eBook Gratuit

APPRENEZ nginx

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#nginx

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: nginx

It is an unofficial and free nginx ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nginx.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec nginx

Remarques

NGINX se prononce comme "engine x" et est couramment utilisé comme serveur hautes performances pour les protocoles HTTP, HTTPS, SMTP, POP3 et IMAP. Il peut être utilisé en tant que serveur proxy inverse, cache HTTP ou équilibrage de charge.

C'est un projet open source avec source disponible ici .

Versions

Version	Date de sortie originale	Dernière version	Statut	Date de sortie
0.5	2006-12-04	0.5.38	Héritage	2009-09-14
0,6	2007-06-14	0.6.39	Héritage	2009-09-14
0.7	2008-05-19	0.7.69	Héritage	2011-07-19
0,8	2009-06-02	0.8.55	Héritage	2011-07-19
1.0	2011-04-12	1.0.15	Héritage	2012-04-12
1.2	2012-04-23	1.2.9	Héritage	2013-05-13
1.4	2013-04-24	1.4.7	Héritage	2014-03-18
1.6	2014-04-24	1.6.3	Héritage	2015-04-07
1.8	2015-04-21	1.8.1	Héritage	2016-01-26
1,9	2015-04-28	1.9.15	Héritage	2016-04-19
1.10	2016-04-26	1.10.3	Stable	2016-05-31
1.11	2016-05-24	1.11.9	Mainline	2016-07-26

Examples

Installation et configuration

Nginx est un serveur Web utilisé pour servir les requêtes HTTP sur Internet.

Nginx est disponible sous Linux, Windows et autres systèmes d'exploitation en téléchargement direct et peut également être créé à partir de sources. Pour des instructions détaillées, voir la

référence officielle Nginx.

Ubuntu / Debian

La version stable de nginx est disponible en repo officiel, elle peut être installée en utilisant

```
sudo apt-get install nginx
```

Il va installer et configurer les fichiers de démarrage du système, mais si vous avez besoin de la dernière version, vous devrez peut-être ajouter un fichier ppa officiel.

```
sudo add-apt-repository ppa:nginx/stable
sudo apt-get update
sudo apt-get install nginx
```

les instructions ci-dessus installeront la dernière édition stable.

Installation de Nginx sur les distributions Debian et Debian comme Ubuntu

Exécutez la commande ci-dessous pour installer nginx.

```
sudo apt-get install nginx
```

Par défaut, Nginx démarre automatiquement lorsqu'il est installé. Vous pouvez accéder à la page d'accueil par défaut de Nginx pour vérifier que le logiciel fonctionne correctement en consultant le nom de domaine ou l'adresse IP publique de votre serveur dans votre navigateur Web.

mais si vous avez besoin de la dernière version, vous devrez peut-être ajouter un ppa officiel.

```
sudo add-apt-repository ppa:nginx/stable
sudo apt-get update
sudo apt-get install nginx
```

Redémarrer NGINX

En tant qu'utilisateur root:

```
nginx -s restart
```

Exemple Ubuntu

```
sudo service nginx restart
```

Recharger le fichier de configuration NGINX

En tant qu'utilisateur root:

```
sudo nginx -s reload
```

Ubuntu 14.04 exemple

```
sudo service nginx reload
```

Ubuntu 16.04 exemple

```
sudo systemctl reload nginx
```

Avant de recharger, il est conseillé de vérifier la configuration des erreurs de syntaxe:

```
sudo nginx -t
```

Ou

```
sudo service nginx configtest
```

Arrêt NGINX

Exécuter en tant qu'utilisateur root.

Arrêt rapide:

```
nginx -s stop
```

Arrêt gracieux:

```
nginx -s quit
```

Nginx à l'intérieur

L'un des plus grands avantages de Nginx est la différence de fonctionnement interne par rapport aux autres serveurs populaires, en particulier Apache.

Les serveurs sont des programmes occupés car ils doivent servir des demandes provenant de plusieurs clients. Plus le serveur peut répondre à des requêtes par seconde, mieux c'est.

