

FreeBSD Administration - Support #851

Install Magento 2 on an Nginx FreeBSD Web Server

09/11/2016 04:12 PM - Daniel Curtis

Status:	Closed	Start date:	09/11/2016
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Web Server	Estimated time:	1.00 hour
Target version:	FreeBSD 10	Spent time:	1.50 hour

Description

This is a guide for installing Magento 2 on FreeBSD 10 with Nginx as the web server.

**WARNING:** Installing Magento 2 from GitHub requires an account be made to connect to repo.magento.com. Go To <https://marketplace.magento.com/> and create an account. This requires personal information to be given that is irrelevant to the installation process, but mandatory nonetheless. Once created go to Developer -> My Access Keys -> Create a New Access Key

Pre-installation requirements

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

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

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

- Install PHP 5.6 and dependencies:

```
pkg install php56 php-composer php56-{bcmath,curl,gd,mbstring,mcrypt,hash,openssl,pdo_mysql,si
mplexml,soap,xml,xsl,zip,json,iconv}
```

- 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
always_populate_raw_post_data = -1
```

## 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 git:

```
pkg install git
```

- Clone the magento 2 repo from GitHub:

```
cd /usr/local/www  
git clone https://github.com/magento/magento2.git  
cd magento2
```

```
git checkout 2.0
```

- Run composer to install any missing dependencies:

```
composer install
```

- When the username prompt appears enter the **public key**
- When the password prompt appears enter the **private key**

- Change the ownership and file permissions of magento to the nginx user:

```
cd /usr/local/www/magento
chown -R www:www .
find var vendor pub/static pub/media app/etc -type f -exec chmod u+w {} \;
find var vendor pub/static pub/media app/etc -type d -exec chmod u+w {} \;
chmod u+x bin/magento
```

- Add a **magento2.example.com** server block:

```
vi /usr/local/etc/nginx/conf.d/magento2.example.com.conf
```

- Add the following:

```
upstream magento2 {
    server    unix:/var/run/php-fpm.sock;
}

server {
    listen      80;
    server_name magento2.example.com;
    root        /usr/local/www/magento2/pub;
    access_log  /var/log/magento2.example.com-access.log;
    error_log   /var/log/magento2.example.com-error.log;

    index index.php;
    autoindex off;
    charset UTF-8;
    error_page 404 403 = /errors/404.php;
    #add_header "X-UA-Compatible" "IE=Edge";

    # PHP entry point for setup application
    location ~* ^/setup($|/) {
        root /usr/local/www/magento2;
        location ~ ^/setup/index.php {
            fastcgi_pass magento2;

            fastcgi_param PHP_FLAG "session.auto_start=off \n suhosin.session.cryptua=off";
            fastcgi_param PHP_VALUE "memory_limit=768M \n max_execution_time=600";
            fastcgi_read_timeout 600s;
            fastcgi_connect_timeout 600s;

            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            include fastcgi_params;
        }

        location ~ ^/setup/(?!pub/). {
            deny all;
        }

        location ~ ^/setup/pub/ {
```

```

        add_header X-Frame-Options "SAMEORIGIN";
    }
}

# PHP entry point for update application
location ~* ^/update($|/) {
    root /usr/local/www/magento2;

    location ~ ^/update/index.php {
        fastcgi_split_path_info ^(/update/index.php)(/.*+)$;
        fastcgi_pass magento2;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        include fastcgi_params;
    }

    # Deny everything but index.php
    location ~ ^/update/(?!pub/). {
        deny all;
    }

    location ~ ^/update/pub/ {
        add_header X-Frame-Options "SAMEORIGIN";
    }
}

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

location /pub/ {
    location ~ ^/pub/media/(downloadable|customer|import|theme_customization/.*\.xml) {
        deny all;
    }
    alias /usr/local/www/magento2/pub/;
    add_header X-Frame-Options "SAMEORIGIN";
}

location /static/ {
    # Uncomment the following line in production mode
    # expires max;

