Onion Ecommerce Marketplace Script v1.0

Intro

This is not a copy-paste tutorial. Some software required is constantly updated and changed so you should always look for up-to-date version of software online.

You do not need to follow this tutorial. You can host Marketplace on whatever server or system you want as long as your server meets the requiremnets.

Server requirements

> VPS with at least 2GB of RAM Daemon for each coin
> that is enabled on marketplace

Software Requirements

> PHP7 (7.2 recommended) SQL Database (MySQL,PostgreSQL, SQLite, SQL Server)
> Elasticsearch (Search interface that will keep track of search records and provide great search performance) Redis (Optional, but will greatly increase app performance)

Installation instructions

Most of this will be simple copy-paste commands that you enter in your VPS. I'm writing this tutorial based on Ubuntu 18.04

When you first login on your VPS run:

```
sudo apt-get update
```

Nginx

You can use any web server you want (Apache etc.) but I will use Nginx. To install it run:

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

After installation is done we need to allow nginx in firewall by running:

```
sudo ufw allow 'Nginx HTTP'
```

After both steps are done, you should check whats your VPS IP address and enter that IP in a browser. You should see welcome to nginx ! page. If you do see it, nginx is installed correctly.

MySQL

Marketplace supports multiple databases like: MySQL,PostgreSQL, SQLite, SQL Server We will use MySQL.

```
sudo apt-get install mysql-server
```

After MySQL is installed, run

```
mysql_secure_installation
```

that will guide you trough securing your MySQL connection.

After secure installation is done, we need to create database for Marketplace by running series of commands:

```
mysql -u root -p CREATE DATABASE marketplace DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci; exit
```

(above code are 3 separate commands)

PHP

We need to install PHP (PHP-FPM) to run our code.

```
sudo apt-get install php7.2-fpm php-mysql
```

After the installation is done, we can check if php is correctly installed by running:

```
php -v
```

It should say PHP 7.2

We need to edit php.ini file. We can do that by runnin the command (assuming you installed php7.2, if you installed other version change that parameter)

```
sudo nano /etc/php/7.2/fpm/php.ini
```

Inside this file, there is commented line # cgi.fix_pathinfo=1 You need to uncomment the line and set value to cgi.fix_pathinfo=0 (without #)

In order for changes to take effect, php-fpm must be restarted.

```
sudo systemctl restart php7.2-fpm
```

Now we need to install some PHP extensions that are required by Marketplace as well as composer and unzip tools.

```
sudo apt-get install php7.2-mbstring php7.2-xml php7.2-xmlrpc php7.2-gmp php7.2-curl php7.2-gd composer unzip -y
```

(Above code is single command)

Elasticsearch

Marketplace uses Elasticsearch software that provices great search speeds and flexibility.

Elasticsearch requires Java in order to run

Oracle JDK

Add repository to apt

```
sudo add-apt-repository ppa:webupd8team/java
```

Update apt

```
sudo apt update
```

Install Java:

```
sudo apt install oracle-java8-installer
```

To see if Java is installed correctly run:

```
sudo update-alternatives --config java
```

Exit out of the command. You should see the path similar to this: /usr/lib/jvm/java-8-oracle/jre/bin/java

now we need to use that path and create environment variable.

```
sudo nano /etc/environment
```

At the bottom of the file add

```
JAVA_HOME="/usr/lib/jvm/java-8-oracle/jre/bin/java"
```

(Based on path above, if yours is different change it here)

In order for changes to take effect we need to reload environment file

```
source /etc/environment
```

To check if everything is working enter:

```
echo $JAVA_HOME
```

Command should give same path as before as output.

Elasticsearch installation

Now that java is installed, we can proceed with installation of Elasticsearch.

```
wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/d
eb/elasticsearch/2.3.1/elasticsearch-2.3.1.deb
```

(Above code is single command)

Download .deb package and install it with:

```
sudo dpkg -i elasticsearch-2.3.1.deb
```

We want Elasticsearch service to start when system boots up, so we enter:

```
sudo systemctl enable elasticsearch.service
```

Now we need to start it up.

```
sudo systemctl start elasticsearch
```

Give it 10-15 seconds from last command, and then run:

```
curl -X GET "localhost:9200"
```

If you see information about your Elasticsearch engine, then installation is completed successfully.

Elasticsearch installation error

Elasticsearch has some problems on servers with low memory. In order to make it work we need to limit max memory Java is using. To check if this is an issue run:

```
sudo service elasticsearch status
```

If you see "There is insufficient memory for the Java Runtime..." inside the text, continue, if not then your installation is not done properly and you should remove all Elasticsearch packages and go back to installing it from the start.