Nginx travaille sur un paradigme de concurrence appelé IO asynchrone.

Dans un serveur conventionnel, un thread est dédié à une requête. Cela signifie qu'une fois qu'un thread prend une requête, celle-ci est effectivement indisponible pour d'autres requêtes. Mais en réalité, un thread pourrait faire beaucoup mieux en acceptant un tas de requêtes et en les servant simultanément. Ce sont les IO asynchrones qui permettent cela.

Nginx, par conséquent, avec son architecture d'E / S asynchrone, peut servir plusieurs requêtes au sein d'un même thread.

Une autre bonne chose à propos de Nginx est son empreinte relativement réduite. Par rapport à Apache, Nginx est moins gourmand en ressources, ce qui le rend adapté aux serveurs Cloud, qui ne sont pas très puissants.

Il existe certainement d'autres serveurs Async IO, mais Nginx est le plus pris en charge parmi tous les pluginx (aka Modules Nginx).

Testez si vos modifications dans nginx.config sont valides

Ubuntu 14.04 exemple

```
sudo nginx -t
```

Lire Démarrer avec nginx en ligne: https://riptutorial.com/fr/nginx/topic/1121/demarrer-avec-nginx

Chapitre 2: Configurations Nginx

Examples

Structure du fichier de configuration

nginx est constitué de modules contrôlés par des directives spécifiées dans le fichier de configuration.

Directives simples

Une directive simple se compose du nom et des paramètres séparés par des espaces et se termine par un point-virgule (;).

Directive de bloc

Une directive de bloc a la même structure qu'une directive simple, mais au lieu du point-virgule, elle se termine par un ensemble d'instructions supplémentaires entourées d'accolades ({et}).

Le contexte

Si une directive de bloc peut avoir d'autres directives entre accolades, cela s'appelle un contexte (exemples: événements, http, serveur et emplacement).

Les directives placées dans le fichier de configuration en dehors de tout contexte sont considérées comme étant dans le contexte principal. Les événements et les directives http résident dans le contexte principal, le serveur HTTP et l'emplacement dans le serveur.

Commentaire

Le reste d'une ligne après le signe # est considéré comme un commentaire.

Test du fichier de configuration NGINX

Vous pouvez rechercher les erreurs de syntaxe et les fichiers référencés dans un fichier de configuration NGINX avant de l'exécuter avec:

```
nginx –t
```

Sinon, vous pouvez exécuter un script de service

```
service nginx configtest
```

Bien que ces deux commandes vous indiquent si votre nouvelle configuration nginx est correcte [sans tuer votre instance actuelle]. Configtest utilise le service en cours d'exécution et vous indique s'il réussit ou échoue la vérification, alors que nginx -t vérifie non seulement la configuration, mais imprime toutes les informations, avertissements et messages d'erreur.

Source: http://devget.net/nginxapache/nginx-configtest-vs-nginx-t/

Spécifier le fichier de configuration à charger

```
nginx -c <file name>
```

Démarrez NGINX avec un fichier de configuration explicite.

modifications de configuration sans redémarrage

```
nginx -s reload
```

Connexion à Syslog, incl. mappage des codes HTTP aux sévérités Syslog.

Collez cet extrait quelque part dans le bloc `http {}` ; ou placez-le dans son propre fichier dans le dossier `/etc/nginx/conf.d/` . Consultez également les documents officiels pour vous connecter à syslog.

```
#
# Access Log
#
log_format fmt_syslog '[$time_local] $status $remote_addr $http_host "$request"
$body_bytes_sent $request_time "$http_user_agent" $remote_user';
map $status $log_is_error { "~^5\d\d"     1; default 0; }
map $status $log_is_warn  { "~^4[0-8]{2}" 1; default 0; }
map $status $log_is_info  { "~^[1-3]\d\d" 1; default 0; }
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=error
fmt_syslog if=$log_is_error;
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=warn
fmt_syslog if=$log_is_warn;
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=info
fmt_syslog if=$log_is_info;
#
# Error Log
#
error_log syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2 error;
```

Cet exemple suppose que rsyslog (ou similaire) écoute sur Socket `/run/systemd/journal/syslog` - par défaut sur Debian 8 lorsque journald a activé ForwardToSyslog . En utilisant cette socket, vous contournez journald. Si cette socket n'est pas disponible, essayez plutôt `/dev/log` .

