Installation de Nginx

Compiler Nginx

- Passer en admin : '' su - admin ''
- Se rendre dans le dossier des scripts : '' cd scripts/nginx/ ''
- Lancer le script : '' sh nginx_1.6.sh ''

Configurer Nginx

- Créer le dossier de cache de nginx :
  - '' mkdir /var/cache/nginx ''
  - '' chown www-data:www-data /var/cache/nginx ''
- Préparer le dossier de config :
  - '' cd /etc/nginx ''
  - '' mkdir sites-available '' : contient les fichiers de config pour chaque domaine virtuel
  - '' mkdir sites-enabled '' : permet à l'aide de lien d'activer ou pas un site sans supprimer son fichier de config
- Créer un espace “localhost” :
  - '' mkdir /home/admin/www/yilgarn ''
  - '' chown admin:admin /home/admin/www/yilgarn ''
  - Ajouter le fichier “yilgarn.clapas.ovh.conf” décrit ci-dessous au dossier /etc/nginx/sites-available/
  - Créer un lien dans le dossiers /etc/nginx/sites-enabled/ : '' ln -s /etc/nginx/sites-available/yilgarn.clapas.ovh.conf /etc/nginx/sites-enabled/yilgarn.clapas.ovh.conf''
- Adapter le fichier nginx.conf avec le contenu indiqué ci-dessous : '' vi /etc/nginx/nginx.conf ''
- Démarrer le serveur : '' service nginx start ''

Config générale de nginx

Fichier /etc/nginx/nginx.conf :

```
user  www-data www-data; # Default: nobody
worker_processes  4; # Default : 1
# Nombre de fichiers ouverts par processeur
worker_rlimit_nofile 100000;

pid /var/run/nginx.pid;

# [ debug | info | notice | warn | error | crit ]
error_log  /var/log/nginx-error.log  info;

events {
    worker_connections  1024; # Default : 1024
    # use [ kqueue | rtsig | epoll | /dev/poll | select | poll ] ;
```

```
    # use kqueue;
    use epoll;
}

http {
    include  mime.types;
    # include  /etc/nginx/fastcgi.conf;
    index    index.html index.htm index.php;

    default_type  application/octet-stream;

    log_format main   '"$remote_addr" $host [$time_local] '
                              '"$request" $status $body_bytes_sent '
                              '$request_length $bytes_sent "$http_referer"
'
                              '"$http_user_agent" $request_time
"$gzip_ratio"';

    # La ligne ci-dessous semble inutile puisqu'il faut la redéfinir dans
chaque vhost : server {}
    access_log  /var/log/nginx-access.log  main;

    sendfile       on;
    tcp_nopush     on;

    # 1er param : timeout pour les connections keep-alive ; 2ème param :
valeur du header "Keep-Alive: timeout=" envoyé en réponse
    keepalive_timeout 30 30;
    # délai d'attente de la réponse de php-fpm : à accorder avec
request_terminate_timeout dans la conf de php-fpm
    fastcgi_read_timeout 3600s;

    gzip on;
    gzip_comp_level 9; # 1 <= level <= 9
    gzip_min_length 50; # les fichiers plus petits que 50 octets ne sont pas
compressés
    # types de fichiers à compresser
    gzip_types
        text/css
        text/plain
        text/javascript
        application/javascript
        application/json
        application/x-javascript
        application/xml
        application/xml+rss
        application/xhtml+xml
        application/x-font-ttf
        application/x-font-opentype
        application/vnd.ms-fontobject
        image/svg+xml
```

```
        image/x-icon
        application/rss+xml
        application/atom_xml;

    # Load all vhosts !
    include /etc/nginx/sites-enabled/*.conf;
}
```

