

FreeBSD Administration - Support #633

Install Magento on an Nginx FreeBSD Web Server

06/26/2015 01:00 PM - Daniel Curtis

Status:	Closed	Start date:	06/26/2015
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Web Server	Estimated time:	3.00 hours
Target version:	FreeBSD 10	Spent time:	5.00 hours

Description

This is a guide for installing Magento on FreeBSD 10 with Nginx as the web server.

Pre-installation requirements

- Before installation of the components, make sure everything is up to date using the following command:

```
pkg update -f && pkg upgrade
```

- Next update the ports tree:

```
portsnap fetch extract
```

- Install portmaster:

```
pkg install portmaster
```

Install Nginx

- Install Nginx

```
pkg install nginx
```

- Start and enable nginx at boot:

```
echo 'nginx_enable="YES"' >> /etc/rc.conf  
service nginx start
```

- Create a configuration directory to make managing individual server blocks easier

```
mkdir /usr/local/etc/nginx/conf.d
```

- Edit the main nginx config file:

```
vi /usr/local/etc/nginx/nginx.conf
```

- And strip down the config file and add the include statement at the end to make it easier to handle various server blocks:

```
load_module /usr/local/libexec/nginx/nginx_mail_module.so;
load_module /usr/local/libexec/nginx/nginx_stream_module.so;

worker_processes 1;
error_log /var/log/nginx-error.log;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    include /usr/local/etc/nginx/conf.d/*.conf;
}
```

Install PHP

~~NOTE: Magento currently supports PHP 5.5 only.~~

- Install PHP 5.6 and other supporting packages:

```
portmaster lang/php56
```

- Install PHP extensions and a few modules:

```
portmaster lang/php56-extensions databases/php56-mysqli databases/php56-pdo_mysql www/php56-session security/php56-mcrypt graphics/php56-gd ftp/php56-curl php56-zlib php56-soap
```

- Configure the default PHP settings

```
cp /usr/local/etc/php.ini-production /usr/local/etc/php.ini
```

- Change max execution time limit in the main PHP config from the default 30 seconds to 300 seconds.

```
vi /usr/local/etc/php.ini
```

- And set:

```
max_execution_time = 300
```

Configure PHP-FPM

- Edit /usr/local/etc/php-fpm.conf:

```
vi /usr/local/etc/php-fpm.conf
```

- Make the following changes:

```
listen = /var/run/php-fpm.sock
```

```
listen.owner = www
listen.group = www
listen.mode = 0660
```

- Start and enable PHP-FPM at boot:

```
echo 'php_fpm_enable="YES"' >> /etc/rc.conf
service php-fpm start
```

- Restart nginx:

```
service nginx restart
```

Install MariaDB

- Install MariaDB server and client:

```
pkg install mariadb100-{server,client}
```

- Start and enable MariaDB at boot:

```
echo 'mysql_enable="YES"' >> /etc/rc.conf
service mysql-server start
```

- Secure your installation:

```
mysql_secure_installation
```

Configure a new MariaDB database

- Log into the MySQL console:

```
mysql -h localhost -u root -p
```

- Create the **magentouser** user with the **SuperSecretPassword** password and the **magentodb** database:

```
CREATE USER 'magentouser'@'localhost' IDENTIFIED BY 'SuperSecretPassword';
CREATE DATABASE IF NOT EXISTS `magentodb` CHARACTER SET utf8 COLLATE utf8_general_ci;
GRANT ALL PRIVILEGES ON `magentodb`.* TO 'magentouser'@'localhost';
```

```
flush privileges;
exit
```

- And wrap up by restarting the nginx and mariadb servers:

```
service nginx restart
service mysql-server restart
```

Install Magento

- Install Magento:

```
portmaster www/magento
```

- Change the ownership of magento to the nginx user:

```
chown -R www:www /usr/local/www/magento
```

- Then add write permissions to a few necessary files:

```
chmod u+w /usr/local/www/magento/var/package/*.xml
chmod u+w /usr/local/www/magento/media/xmlconnect/{original,system,custom}/ok.gif
chmod u+w /usr/local/www/magento/media/dhl/logo.jpg
```

- Add a **magento.example.com** server block:

```
vi /usr/local/etc/nginx/conf.d/magento.example.com.conf
```

- Add the following:

```
server {
    listen      80;
    server_name magento.example.com;
    root        /usr/local/www/magento;
    access_log  /var/log/magento.example.com-access.log;
    error_log   /var/log/magento.example.com-error.log;

    location / {
        index index.html index.php;
        try_files $uri $uri/ @handler;
        expires 30d;
    }

    location ^~ /app/           { deny all; }
    location ^~ /includes/     { deny all; }
    location ^~ /lib/           { deny all; }
    location ^~ /media/downloadable/ { deny all; }
    location ^~ /pkginfo/      { deny all; }
    location ^~ /report/config.xml { deny all; }
    location ^~ /var/           { deny all; }

    location /usr/local/www/magento/var/export/ {
        auth_basic "Restricted";
        auth_basic_user_file htpasswd;
        autoindex on;
    }

    location /. {
        return 404;
    }

    location @handler {
        rewrite / /index.php;
    }

    location ~ .php/ {
```

```
rewrite ^(*.php)/ $1 last;
}

location ~ .php$ {
    if (!-e $request_filename) { rewrite / /index.php last; }
    expires off;
    fastcgi_pass unix:/var/run/php-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param MAGE_RUN_CODE default;
    fastcgi_param MAGE_RUN_TYPE store;
    include fastcgi_params;
    fastcgi_read_timeout 300;
}
}
```