N'hésitez pas à utiliser une autre installation au lieu de local2. Vous pouvez également modifier le

log_format en fonction de vos besoins.

Limiter les méthodes de demande

Un site Web habituel ne nécessite que 3 méthodes HTTP: `GET` , `HEAD` et `POST` . Bloquer toutes les autres méthodes en utilisant limit_except :

```
location / {
    [...]
    # Note: GET includes HEAD
    limit_except GET POST {
        deny  all;
    }
    [...]
}
```

Lire Configurations Nginx en ligne: https://riptutorial.com/fr/nginx/topic/2550/configurations-nginx

Chapitre 3: Enregistrement

Examples

Exemple de base

Syntaxe

```
Syntax:  log_format name string ...;
Syntax:  access_log path [format [buffer=size] [gzip[=level]] [flush=time] [if=condition]];
access_log off;
```

Les utiliser ensemble

```
log_format compression '$remote_addr - $remote_user [$time_local] '
                       '"$request" $status $bytes_sent '
                       '"$http_referer" "$http_user_agent" "$gzip_ratio"';

access_log /spool/logs/nginx-access.log compression buffer=32k;
error_log /spool/logs/nginx-error.log;
```

Rouvrir les fichiers journaux

En tant qu'utilisateur root, exécutez:

```
nginx -s reopen
```

Évitez de vous connecter pour favicon.ico et robots.txt

```
location = /favicon.ico {
    log_not_found off;
    access_log off;
}

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}
```

Lire Enregistrement en ligne: https://riptutorial.com/fr/nginx/topic/2551/enregistrement

Chapitre 4: Intégration du blog Wordpress avec l'application rails utilisant nginx

Examples

bloc serveur nginx

Une fois que vous avez configuré et configuré les paramètres php5-fpm et wordpress, vous pouvez configurer le fichier /etc/nginx/conf/nginx.conf comme indiqué ci-dessous.

Vous devez définir les blocs d'emplacement à l'intérieur du bloc serveur et y réécrire l'URL telle que définie.

```
server {
listen 443 ssl;
        server_name abc.co.uk;
    root /home/ubuntu/www/abc/current/public;
        try_files $uri/index.html $uri @unicorn;
    ssl on;
    ssl_certificate /etc/nginx/ssl/abc.crt;
    ssl_certificate_key /etc/nginx/ssl/abc.key;
    location /blog/wp-admin/ {
        root /var/www/html/;
        index index.php;
        try_files $uri $uri/ /index.php?$args;
        location ~* \.(js|css|xml|txt|jpg)$ {
             expires 14d;
         root /var/www/html/;
             access_log off;
           }

    location ~ \.php$ {
                try_files $uri $uri/ /index.php;
                fastcgi_pass unix:/var/run/php5-fpm.sock;
                fastcgi_param SCRIPT_FILENAME $request_filename;
                fastcgi_index index.php;
                include /etc/nginx/conf/fastcgi_params;
    }

}
location ^~ /blog {
    root /var/www/html/;
    index index.php;
    try_files $uri $uri/ /index.php?$args;
    rewrite ^/blog/(.*)+$ /blog/index.php?$1;
    location ~* \.(js|css|xml|txt|jpg)$ {
         expires 14d;
     root /var/www/html/;
         access_log off;
       }

    location ~ \.php$ {
                try_files $uri $uri/ /index.php;
                fastcgi_pass unix:/var/run/php5-fpm.sock;
```

```
                fastcgi_param SCRIPT_FILENAME $request_filename;
                fastcgi_index index.php;
                include /etc/nginx/conf/fastcgi_params;
    }

  }



}
```

