

 無料電子ブック

学習

nginx

Free unaffiliated eBook created from
Stack Overflow contributors.

#nginx

.....	1
1: nginx	2
.....	2
.....	2
Examples.....	2
.....	2
UbuntuDebianDebianNginx.....	3
NGINX.....	3
Ubuntu.....	3
NGINX.....	3
Ubuntu 14.04.....	4
Ubuntu 16.04.....	4
NGINX.....	4
Nginx.....	4
nginx.config.....	5
2: Nginx	6
Examples.....	6
.....	6
.....	6
.....	6
.....	6
.....	6
.....	6
NGINX.....	6
.....	7
.....	7
SyslogHTTPSyslog.....	7
.....	8
3: nginx	9
Examples.....	9
.....	9
4: nginxURL	10

Examples.....	10
.....	10
5: nginxWordpress.....	12
Examples.....	12
nginx.....	12
6:	14
Examples.....	14
.....	14
.....	14
.....	14
favicon.icorobots.txt.....	14
7:	15
Examples.....	15
HTTPS.....	15
.....	15
.....	15
HTTPSHTTP.....	16
.....	16
.....	18

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

1: nginxをいめる

NGINXは「エンジンx」のようにされ、プロトコルHTTP、HTTPS、SMTP、POP3、IMAPのサーバとしてにされます。これは、リバースプロキシサーバー、HTTPキャッシュ、またはとしてできます。

これは、[ここでソースをできるオープンソースプロジェクト](#)です。

バージョン

バージョン	オリジナルリリース	バージョン		
0.5	2006-12-04	0.5.38	レガシー	2009-09-14
0.6	20070614	0.6.39	レガシー	2009-09-14
0.7	2008-05-19	0.7.69	レガシー	2011719
0.8	2009-06-02	0.8.55	レガシー	2011719
1.0	2011-04-12	1.0.15	レガシー	2012-04-12
1.2	2012-04-23	1.2.9	レガシー	2013-05-13
1.4	2013-04-24	1.4.7	レガシー	2014-03-18
1.6	2014-04-24	1.6.3	レガシー	2015-04-07
1.8	2015-04-21	1.8.1	レガシー	2016-01-26
1.9	2015-04-28	1.9.15	レガシー	2016-04-19
1.10	2016-04-26	1.10.3		2016-05-31
1.11	2016524	1.11.9	メインライン	2016-07-26

Examples

インストールとセットアップ

Nginxは、インターネットでHTTPリクエストをするためにされるWebサーバーです。

Nginxは、Linux、Windows、およびそのOSでダウンロードすることができ、ソースからビルドすることもできます。なについては、[Nginxのリファレンスをしてください](#)。

ubuntu / debian

nginxのはレポでできます。

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

システムのファイルをインストールしてしましますが、のバージョンがなは、のppaファイルをするがあります。

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

のがインストールされます。

UbuntuのようなDebianとDebianベースのディストリビューションへのNginxのインストール

のコマンドをしてnginxをインストールします。

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

デフォルトでは、Nginxはインストールににします。デフォルトのNginxランディングページにアクセスして、WebブラウザのサーバーのドメインまたはパブリックIPアドレスにアクセスして、ソフトウェアがにしていることをできます。

のバージョンがなは、のppaをするがあります。

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

NGINXをする

rootユーザーとして

```
nginx -s restart
```

Ubuntuの

```
sudo service nginx restart
```

NGINXファイルをリロードする

rootユーザーとして

```
sudo nginx -s reload
```

Ubuntu 14.04の

```
sudo service nginx reload
```

Ubuntu 16.04の

```
sudo systemctl reload nginx
```

ロードするに、`config`でエラーがないかすることをおめします。

```
sudo nginx -t
```

または

```
sudo service nginx configtest
```

シャットダウン NGINX

rootユーザーとしてします。

シャットダウン

```
nginx -s stop
```

グレースフルシャットダウン

```
nginx -s quit
```

のNginx

Nginxののの1つは、のなサーバー、にApacheにべ、にどのようにするかのです。

サーバーは、のクライアントからのをやるがあるため、ビジーのプログラムです。サーバーが1あたりのにすればするほど、はくなります。

Nginxは、IOとばれるパラダイムでします。

のサーバーでは、1つのスレッドが1つのです。つまり、スレッドがリクエストをけると、それはのリクエストにしてできなくなります。しかし、には、スレッドはのをけててにすることで、よりうまくいくがあります。IOはこれをにするものです。