Config du domaine d'administration

- Créer les dossiers suivant :
  - '' mkdir /home/admin/_logs ''
  - '' mkdir /home/admin/_sessions ''
- Exemple de fichier de config pour le domaine hébergeant le statut de Nginx.
  - Domaine // yilgarn.clapas.ovh //, fichier '' /etc/nginx/site-available/yilgarn.clapas.ovh.conf '' :

```
server {
    server_name yilgarn.clapas.ovh www.yilgarn.clapas.ovh;
    root "/home/admin/www/yilgarn";

    index index.php index.html index.htm;
    client_max_body_size 10m;

    access_log /home/admin/_logs/access.log;
    access_log /var/log/nginx-access.log main;
    error_log /home/admin/_logs/error.log;

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''#
    # BLOCAGE - Bloquer l'accès à certains robots et extracteurs
de contenu

    # Bloquer l'accès à certains robots
    if ($http_user_agent ~*
(Baiduspider|webalta|nikto|wkito|pikto|scan|acunetix|morfeus|webco
llage|youdao) ) {
        return 401;
    }
    # Bloquer l'accès aux extracteurs de contenu
    if ($http_user_agent ~*
(HTTrack|clshttp|archiver|loader|email|harvest|extract|grab|miner)
) {
        return 401;
    }

    # PROTECTION - Éviter l'affichage de fichiers sensibles

    # Éviter l'affichage du contenu des dossiers des gestionnaires
```

```
de versions
    location ~ "\.(svn|git|hg|bzr|cvs)" {
        return 404;
    }

    # Éviter l'affichage des fichiers de config en .ini
    location ~ "\.ini$" {
        return 404;
    }

    # Éviter l'affichage des fichiers sensibles
    location ~ "\.(htaccess|htpasswd)$" {
        return 404;
    }

    # AUTHENTIFICATION HTTP - Zones à accès restreint par mot de
passe

    # Fichiers php de gestion : OpCache, Log, Info PHP...
    location ~ "/(info|log|op).php" {
        auth_basic "Zone restreinte. Indiquez un login et mot de
passe.";
        auth_basic_user_file /home/admin/www/.htpasswd;

        include /etc/nginx/fastcgi.conf;
        fastcgi_pass   unix:/var/run/admin_fpm.sock;
    }

    # STATUTS

    location /nginx_status {
        stub_status on;
        access_log off;
        allow 127.0.0.1;
        allow 82.239.139.74;
        allow 193.54.123.216;
        allow 162.38.234.0/24;
        deny all;
    }

    # Statuts PHP-FPM 5.6
    # ATTENTION : pour voir ces infos, le fichier php-56-fpm.sock
doit avoir un droit de lecture pour l'utilisateur "admin"
    location ~ ^/php-56-fpm-(status|ping)$ {
        access_log off;
        include fastcgi_params;
        fastcgi_pass   unix:/var/run/php-56-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        allow 127.0.0.1;
        allow 82.239.139.74;
        deny all;
```

```
    }

    # Statuts PHP-FPM 5.5
    # ATTENTION : pour voir ces infos, le fichier php-55-fpm.sock
doit avoir un droit de lecture pour l'utilisateur "admin"
    location ~ ^/php-55-fpm-(status|ping)$ {
        access_log off;
        include fastcgi_params;
        fastcgi_pass   unix:/var/run/php-55-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        allow 127.0.0.1;
        allow 82.239.139.74;
        deny all;
    }

    # Statuts PHP-FPM 5.4
    # ATTENTION : pour voir ces infos, le fichier php-54-fpm.sock
doit avoir un droit de lecture pour l'utilisateur "admin"
    location ~ ^/php-54-fpm-(status|ping)$ {
        access_log off;
        include fastcgi_params;
        fastcgi_pass   unix:/var/run/php-54-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        allow 127.0.0.1;
        allow 82.239.139.74;
        deny all;
    }

    # Réecritures générales

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

    location ~ "^(.+\.php)($|/)" {
        fastcgi_split_path_info ^(.+\.php)(.*)$;

        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SERVER_NAME $host;

        if ($uri !~ "^/uploads/") {
            fastcgi_pass   unix:/var/run/admin_fpm.sock;
        }
        include fastcgi_params;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
```

```
        log_not_found off;
        access_log off;
    }

    location ~* \.(html|htm)$ {
        expires 30m;
    }
}
```

