

Getting started with nginx

Complete Guide

Update on March 5, 2025

NGINX is an open-source web server software

Nginx is used to host websites, manage reverse proxies, and optimize application performance.



Documentation

Official website

<https://www.nginx.com/>

List of latest versions

<https://github.com/nginx/nginx/releases>

Installation under Ubuntu

```
# installation
sudo apt-get --yes install nginx
sudo apt-get update

# start the nginx service
sudo service nginx start

# stop the service nginx
sudo service nginx stop

# restart the nginx service
sudo service nginx restart

# Uninstall
sudo apt-get purge nginx nginx-common nginx-full --yes
```

Test nginx

```
# Test nginx operation
systemctl status nginx

systemctl status nginx.service

# Test the configuration file
nginx -t
```

Concepts

root : default path

alias : specific path

```
root /var/www/html/app/;
location / {
    try_files $uri /index.html;
}
location /app1 {
    alias /var/www/html/app/app1/;
    try_files $uri$args $uri$args/ /index.html;
}
```

Comments

The **#** syntax allows you to comment out lines of code.

```
##
# Lignes de commentaires
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
```

Default configuration files

Default locations when installing nginx.

Configuration files

- **/etc/nginx/nginx.conf**
- /etc/nginx/sites-enabled/ **default**
- **/var/www/html/index.nginx-debian.html**

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    default_type application/octet-stream;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    gzip on;

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```

/etc/nginx/sites-enabled/default

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Simplified configuration

- We simplify and adapt the **nginx.conf** file

If you type **127.0.0.1/url-unknown** you get a **404 Not Found** error.

The next option will allow you to manage 404 redirects.

try_files \$uri /index.html;

- We copy a responsive and simple **index.html** file into **/var/www/html**

/var/www/html/index.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, user-scalable=no">
  <title>titre de notre page</title>
</head>

<body>
  Contenu de notre page
</body>

</html>
```

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    default_type application/octet-stream;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    gzip on;

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

        root /var/www/html;

        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
            try_files $uri /index.html;
        }
    }
}
```

Angular Application with AOT

Noticed

The root directory can be any other directory that we create.
for example root `/home/services/frontend/`

Deploying an Angular application with Nginx.

- **Creating** a Basic Angular Project
- **Compilation**
- Copying the contents of the **dist** directory to `/home/services/frontend/`
- Copy of the **nginx.conf** file containing a configuration **optimization**
- Restarting the **nginx service**

```
# Générer un projet appelé angular-starter avec options par défaut
ng new angular-starter --defaults

# Compilation
npm run build

# Copie du contenu de /dist/angular-starter dans /var/www/html/

# Redémarrage du service nginx
sudo service nginx restart
```

nginx.conf

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {

    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    default_type application/octet-stream;

    server {

        listen 80 default_server;
        listen [::]:80 default_server;

        root /home/services/frontend;

        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
            try_files $uri /index.html;
        }
    }
}
```

Angular and universal

Added Server side Rendering functionality in Angular application.

Installing Node.js.

Installation of pm2.

Creating the process.config.js file

Editing the nginx.conf file.

Copy the entire dist directory to the server.

```
# Ajout d'angular universal avec Angular CLI
ng add @nguniversal/express-engine

# Installation de nodejs sur le serveur
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -

sudo apt-get install -y nodejs

# Installation
sudo npm install -g pm2

# Installation d'un process Méthode 1
pm2 start process.config.js --env prod

# Installation d'un process Méthode 2
pm2 start dist/angular-starter/server/main.js --name frontend

# Mise en mémoire
pm2 startup
pm2 save
```

process.config.js

```
module.exports = {
  apps : [
    {
      name    : 'frontend',
      script  : 'dist/angular-starter/server/main.js',
      env: {
        COMMON_VARIABLE: 'true'  },
      env_dev : {
        NODE_ENV: 'dev'  },
      env_prod : {
        NODE_ENV: 'prod' }
    }
  ],
};
```

nginx.conf

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {

    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    default_type application/octet-stream;

    server {

        listen 80 default_server;
        listen [::]:80 default_server;

        server_name _;

        location / {
            proxy_pass http://127.0.0.1:4000;
        }

    }
}
```

Multi-domain configuration

In our example

1 domain and 3 subdomains

- `www.your_domain.com`
- `angular.your_domain.com`
- `react.your_domain.com`
- `vue.your_domain.com`