Lire Intégration du blog Wordpress avec l'application rails utilisant nginx en ligne: https://riptutorial.com/fr/nginx/topic/3891/integration-du-blog-wordpress-avec-l-application-rails-utilisant-nginx

Chapitre 5: nginx reverse proxy

Examples

proxy inverse simple

```
# Define which servers to include in the load balancing scheme.
# It's best to use the servers' private IPs for better performance and security.

upstream backend {

    ip_hash;
    server 10.10.10.10 slow_start=30s max_fails=3 fail_timeout=15s;
    server 10.10.10.12 slow_start=30s max_fails=3 fail_timeout=15s;

    # Activates the cache for connections to upstream servers.
    keepalive 20;
}

# This server accepts all traffic to port 80 and passes it to the upstream.
# Notice that the upstream name and the proxy_pass need to match.

server {
    listen 80;
    server_name example.com;

    location / {
        proxy_pass http://backend/;
    }
}
```

Lire nginx reverse proxy en ligne: https://riptutorial.com/fr/nginx/topic/7431/nginx-reverse-proxy

Chapitre 6: Redirections utiles

Examples

Redirection HTTPS

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name example.com www.example.com;
    return 307 https://$host$request_uri;
}
```

Une redirection 301 est également une alternative, mais si un POST est effectué sur un 301, de nombreux clients soumettront à nouveau la requête en tant que GET - ce qui échouera presque toujours sur les serveurs. Un 307 impose le même type de demande.

Rediriger les demandes vers un autre serveur

```
server {
   server_name example.com;
   return 301 $scheme://example.net$request_uri;
}
```

Redirection de site mobile

Configuration Nginx pour détecter les demandes de l'agent utilisateur mobile et les rediriger vers le site mobile.

```
location / {

    #mobile site handling as per user agent
    set $mobile_rewrite do_not_perform;     // variable to store action. default set to not
perform redirection to mobile site.

    if ($http_user_agent ~*
"(android|bb\d+|meego).+mobile|avantgo|bada\/|blackberry|blazer|compal|elaine|fennec|hiptop|iemobile|ip
|maemo|midp|mmp|mobile.+firefox|netfront|opera m(ob|in)i|palm(
os)?|phone|p(ixi|re)\/|plucker|pocket|psp|series(4|6)0|symbian|treo|up\.(browser|link)|vodafone|wap|win
ce|xda|xiino") {
        set $mobile_rewrite perform;
    }

    if ($http_user_agent ~* "^(1207|6310|6590|3gso|4thp|50[1-6]i|770s|802s|a
wa|abac|ac(er|oo|s\-
)|ai(ko|rn)|al(av|ca|co)|amoi|an(ex|ny|yw)|aptu|ar(ch|go)|as(te|us)|attw|au(di|\-m|r |s
)|avan|be(ck|ll|nq)|bi(lb|rd)|bl(ac|az)|br(e|v)w|bumb|bw\-(n|u)|c55\/|capi|ccwa|cdm\-
|cell|chtm|cldc|cmd\-|co(mp|nd)|craw|da(it|ll|ng)|dbte|dc\-s|devi|dica|dmob|do(c|p)o|ds(12|\-
d)|el(49|ai)|em(l2|ul)|er(ic|k0)|esl8|ez([4-7]0|os|wa|ze)|fetc|fly(\-|_)|g1 u|g560|gene|gf\-
5|g\-mo|go(\.w|od)|gr(ad|un)|haie|hcit|hd\-(m|p|t)|hei\-|hi(pt|ta)|hp( i|ip)|hs\-c|ht(c(\-|
|_|a|g|p|s|t)|tp)|hu(aw|tc)|i\-(20|go|ma)|i230|iac( |\-
```

