (freshness) based on status code

PHP pools

Pool = ‘forked’ environment

- Own listener (separated interpreter), workers, chroot possibility…
- May override `php.ini` values for local interpreter
- Own statistics
  - `pm.status` set to status path
  - Matching nginx `location`: beware the access rights!

https://php.net/manual/en/install.fpm.configuration.php

Logs

https://php.net/error_log

- By default, `error_log` not set → 'sent to SAPI error logger'
  - PHP-FPM: sent back through FastCGI to nginx → nginx error log
- `error_log` can be set
  - from `php.ini`: relative to `DOCUMENT_ROOT` (sent by nginx)
  - from `pools/*.conf`: relative to pool prefix (default /usr/)
- Be careful of log rights & location
  - Write: PHP shall be able to write them
  - Read: Avoid serving them as Web content…
- `catch_workers_output` was supposed to be useful…

  `"If not set, stdout and stderr will be redirected to /dev/null according to FastCGI specs."`

Listening to nginx – PHP-FPM

Sometimes things just go wrong…

Problem

- Empty page or raw file sent
- No (or not explicit enough) error message
- Is the problem coming from nginx or PHP-FPM?

Solution

- Listen to what each says to the other
  1. Configure nginx and PHP-FPM to talk on TCP sockets (use network stack)
  2. `tcpdump`
     a) `sudo tcpdump -l -w output.raw -i lo port 9000`
     b) `sudo tcpdump -Al -r output.raw > output.log`

FastCGI index for directories

You want to serve content from a location matching a directory

Problem

- `index` will do its job and provide the `index.php` file to the backend
- However backend will receive the matched location to process

➔ A directory is not a valid file for PHP to process…

Solution

- Use `fastcgi_index` to provide backend with the index filename

Simple PHP load-balancing

- Several PHP instances either local or distributed among your network
  Basically anywhere your nginx instance can connect to (LAN, VPN, …)
- Files served locally on each PHP instance
  You might wish to make sure content is consistent among locations…

1. `upstream` block at `http` level
2. 1 `server` directive per… server
3. Domain names are automatically resolved (1 server / IP address)!
4. Change `fastcgi_pass` to point at the `upstream` group name
5. Play with weights, max fails, backup flag…

Merci

спасибо

धन्यवाद

Danke schön

Thank you (Australian accent)

http://goo.gl/forms/P5gcuWtKUx

https://rosset.net/LAMP_just_died.pptx

Thank you!