Default configuration for all urls

- `listen 80 default_server;`
`listen [::]:80 default_server;`

Followed by a specific configuration for the desired urls

- `server_name react.wosiris.com;`

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {

    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    default_type application/octet-stream;

    server {
        listen 80 default_server;
        listen [::]:80 default_server;
        location / {
            proxy_pass http://127.0.0.1:4000;
        }
    }

    server {
        listen 80;
        server_name react.wosiris.com;
        location / {
            proxy_pass http://127.0.0.1:4001;
        }
    }
}
```

Streams et Load balancer

Documentation

http://nginx.org/en/docs/http/nginx_http_upstream_module.html

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {
    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
        application/atom+xml
        application/javascript
        application/json
        application/rss+xml
        application/vnd.ms-fontobject
        application/x-font-ttf
        application/x-web-app-manifest+json
        application/xhtml+xml
        application/xml
        font/opentype
        image/svg+xml
        image/x-icon
        text/css
        text/plain
        text/x-component;

    sendfile on;
    default_type application/octet-stream;

    upstream main.module {
        server 127.0.0.1:4000;
        server 192.168.100.1:4000;
    }

    upstream react.module {
        server 192.168.0.2:4000;
    }

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

        location / {
            proxy_pass http://main.module;
        }
    }

    server {
        listen 80;
        server_name react.wosiris.com;

        location / {
            proxy_pass http://react.module;
        }
    }
}
```

Multi Application

1 domain only

3 different applications

```
# Compiling each application  
ng build --base-href=/app1/  
  
ng build --base-href=/app2/  
  
ng build --base-href=/app3/
```

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {

    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    default_type application/octet-stream;

    server {
        listen 80 default_server;
        listen [::]:80 default_server;
        root /var/www/html/app/;
        location / {
            try_files $uri /index.html;
        }
        # Variante 1 utilisation de root pour l'emplacement de index.html
        location /app1 {
            try_files $uri /app1/index.html;
        }
        # Variante 2 utilisation de alias pour l'emplacement de index.html
        location /app2 {
            alias /var/www/html/app/app2/;
            try_files $uri$args $uri$args/ /index.html;
        }
        # Variante 1 utilisation de root pour l'emplacement de index.html
        location /app3 {
            try_files $uri /app3/index.html;
        }
    }
}
```

Nginx et https

```
user nobody nogroup;
worker_processes auto;

events {
    worker_connections 1024;
}

http {
    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    default_type application/octet-stream;

    server {
        listen 80;
        server_name www.wosiris.com;
        return 301 https://www.wosiris.com$request_uri;
    }

    server {
        listen 443 ssl http2;
        listen [::]:443 ssl http2;
        server_name www.wosiris.com;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_certificate ssl/wosiris.com/fullchain.pem;
        ssl_certificate_key ssl/wosiris.com/privkey.pem;
        location / {
            root /var/www/html;
            try_files $uri /index.html;
        }
    }
}
```

```
worker_processes auto;

events {
    worker_connections 768;
}

http {

    gzip on;
    gzip_http_version 1.1;
    gzip_comp_level 5;
    gzip_min_length 256;
    gzip_proxied any;
    gzip_vary on;
    gzip_types
    application/atom+xml
    application/javascript
    application/json
    application/rss+xml
    application/vnd.ms-fontobject
    application/x-font-ttf
    application/x-web-app-manifest+json
    application/xhtml+xml
    application/xml
    font/opentype
    image/svg+xml
    image/x-icon
    text/css
    text/plain
    text/x-component;

    sendfile on;

    server {
        listen 80;
        server_name my-domain.com;
        return 301 https://my-domain.com$request_uri;
    }

    server {
        listen 443 ssl http2;
        listen [::]:443 ssl http2;
        server_name www.wosiris.com;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_certificate ssl/my-domain.com/fullchain.pem;
        ssl_certificate_key ssl/my-domain.com/privkey.pem;

        location / {
            proxy_pass http://127.0.0.1:4000;
        }
    }
}
```

Forward IP Addresses

Proxying a client's IP address

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name your-domain.com;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_certificate ssl/your-domain.com/fullchain.pem;
    ssl_certificate_key ssl/your-domain.com/privkey.pem;

    root /home/services;

    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Host $remote_addr;
        proxy_pass http://backend.module;
    }
}
```