```
|\/)|ibro|idea|ig01|ikom|im1k|inno|ipaq|iris|ja(t|v)a|jbro|jemu|jigs|kddi|keji|kgt(
|\/)|klon|kpt |kwc\-|kyo(c|k)|le(no|xi)|lg( g|\/(k|l|u)|50|54|\-[a-w])|libw|lynx|m1\-
w|m3ga|m50\/|ma(te|ui|xo)|mc(01|21|ca)|m\-
cr|me(rc|ri)|mi(o8|oa|ts)|mmef|mo(01|02|bi|de|do|t(\-| |o|v)|zz)|mt(50|p1|v )|mwbp|mywa|n10[0-
2]|n20[2-3]|n30(0|2)|n50(0|2|5)|n7(0(0|1)|10)|ne((c|m)\-
|on|tf|wf|wg|wt)|nok(6|i)|nzph|o2im|op(ti|wv)|oran|owg1|p800|pan(a|d|t)|pdxg|pg(13|\-([1-
8]|c))|phil|pire|pl(ay|uc)|pn\-2|po(ck|rt|se)|prox|psio|pt\-g|qa\-a|qc(07|12|21|32|60|\-[2-
7]|i\-)|qtek|r380|r600|raks|rim9|ro(ve|zo)|s55\/|sa(ge|ma|mm|ms|ny|va)|sc(01|h\-|oo|p\-
)|sdk\/|se(c(\-|0|1)|47|mc|nd|ri)|sgh\-|shar|sie(\-|m)|sk\-
0|sl(45|id)|sm(al|ar|b3|it|t5)|so(ft|ny)|sp(01|h\-|v\-|v
)|sy(01|mb)|t2(18|50)|t6(00|10|18)|ta(gt|lk)|tcl\-|tdg\-|tel(i|m)|tim\-|t\-
mo|to(pl|sh)|ts(70|m\-|m3|m5)|tx\-9|up(\.b|g1|si)|utst|v400|v750|veri|vi(rg|te)|vk(40|5[0-
3]|\-v)|vm40|voda|vulc|vx(52|53|60|61|70|80|81|83|85|98)|w3c(\-| )|webc|whit|wi(g
|nc|nw)|wmlb|wonu|x700|yas\-|your|zeto|zte\-)") {
        set $mobile_rewrite perform;
    }

    #google bot mobile handling
    if ($http_user_agent ~* "(googlebot-mobile)") {
        set $mobile_rewrite perform;
    }

    if ($mobile_rewrite = perform) {
        proxy_pass http://www.mobile-domain.com:$port;
    }
}
```

Emplacement HTTP sur le serveur HTTPS

Serveur HTTPS avec emplacement http:

```
server {
    listen 443;
    root /var/www/
    location / {
       ...
    }
    location /http {
        rewrite ^ http://$host$request_uri? permanent;
    }
}
```

Le serveur HTTP redirige vers HTTPS sauf un emplacement:

```
server {
    root /var/www/
    location / {
        rewrite ^ https://$host$request_uri? permanent;
    }
    location /http {
       ...
    }
}
```

Interdire les demandes basées sur le code de l'hôte ou du pays

Utilisez Nginx map pour analyser les champs et rejeter les demandes.

```
# Allowed hosts
map $http_host $name {
    hostnames;

    default       no;

    example.com   yes;
    *.example.com yes;
    example.org   yes;
    *.example.org yes;
    .example.net  yes;
    wap.*         yes;
}

# Allowed countries
map $geoip_country_code $allowed_country {
    default no;
    country_code_1 yes;
    country_code_2 yes;
}
```

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # Disallow access based on hostname
    if ($name = no) {
        return 444;
    }

    # Disallow access based on GeoIP
    if ($allowed_country = no) {
        return 444;
    }
...
```

Lire Redirections utiles en ligne: https://riptutorial.com/fr/nginx/topic/3631/redirections-utiles

Chapitre 7: Utilisation de nginx pour fournir des URL de navigateur propres

Examples

Redirection vs proxy inverse

Les applications Web conçues par des professionnels n'exposent pas les détails internes de l'environnement du serveur à l'utilisateur. Lorsque vous passez une commande chez votre revendeur, vous ne voyez pas (ou vous devez taper) https://mydealer.com:8443/Dealerapp/entryPage.html , mais seulement mydealer.com, bien que le serveur d'applications ait besoin de tous les détails donné dans la plus longue URL. Cela peut être réalisé via:

```
if ($scheme = http) {
    return 301 https://$server_name$request_uri;
}
```