Exemple de config pour le sous-domaine sql.clapas.ovh

Domaine // sql.clapas.ovh //, fichier '' /etc/nginx/site-available/sql.clapas.ovh.conf '' :

```
server {
    server_name sql.clapas.ovh www.sql.clapas.ovh;
    root "/home/admin/www/phpmyadmin";

    index index.php index.html index.htm;
    client_max_body_size 10m;

    access_log /home/admin/_logs/access.log;
    access_log /var/log/nginx-access.log main;
    error_log /home/admin/_logs/error.log;

    # BLOCAGE - Bloquer l'accès à certains robots et extracteurs de contenu

    # Bloquer l'accès à certains robots
    if ($http_user_agent ~*
(Baiduspider|webalta|nikto|wkito|pikto|scan|acunetix|morfeus|webcollage|youd
ao) ) {
        return 401;
    }
    # Bloquer l'accès aux extracteurs de contenu
    if ($http_user_agent ~*
(HTTrack|clshttp|archiver|loader|email|harvest|extract|grab|miner) ) {
        return 401;
    }

    # PROTECTION - Éviter l'affichage de fichiers sensibles

    # Éviter l'affichage du contenu des dossiers des gestionnaires de
versions
    location ~ "\.(svn|git|hg|bzr|cvs)" {
        return 404;
    }

    # Éviter l'affichage des fichiers de config en .ini
    location ~ "\.ini$" {
```

```
        return 404;
    }

    # Éviter l'affichage des fichiers sensibles
    location ~ "\.(htaccess|htpasswd)$" {
        return 404;
    }

    # Réecritures générales

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

    location ~ "^(.+\.php)($|/)" {
        fastcgi_split_path_info ^(.+\.php)(.*)$;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SERVER_NAME $host;

        if ($uri !~ "^/uploads/") {
            fastcgi_pass   unix:/var/run/admin_fpm.sock;
        }
        include fastcgi_params;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
        access_log off;
    }

    location ~* \.(html|htm)$ {
        expires 30m;
    }
}
```

Exemple de config d'un domaine fournissant une api

Fichier '' /etc/nginx/site-available/api.bota.ovh.conf '' (s'en inspirer pour api-test.bota.ovh , en ajustant le server_name, le root, les logs etc.) :

```
server {
    server_name api.bota.ovh www.api.bota.ovh;
    root "/home/apibota/www";

    index index.php index.html index.htm;
```

```
    client_max_body_size 50m;

    access_log /home/apibota/_logs/access.log;
    access_log /var/log/nginx-access.log main;
    error_log /home/apibota/_logs/error.log;

    if ($http_user_agent ~*
(Baiduspider|webalta|nikto|wkito|pikto|scan|acunetix|morfeus|webcollage|youd
ao) ) {
        return 401;
    }

    if ($http_user_agent ~*
(HTTrack|clshttp|archiver|loader|email|harvest|extract|grab|miner) ) {
        return 401;
    }

    # Éviter l'affichage du contenu des dossiers des gestionnes de versions
    location ~ "\.(svn|git|hg|bzr|cvs)" {
        return 404;
    }

    # Eviter l'affichage des fichiers de config en .ini
    location ~ "\.ini$" {
        return 404;
    }

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

    # CORS dynamique multi-domaines
    set $cors_origine_acceptee "http://www.bota.ovh";
    if ($http_origin ~* (www\.bota\.ovh|localhost)) {
        set $cors_origine_acceptee "$http_origin";
    }

    # Entêtes pour CORS
    add_header Access-Control-Allow-Origin "$cors_origine_acceptee"; # mode
dynamique
    add_header Access-Control-Allow-Methods "GET, POST, PUT, DELETE,
OPTIONS";
    add_header Access-Control-Allow-Credentials "true";
    add_header Access-Control-Expose-Headers "X-DebugJrest-Data";

    # redéfinition par défaut de la query_string
    set $query_string_api $args;

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

```
        include /etc/nginx/fastcgi.conf;
        fastcgi_pass   unix:/var/run/apibota_fpm.sock;