したがって、NginxはIOアーキテクチャをして、1つのスレッドでくのをできます。

Nginxについてのもう一つの点は、リーンなリソースフットプリントです。Apacheとして、Nginxはリソースが少なく、サーバーをクラウドするのにしており、にはないがあります。

かにそこにのIOサーバーがありますが、NginxはpluginNginxモジュールのでもよくサポートされています。

nginx.configのがかどうかをテストする

Ubuntu 14.04の

```
sudo nginx -t
```

オンラインでnginxをいめるをむ <https://riptutorial.com/ja/nginx/topic/1121/nginxをいめる>

2: Nginxの

Examples

ファイルの

nginxは、ファイルでされたによってされるモジュールでされています。

な

なはとパラメータをスペースでってセミコロン;でわかります。

ブロック

ブロックディレクティブはなディレクティブとじをちますが、セミコロンのわりに、{と}でまれたのでわかります。

コンテキスト

ブロックディレクティブがにのディレクティブをつことができる、それはコンテキストイベント、http、サーバー、およびとばれます。

コンテキストのファイルにかれたディレクティブは、メインコンテキストとみなされます。eventsおよびhttpディレクティブは、メインコンテキスト、httpサーバー、およびserverのにします。

コメント

のののりはコメントとみなされます。

NGINX ファイルをテストする

エラーおよびされたファイルがNGINXファイルにあるかどうかをするには、のコマンドをしてください。

```
nginx -t
```

あるいは、サービススクリプトをすることもできます

```
service nginx configtest
```

これらのコマンドで、あなたの新しいnginxが[のインスタンスをせずに] okかどうかわかります。Configtestはこのサービスをして、チェックがしたかしたかをさせますが、nginx -tはチェックするだけでなく、や、エラーメッセージをします。

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

ロードするファイルをする

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

なファイルでNGINXをします。

せずにをする

```
nginx -s reload
```

SyslogへのロギングHTTPコードのSyslogへのマッピング

このスニペットをhttp {}ブロックのどこかにりけます。 /etc/nginx/conf.d/フォルダーにのファイルを/etc/nginx/conf.d/ください。 syslogにログをするためのドキュメントもてください。

```
#
# 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;
```

ここでは、rsyslogのまたはのソケットでリッスンしているとし/run/systemd/journal/syslog journaldがしたとき、それはDebianの8のデフォルトだとして- [ForwardToSyslog](#)を。このソケットをすると、journaldをバイパスします。そのソケットができないは、わりに/dev/logしてください。

local2のわりにのをにしてください。また、にじてlog_formatをすることもできます。

のメソッドをする

のウェブサイトには、GET、HEAD、POST 3つのHTTPメソッドがです。limit_exceptをしてのすべてのメソッドをブロックする

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

オンラインでNginxのをむ <https://riptutorial.com/ja/nginx/topic/2550/nginx/>の

3: nginx リバースプロキシ

Examples

なリバースプロキシ

```
# 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/;
    }
}
```

オンラインでnginxリバースプロキシをむ [https://riptutorial.com/ja/nginx/topic/7431/nginxリバースプロキシ](https://riptutorial.com/ja/nginx/topic/7431/nginx%20リバース%20プロキシ)

4: nginxをしてクリーンなブラウザURLをする

Examples

リダイレクトとリバースプロキシ

にされたWebアプリケーションは、サーバーのをユーザーにしません。ですと、 <https://mydealer.com:8443/Dealerapp/entryPage.html>はされませんまたはするがありますが、mydealer.comだけですが、アプリサーバーにはすべてののがですURLにされています。これはのでできます。

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

これはリダイレクトをいます。クライアントがセキュアなをしていないでも、ブラウザはすぐにリダイレクトされます。データプライバシーにするがしいでは、サイトでもこれをうがあります。リダイレクトは、ここではブラウザがなをるがあるため、そうでなければサーバとにしないためです。

