Chapter 9: Case Studies

Deploying a WordPress site

System requirements

- PHP
- MySQL

PHP Configuration

- set cgi.fix_pathinfo to 0.
- post_max_size: increase if necessary
- upload_max_filesize: upload if necessary
- date.timezone: find the proper value on php.net/manual/en/timezones.php

PHP-FPM side

- php-fpm.conf does not require immediate changes
  - create a pool declaring `[wordpress]`

```
; specify user account and group for the pool
; assume created wordpress user and group
user=wordpress
group=wordpress
; Network interface and listening port
; user 127.0.0.1 if nginx runs on the same machine
listen=127.0.0.1:9000
; only allow connections from local computer
; Change this value if Nginx runs on a different machine
allowed_clients=127.0.0.1
```

Optionally enable chrooting: specify a root directroy for the PHP processes of this pool. For example if you set the chroot to /home/wordpress/www PHP scripts will only be able to read the files and directories within the specified path. Anyt attemp to read or write a file or directroy outside /home/wordpress/www will fail. Highly recommended for security. Attackers must only be able to expoloit files within the reach of PHP process, the rest of the server wohuld not to be compromised.

```
chroot=/home/wordpress/www;
```

My SQL Config

```
mysql -u root -p
mysql> CREATE DATABASE wordpress;
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'localhost'
IDENTIFIED BY 'password';
mysql> exit
# mysql -u wordpress -p
mysql> SHOW DATABASES;
```

Downloading and Extracting WordPress

- wordpress.org/

Ngix Config

HTTP Blocks

Open configuration file nginx.conf and insert update on the following directives

```
# Sets the user and group under which the worker processes will run
# The following values are valid assuming the server will only be hosting
one website
user wordpress wordpress;
pid /var/run/nginx.pid;

events {
    # Edit this value depending on the server hardware
    worker_connections 768;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    # Default Nginx values
    keepalive_timeout 65;
    types_hash_max_size 2048;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    # Set access and error log paths
    access_log /var/log/nginx/acess.log;
    error_log /var/log/nginx/error.log;

    # Enable gzipping of files matching the given mime types
    gzip on;
    gzip_disable "msie6";
    gzip_types text/plain text/css application/json application/x-
javascript text/xml application/xml application/xml+rss text/javascript;

    # Inclunde virtual host configuration files;
    # Edit path accordingly
    include /etc/nginx/sites-enabled/*;
}
```

Server Block

Create a new file in the directory specified prevously. For example, create a file called wordpress.conf in the /etc/nginx/sites-enabled/ folder. Define virtual host configuration by inserting or updating the following directives

```
server {
    # Listen all network interfaces on port 80
    listen 80;

    # Specify the host name(s) that will match the site
    # The following value allows both www. and no subdomain
    server_name .example.com;

    # Set the path of the WordPress files
    root /home/wordpress/www;

    # Load index.php
    index index.php

    client_body_in_file_only clean;
    client_body_buffer_size 32K;

    client_max_body_size 300M;

    send_timeout 10s;

    # applies to static files:
    location ~* ^.+/.(jpg|jpeg|png|gif|ico|css|js)$ {
        access_log off; # disable logging
        # allow client browsers to cache files for long period
        expires 180d;
    }

    # Applies to every request
    location / {
        # Try servingt the requestd URI:
        # If the file does not exist, append /
        # If directory does not exist, redirect
            # to /index.php forwarding the request URI
        try_files $uri $uri/ /index.php?q=$uri&$args;
    }

    # Applies to every PHP file
    location ~ /.php$ {
        # Ensure file really exists
        if (!-e $request_filename) {
            return 404;
        }
        # Pass the request to your PHP-FPM backend
        fastcgi_pass 127.0.0.1:9000
        fastcgi_index index.php;
        fastcgi_param PATH_INFO $fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Self Signed Certificates

- Self-signed certificates can be generated by yourselff on your own server
- Certificated signed by a trusted certificate authority offer an additional level of security: a tihird party ascertains the authenticity of the server to the visitors

For testing env or websites for a restrictued amount of visitors, self signed certificates can be an option

```
apt-get install openssl
openssl genrsa --out example.com.key 2048
openssl req -new -key example.com.key example.com.csr
```

Enabling HTTPS in NGINX Configurtion

```
cat your_site_certificate.crt certificate_authority.crt > example.com.crt
```

1. Site certificate
2. CA certificate

```
listen 443 default_server ssl;

# Generated certificated file
ssl_certificate /etc/ssl/private/example.com.crt
# Private key file generated
ssl_certificate_key /etc/ssl/private/example.com.key;

ssl_session_cache shared:SSL:20m;
ssl_session_timeout 60m;

# Disable SSL in favor of TLS
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

Own Cloud Drive

SImilar Dropbox services that allow to store files onlijne and retrieve them easily from all sorts of devices, including mobile phones and tablets

1. Install PHP
2. Install MySQL
3. Configure PHP taking care of directives regarding maximum file upload size.
4. Create PHP-FPM pool dedicated to ownCloud.
5. Set up a SQL database and user.

- www.owncloud.org