        # Réecriture des headers pour l'API d'eFlore
        fastcgi_param REQUEST_URI $request_uri_api;
        fastcgi_param QUERY_STRING $query_string_api;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
        access_log off;
    }

    location ~* \.(html|htm)$ {
        expires 30m;
    }
}
```

Installation du script de création des domaines virtuels

- '' cd /home/admin/script/nginx/ ''
- `wget http://www.sebdangerfield.me.uk/wp-content/uploads/2012/05/create_php_vhost.tar.gz`
- '' tar xzvf create_php_vhost.tar.gz ''
- '' mv create_php_vhost.tar.gz create_php_vhost/ ''
- '' cd create_php_vhost/ ''
- Créer un fichier de test : '' vi test.php ''
    - Contenu : '' <pre><?php var_export($_SERVER)?></pre> ''
- Modifier le script général : '' vi create_php_site.sh ''
    - Remplacer le nom de dossier “public_html” par “www” dans le script (lignes 51, 53 et 55)
    - Remplacer la ligne 50 par :

```
 if [ "$CHANGEROOT" = "y" ]; then
```

    - Après la ligne 98 ajouter :

```
# Copy the test.php file''
cp $CURRENT_DIR/test.php /home/$HOME_DIR$PUBLIC_HTML_DIR/
echo "Test installation with : http://$DOMAIN/test.php"
```

- Créer un script spécifique à une version de php :
    - '' cp create_php_site.sh create_php56_site.sh ''
    - '' vi create_php56_site.sh ''
        - '' PHP_INI_DIR='/usr/local/php/5.6/etc/fpm.d/' ''
        - '' PHP_FPM_INIT='/etc/init.d/php-56-fpm' ''
- '' cp nginx.vhost.conf.template nginx.vhost.conf.template.default ''
- '' cp pool.conf.template pool.conf.template.default ''
- Changer les droits de tous les fichiers : '' chown admin:users ./* ''

Remplacer le contenu de nginx.vhost.conf.template par :

```
server {
    server_name @@HOSTNAME@@ www.@@HOSTNAME@@;
    root "@@PATH@@";

    index index.php;
    client_max_body_size 10m;

    access_log @@LOG_PATH@@/access.log;
    access_log /var/log/nginx-access.log main;
    error_log @@LOG_PATH@@/error.log;

    if ($http_user_agent ~*
(Baiduspider|webalta|nikto|wkito|pikto|scan|acunetix|morfeus|webcollage|youd
ao) ) {
        return 401;
    }

    if ($http_user_agent ~*
(HTTrack|clshttp|archiver|loader|email|harvest|extract|grab|miner) ) {
        return 401;
    }

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

    location ~ "^(.+\.php)($|/)" {
        fastcgi_split_path_info ^(.+\.php)(.*)$;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SERVER_NAME $host;

        if ($uri !~ "^/uploads/") {
            fastcgi_pass   unix:@@SOCKET@@;
        }
        include        fastcgi_params;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
        access_log off;
    }

    location ~* \.(html|htm)$ {
        expires 30m;
    }
```

```
    location ~* /\.(ht|git|svn) {
        deny  all;
    }
}
```

Remplacer le contenu de pool.conf.template par :

```
; Doc sur les paramètres :
http://www.php.net/manual/fr/install.fpm.configuration.php
[@@USER@@]
listen = /var/run/@@USER@@_fpm.sock
listen.owner = @@USER@@
listen.group = @@USER@@
listen.mode = 0660
user = @@USER@@
group = @@USER@@
pm = dynamic
pm.max_children = @@MAX_CHILDS@@
pm.start_servers = @@START_SERVERS@@
pm.min_spare_servers = @@MIN_SERVERS@@
pm.max_spare_servers = @@MAX_SERVERS@@
pm.max_requests = 500
request_terminate_timeout = 30s
chdir = /
php_admin_value[session.save_path] = "@@HOME_DIR@@/_sessions"
php_admin_value[open_basedir] =
"@@HOME_DIR@@:/usr/share/pear:/usr/share/php:/tmp:/usr/local/lib/php"
```

Activer Speedy sur Nginx

- Ressource : http://nginx.org/en/docs/http/ngx_http_spdy_module.html
- Pour activer Speedy, il faut :
  - compiler Nginx avec le support de Speedy (`--with-http_spdy_module`)
  - avoir OpenSSL installé en version 1.0.1 minimum
- Modifier ensuite dans le fichier de conf du domaine où l'on veut activer Speedy, avec les infos suivante :

```
server {
    listen 443 ssl spdy;

    ssl_certificate server.crt;
    ssl_certificate_key server.key;
    ...
}
```