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

これはリバースプロキシです。ブラウザにはです。したがって、エンドユーザーから、1つまたはのアプリケーションサーバー、ポートをリッスンするかどうか、またはアプリケーションのけをにすることができます。そのの、データセンターが8543でリッスンするのサーバーにアプリケーションをするがある、クライアントのすべてのブックマークはになります。このでは、ユーザーをページにします。エントリページのをindex.htmlにすると、これはできます。しかし、ユーザがブックマークしたいかもしれないアプリにのエントリページがあるかもしれないので、index.htmlにられていなければ、よりになります。

はこのURLを/appにしました。ドメインがこのアプリのみをしているは、これをするすることができます。アプリにえて、あなたのののようなくつかのコンテンツがある、そのためのなURLをつことができます。

このプロキシはエントリページでのみします。は、さらに2つのURLスキームがです。1つはJavaScript、CSS、イメージなどのWebアプリケーションのコンテンツです。はこれらをserverappというフォルダにあるフォルダにれ、のプロキシをきます

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

また、RESTサービスのもう1つのパス。ここではりのURLパスはフォルダとしませんが、jax-rs RESTサービスのパスはのとおりです。

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

まれにしかされていないもう一つのステップがです。アプリケーションサーバーはURLパス/ Dealerappのでされるため、プロパティパス= DealerappをつセッションCookieがされます。ブラウザはこのパスをしません。じポリシーにより、Cookieはされます。これをにするためにCross-Origin Resource Sharingをってできますが、/にパスをしてクッキーパスをするがです

```
<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>
```

たちのweb.xmlに。

[オンラインでnginxをしてクリーンなブラウザURLをするをむ](#)

<https://riptutorial.com/ja/nginx/topic/6270/nginx>をしてクリーンなブラウザurlをする

5: nginxをしてレールアプリケーションと Wordpressのブログの

Examples

nginxサーバーブロック

php5-fpmとwordpressのをしてしたら、のように/etc/nginx/conf/nginx.confファイルをすることができます。

サーバーブロックにロケーションブロックをし、そこにURLをきえてするがあります。

```
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;
}

}
```

オンラインでnginxをしてルールアプリケーションとWordpressのブログのをむ

<https://riptutorial.com/ja/nginx/topic/3891/nginx>をしてルールアプリケーションとwordpressのブログのを

6: ロギング

Examples

な

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

にう

```
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;
```

ログファイルをく

rootユーザーとして

```
nginx -s reopen
```

favicon.icoとrobots.txtのロギングをける

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

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

オンラインでロギングをむ <https://riptutorial.com/ja/nginx/topic/2551/ロギング>

7: なりダイレクト

Examples

HTTPSリダイレクト

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

301リダイレクトもわりですが、POSTが301にわたった、くのクライアントはをGETとしてします。ほとんどの、サーバでします。A 307はじタイプをする。

をのサーバーにリダイレクトする

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

モバイルサイトリダイレクト

モバイルユーザーエージェントからのをし、モバイルサイトにリダイレクトするためのNginx。

```
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)ods(12|\-
d)|el(49|ai)|em(12|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|imlk|inno|ipaq|iris|ja(t|v)al|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|x)|mc(01|21|ca)|m\-
cr|me(rc|ri)|mi(o8|oa|ts)|mme|f|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|t|f|w|f|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\q(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;
}
}
```

HTTPSサーバーのHTTPの

HTTPのHTTPSサーバー

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

HTTPサーバーはHTTPSにリダイレクトされます。

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

ホストコードまたはコードにづくをする

[Nginx マップ](#)をしてフィールドをし、をします。

```
# 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;
    }

    ...
}
```

オンラインでなりダイレクトをむ <https://riptutorial.com/ja/nginx/topic/3631/なりダイレクト>

クレジット

S. No		Contributors
1	nginxをいめる	Bbak , Community , I Am Batman , James , Marek Skiba , Mark Stosberg , Neo , Przemysław Jagielski , rajarshig , RamenChef , RationalDev , theDrifter , treecoder , Xevaquor
2	Nginxの	Bbak , James , Pablo Fernandez , RationalDev
3	nginxリバースプロキシ	smart-developer
4	nginxをしてクリーンなブラウザURLをす	TAM
5	nginxをしてルールアプリケーションとWordpressのブログの	Abid Iqbal
6	ロギング	Gustav , RationalDev , timbo
7	なりダイレクト	Aleksey Deryagin , Alexandre Maciel , Alexey Ten , Gaurav Kumar , Joshua DeWald , Justin W. , Keelan , Muaaz Rafi , smart-developer , timbo