    # Remove signature of the static files that is used to overcome the browser cache
    location ~ ^/static/version {
        rewrite ^/static/(version\d*/)?(.*$ /static/$2 last;
    }

    location ~* \.(ico|jpg|jpeg|png|gif|svg|js|css|swf|eot|ttf|otf|woff|woff2)$ {
        add_header Cache-Control "public";
        add_header X-Frame-Options "SAMEORIGIN";
        expires +1y;

        if (!-f $request_filename) {
            rewrite ^/static/(version\d*/)?(.*$ /static.php?resource=$2 last;
        }
    }
    location ~* \.(zip|gz|gzip|bz2|csv|xml)$ {
        add_header Cache-Control "no-store";
        add_header X-Frame-Options "SAMEORIGIN";
        expires off;

        if (!-f $request_filename) {
            rewrite ^/static/(version\d*/)?(.*$ /static.php?resource=$2 last;
        }
    }
    if (!-f $request_filename) {

```

```

        rewrite ^/static/(version\d*/)?(.*)$ /static.php?resource=$2 last;
    }
    add_header X-Frame-Options "SAMEORIGIN";
}

location /media/ {
    try_files $uri $uri/ /get.php$sis_args$args;

    location ~ ^/media/theme_customization/.*\.xml {
        deny all;
    }

    location ~* \.(ico|jpg|jpeg|png|gif|svg|js|css|swf|eot|ttf|otf|woff|woff2)$ {
        add_header Cache-Control "public";
        add_header X-Frame-Options "SAMEORIGIN";
        expires +1y;
        try_files $uri $uri/ /get.php$sis_args$args;
    }
    location ~* \.(zip|gz|gzip|bz2|csv|xml)$ {
        add_header Cache-Control "no-store";
        add_header X-Frame-Options "SAMEORIGIN";
        expires off;
        try_files $uri $uri/ /get.php$sis_args$args;
    }
    add_header X-Frame-Options "SAMEORIGIN";
}

location /media/customer/ {
    deny all;
}

location /media/downloadable/ {
    deny all;
}

location /media/import/ {
    deny all;
}

# PHP entry point for main application
location ~ (index|get|static|report|404|503)\.php$ {
    try_files $uri =404;
    fastcgi_pass magento2;
    fastcgi_buffers 1024 4k;

    fastcgi_param PHP_FLAG "session.auto_start=off \n suhosin.session.cryptua=off";
    fastcgi_param PHP_VALUE "memory_limit=768M \n max_execution_time=18000";
    fastcgi_read_timeout 600s;
    fastcgi_connect_timeout 600s;

    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}

gzip on;
gzip_disable "msie6";

gzip_comp_level 6;
gzip_min_length 1100;
gzip_buffers 16 8k;
gzip_proxied any;
gzip_types
text/plain
text/css
text/js
text/xml

```

```
text/javascript
application/javascript
application/x-javascript
application/json
application/xml
application/xml+rss
image/svg+xml;
gzip_vary on;
```

```
# Banned locations (only reached if the earlier PHP entry point regexes don't match)
location ~* (\.php$|\.htaccess$|\.git) {
    deny all;
}
}
```

- Restart nginx and php-fpm:

```
service nginx restart
service php-fpm restart
```

- Open a web browser and go to <http://magento2.example.com/setup> to complete the install process.
- Remove write permission to the app/etc folder:

```
chmod -w app/etc
```

## Magento 2 Cronjob

- Edit the www user cron table:

```
crontab -u www -e
```

- And add the following to the end of the file:

```
* * * * * /usr/local/bin/php /usr/local/www/magento2/bin/magento cron:run | grep -v "Ran j
obs by schedule" >> /usr/local/www/magento2/var/log/magento.cron.log
```

## Resources

- <https://github.com/magento/magento2>
- <http://devdocs.magento.com/guides/v2.0/install-gde/install/prepare-install.html>
- <http://devdocs.magento.com/guides/v2.0/install-gde/install/cli/install-cli.html>
- <http://devdocs.magento.com/guides/v2.0/install-gde/install/cli/install-cli-sample-data-clone.html>
- <http://magento.stackexchange.com/questions/121758/how-to-configure-nginx-for-magento-2>

## History

### #1 - 09/12/2016 03:10 PM - Daniel Curtis

- Description updated
- Status changed from New to Resolved
- % Done changed from 0 to 100
- Estimated time set to 1.00 h

### #2 - 09/12/2016 03:16 PM - Daniel Curtis

- Description updated

**#3 - 09/12/2016 03:25 PM - Daniel Curtis**

- *Project changed from Website Hosting to FreeBSD Administration*
- *Category set to Web Server*

**#4 - 09/23/2016 07:45 PM - Daniel Curtis**

- *Status changed from Resolved to Closed*

**#5 - 10/04/2016 11:15 PM - Daniel Curtis**

- *Description updated*