Cela fait une **redirection** . Même si le client n'a pas demandé de connexion sécurisée, le navigateur y est immédiatement redirigé. Dans les pays ayant des lois strictes sur la confidentialité des données, vous pourriez même être obligé de le faire pour tout site commercial. La méthode de redirection est utilisée car ici le navigateur doit connaître la connexion sécurisée, sinon il ne négocierait pas avec le serveur pour le sécuriser.

```
location /app/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/entryPage.html;
}
```

Ceci est un **proxy inverse** . Il est transparent pour le navigateur. Ainsi, vous pouvez complètement masquer à l'utilisateur final si vous avez un ou plusieurs serveurs d'applications, sur quels ports ils écoutent ou comment leurs applications sont nommées. Sinon, si votre centre de données doit déplacer l'application vers un autre serveur qui écoute sur 8543, tous les signets de vos clients ne seront plus valides. Cet exemple dirige également l'utilisateur vers votre page d'entrée. Cela pourrait être omis si vous nommez la page d'entrée index.html. Mais peut-être avez-vous plusieurs pages d’entrée dans une application que l’utilisateur peut vouloir mettre en signet, vous êtes donc plus flexible si vous n'êtes pas lié à index.html.

J'ai postfixé l'URL de l'entreprise avec / app. Cela pourrait être omis si votre domaine ne sert que cette application. Au cas où, en plus de l’application, il existe également du contenu statique, comme la description de votre entreprise, vous pouvez avoir l’URL simple pour cela.

Ce proxy ne fonctionne que pour la page d'entrée. J'ai généralement besoin de deux autres schémas d'URL, un pour le contenu statique de l'application Web, comme JavaScript, CSS et les images. Je mets tous ceux qui se trouvent dans une structure de dossiers sous un dossier appelé serverapp et écris le proxy suivant:

```
location /app/serverapp/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/serverapp/;
}
```

Et un autre chemin pour les services REST; ici, le chemin de l'URL de repos ne correspond pas à un dossier, mais au chemin d'un service REST de jax-rs:

```
location /rest/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/rest/;
}
```

Un pas de plus, rarement mentionné, doit être franchi. Le serveur d'applications s'exécute sous le chemin d'URL / Dealerapp, il émettra donc un cookie de session ayant la propriété path = Dealerapp. Le navigateur ne connaît pas ce chemin et, en raison de sa règle d'origine, ignore le cookie. Nous pourrions le convaincre via le partage de ressources d’origine croisée pour autoriser cela, mais il est probablement plus facile de modifier le chemin des cookies en définissant le chemin vers /, en écrivant quelque chose comme:

```
<session-config>
    <session-timeout>720</session-timeout>
    <cookie-config>
        <name>SZSESSION</name>
        <path>/</path>
        <http-only>true</http-only>
        <secure>true</secure>
    </cookie-config>
</session-config>
```

à notre web.xml.

Lire Utilisation de nginx pour fournir des URL de navigateur propres en ligne: https://riptutorial.com/fr/nginx/topic/6270/utilisation-de-nginx-pour-fournir-des-url-de-navigateur-propres

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec nginx	Bbak, Community, I Am Batman, James, Marek Skiba, Mark Stosberg, Neo, Przemysław Jagielski, rajarshig, RamenChef, RationalDev, theDrifter, treecoder, Xevaquor
2	Configurations Nginx	Bbak, James, Pablo Fernandez, RationalDev
3	Enregistrement	Gustav, RationalDev, timbo
4	Intégration du blog Wordpress avec l'application rails utilisant nginx	Abid Iqbal
5	nginx reverse proxy	smart-developer
6	Redirections utiles	Aleksey Deryagin, Alexandre Maciel, Alexey Ten, Gaurav Kumar, Joshua DeWald, Justin W., Keelan, Muaaz Rafi, smart-developer, timbo
7	Utilisation de nginx pour fournir des URL de navigateur propres	TAM