Enter:`edit /etc/elasticsearch/jvm.options`

change to lower memory :

```
-Xms512m
-Xmx512m
```

Then restart Elasticsearch:

```
sudo systemctl restart elasticsearch
```

Give it 10-15 seconds and then run:

```
curl -X GET "localhost:9200"
```

If you see information about your Elasticsearch engine, then installation is completed successfully.

Redis

This step is optional, but will greatly increase your app performance.

```
sudo apt install redis-server
```

After redis installation is done open redis config file:

```
sudo nano /etc/redis/redis.conf
```

In there find `supervised` and change it from `supervised no` to `supervised systemd` and save the file.

Reload Redis with:

```
sudo systemctl restart redis.service
```

And check if its running with

```
sudo systemctl status redis.service
```

To check if Redis is installed correctly enter:

```
redis-cli
```

it should open Redis interface running on port 6379. By entering ping you should get response PONG If everything is fine, type exit and exit redis-cli.

Node and NPM

We need NodeJs and NPM in order to compile our client side css files.

Install NodeJs:

```
sudo apt-get install -y nodejs
```

Install NPM:

```
sudo apt-get install -y npm
```

To check if they are installed properly run:

```
node -v
npm -v
```

(Above code are 2 commands)

Files

Now we need to copy Marketplace files to the server. Make new directory inside /var/www and put all files there. You can call it whatever you want.

Permissions

After files are copied we need to give them permissions.

Run theese commands based on your file path:

```
sudo chown -R www-data:www-data /var/www/DIRECTORY_NAME/public sudo
chmod 755 /var/www sudo chmod -R 755
/var/www/DIRECTORY_NAME/bootstrap/cache sudo chmod -R 755
/var/www/DIRECTORY_NAME/storage
```

Run:
```
php artisan storage:link
```

To link public directory with storage.

Make this folder: (for product pictures)

```
sudo mkdir /var/www/DIRECTORY_NAME/storage/public/products
```

And give it permissions

```
sudo chmod -R 755 /var/www/DIRECTORY_NAME/storage/public/products sudo
chgrp -R www-data storage/storage/public/products sudo chmod -R ug+rwx
storage/storage/public/products
```

(Above code are 3 commands)

Nginx Config

Nginx is installed but we didn't point it towards marketplace.

To edit nginx config run:

```
sudo nano /etc/nginx/sites-available/default
```

I won't explain what most of the stuff does, so here is an example of configured file:

```
server { listen 80;
        listen [::]:80; listen 443;
                listen [::]:443;

        root /var/www/market/public; index index.php index.html index.htm
        index.nginx-debian.html;

        server_name domain.com;

        location / {

    try_files $uri $uri/ /index.php?$query_string; } location ~ \.php$ { try_files $uri =404;
fastcgi_split_path_info ^(.+\.php)(/.+)$; fastcgi_pass unix:/run/php/php7.2-fpm.sock;
fastcgi_index index.php; fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name; include fastcgi_params; } }
```

after you change the parameters to reflect your environment run

```
sudo nginx -t
```

if your config file is correct output should be:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok nginx: configuration file
/etc/nginx/nginx.conf test is successful
```

Installation

After everything above is done, change current directory to the DIRECTORY_NAME you previously chose (marketplace files) and run series of commands to install all required dependencies:

```
composer install npm install npm
run prod cp .env.example .env
php artisan key:generate
```

(Above code are 4 commands)

Then open your .env file and insert database connection details:

```
sudo nano .env
```

Example of database configuration:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=marketplace
DB_USERNAME=root
DB_PASSWORD=password
```

If you did install redis, change driver from sync to redis

```
CACHE_DRIVER=redis
```

Now you can try running:

```
php artisan migrate
```

Now, you can create some dummy data, with:

```
php artisan db:seed
```

If both commands ran fine, your connection to database is configured fine. If you want to get rid of dummy data, run:

```
php artisan migrate:fresh
```

Your basic marketplace is working now, congratulations !

Connecting coins

Marketplace has support for various coins. Each coin has its on prefix in .env file as well as connection parameters. Connection paramters are:

```
HOST PORT
USERNAME
PASSWORD
```

And coin prefixes are:

> Bitcoin - BITCOIND Litecoin - LITECOIN Monero - MONERO Pivx - PIVX Dash - DASH Verge - VERGE Bitcoin Cash - BITCOIN_CASH

Knowing this, you can input connection parameters in .env accordingly. For example, for Bitcoin you would enter BITCOIND_HOST=server_ip, or for Dash DASH_PASSWORD=password.

Marketplace configuration

Marketplace configuration is split into multiple files located in config folder. Main one is marketplace.php You will find most of the config options described or self-explanatory. Other than marketplace.php You can configure levels and experience in experience.php and marketplace addresses for receiving profits in coins.php

Contact

If you find any error in code, please contact me at:

> Telegram: **@develoerhacker** (Best way to reach me):