- Now finish the installation by going to <http://magento.example.com>

Securing Nginx With SSL

- Install OpenSSL:

```
pkg install openssl
```

- Setup the Diffie-Hellman Key Exchange Parameters

```
openssl dhparam -out /usr/local/etc/nginx/dhparam.pem 4096
```

- Generate a strong SSL key and a CSR to send for signing by a CA:

```
cd
openssl req -sha512 -out /usr/local/etc/nginx/magento.example.com.csr -new -newkey rsa:4096 -nodes -keyout /usr/local/etc/nginx/magento.example.com.key
```

- If the received SSL certificate requires additional bundle certificates, add them together like so:

```
cd /usr/local/etc/nginx
cat magento.example.com.crt magento.example.com.bundle > magento.example.com.chained.crt
```

- Setup the default site configuration:

```
vi /usr/local/etc/nginx/conf.d/magento.example.com.conf
```

- Then add or modify the configuration to look similar to the following:

```
server {
    listen 80;
    listen 443 default ssl;
    server_name magento.example.com;

    # Turn on ans set SSL key/cert
    ssl on;
    ssl_certificate /usr/local/etc/nginx/magento.example.com.crt;
    ssl_certificate_key /usr/local/etc/nginx/magento.example.com.key;

    # Strong SSL configuration
```

```

ssl_ciphers 'AES128+EECDH:AES128+EDH:!aNULL';
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_session_cache builtin:1000 shared:SSL:10m;
ssl_stapling on;
ssl_stapling_verify on;
ssl_prefer_server_ciphers on;
ssl_dhparam /usr/local/etc/nginx/dhparam.pem;
add_header Strict-Transport-Security max-age=63072000;
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;

root /usr/local/www/magento;
index index.html index.htm;
autoindex on;

## Allow a static html file to be shown first
location / {
    index index.html index.php;
    try_files $uri $uri/ @handler; ## If missing pass the URI to Magento's front handler
    expires 30d; ## Assume all files are cachable
}

## These locations would be hidden by .htaccess normally
location ^~ /app/ { deny all; }
location ^~ /includes/ { deny all; }
location ^~ /lib/ { deny all; }
location ^~ /media/downloadable/ { deny all; }
location ^~ /pkginfo/ { deny all; }
location ^~ /report/config.xml { deny all; }
location ^~ /var/ { deny all; }

## Allow admins only to view export folder
location /usr/local/www/magento/var/export/ {
    auth_basic "Restricted"; ## Message shown in login window
    auth_basic_user_file htpasswd; ## See /etc/nginx/htpasswd
    autoindex on;
}

## Disable .htaccess and other hidden files
location /. {
    return 404;
}

## Magento uses a common front handler
location @handler {
    rewrite / /index.php;
}

## Forward paths like /js/index.php/x.js to relevant handler
location ~ .php/ {
    rewrite ^(*.php)/ $1 last;
}

## Execute PHP scripts
location ~ .php$ {
    if (!-e $request_filename) { rewrite / /index.php last; } ## Catch 404s that try_files
miss
    expires off; ## Do not cache dynamic content
    fastcgi_pass unix:/var/run/php-fpm.sock;
    fastcgi_param HTTPS $fastcgi_https;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param MAGE_RUN_CODE default; ## Store code is defined in administration > C
onfiguration > Manage Stores
    fastcgi_param MAGE_RUN_TYPE store;
    include fastcgi_params; ## See /etc/nginx/fastcgi_params
}

```

```
# Uncomment to force HTTPS
# if ($scheme = http) {
#     return 301 https://$server_name$request_uri;
# }
}
```

Certificate Bundles

Some browsers may complain about a certificate signed by a well-known certificate authority, while other browsers may accept the certificate without issues. This occurs because the issuing authority has signed the server certificate using an intermediate certificate that is not present in the certificate base of well-known trusted certificate authorities which is distributed with a particular browser. In this case the authority provides a bundle of chained certificates which should be concatenated to the signed server certificate.

- The server certificate must appear before the chained certificates in the combined file:

```
cat magento.example.com.crt bundle.crt > magento.example.com.chained.crt
```

- The resulting file should be used in the `ssl_certificate` directive:

```
server {
    listen          443 ssl;
    server_name     www.example.com;
    ssl_certificate  www.example.com.chained.crt;
    ssl_certificate_key www.example.com.key;
    ...
}
```

Resources

- http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/configuring_nginx_for_magento
- <http://www.bsdcnow.tv/tutorials/nginx>
- <http://forums.freebsd.org/viewtopic.php?t=30268>
- https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html

History

#1 - 06/26/2015 03:48 PM - Daniel Curtis

- Status changed from *New* to *Resolved*

- % Done changed from *0* to *100*

#2 - 06/27/2015 09:45 AM - Daniel Curtis

- Description updated

#3 - 07/13/2015 02:07 PM - Daniel Curtis

- Status changed from *Resolved* to *Closed*

#4 - 07/29/2015 03:58 PM - Daniel Curtis

- Description updated

#5 - 08/27/2015 06:13 PM - Daniel Curtis

- Description updated

#6 - 08/27/2015 06:35 PM - Daniel Curtis

- Description updated

#7 - 08/31/2015 10:55 AM - Daniel Curtis

- Description updated

#8 - 11/16/2015 05:15 PM - Daniel Curtis

- Description updated

#9 - 09/05/2016 12:51 PM - Daniel Curtis

- Description updated

#10 - 09/05/2016 04:30 PM - Daniel Curtis

- Target version changed from FreeBSD 9 to FreeBSD 10

- Description updated

#11 - 09/11/2016 03:33 PM - Daniel Curtis

- Description updated