

Сервер документації (+тимчасово для скриптів автоматизації)

Первинні налаштування

Треба створити користувача **sudo** (щоб не працювати з під **root**)

```
adduser oleksandr-d
```

занести користувача в групу **sudo**

```
sudo usermod -aG sudo oleksandr-d
```

Встановлюю PHP та модулі

```
sudo apt install -y php-cli php-openssl php-mbstring php-iconv php-tokenizer php-gd php-mysql php-xml php-  
ldap php-curl php-gd php-simplexml php-dom php-pdo
```

Встановлюю php-fpm, щоб використовувати nginx, як web-сервер.

```
sudo apt install php-fpm
```

Встановлюю MySQL

```
sudo apt install mysql-server  
sudo mysql_secure_installation
```

```
# Remove anonymous users? [Y/n]          Type Y
```

```
# Disallow root login remotely? [Y/n]    Type Y
```

```
# Remove test database and access to it? [Y/n]    Type Y
```

```
# Reload privilege tables now? [Y/n]             Type Y
```

Налаштовую php-fpm

```
sudo nano /etc/php/8.3/fpm/php.ini
```

```
file_uploads = On
allow_url_fopen = On
memory_limit = 256M
upload_max_filesize = 100M
cgi.fix_pathinfo = 0
max_execution_time = 360
date.timezone = Europe/Kyiv
```

Встановлюю Composer

Завантажте сценарій інстальатора Composer.

```
curl -sS https://getcomposer.org/installer -o composer-setup.php
```

Виконайте наведені нижче команди, щоб перевірити інстальатор.

```
HASH=`curl -sS https://composer.github.io/installer.sig`
echo $HASH
php -r "if (hash_file('SHA384', 'composer-setup.php') === '$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

Наведені вище команди захоплюють хеш-значення інстальатора та зіставляють його із завантаженим сценарієм. Ви повинні побачити такі дані, якщо інстальатор безпечний для запуску.

```
Installer verified
```

Встановіть Composer.

```
sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

Перевірте інсталяцію, перевіривши її версію.

```
composer --version
```

Створюю базу даних bookstack з користувачем bookstackuser та даю йому повний доступ до цієї бази даних.

```
sudo mysql
```

```
CREATE DATABASE bookstack;  
CREATE USER 'bookstackuser'@'localhost' IDENTIFIED WITH mysql_native_password BY '*****';  
GRANT ALL ON bookstack.* TO 'bookstackuser'@'localhost'; FLUSH PRIVILEGES;  
exit
```

Дозволяю доступ по ssh через порт 22

```
sudo ufw allow ssh
```

Встановлюємо Nginx та надаю доступ до портів

```
sudo apt install nginx  
sudo ufw allow 'Nginx Full'
```

Перевіряємо налаштування брандмауера (необхідні налаштування є в списку відповіді) та потім вмикаємо його

```
sudo ufw app list
```

```
# Available applications:
```

```
# Nginx Full  
# Nginx HTTP  
# Nginx HTTPS  
# OpenSSH
```

```
sudo ufw enable
```

```
sudo ufw status
```

```
# Status: active
```

# To	Action	From
# --	-----	----
# 22/tcp	ALLOW	Anywhere
# Nginx Full	ALLOW	Anywhere
# 22/tcp (v6)	ALLOW	Anywhere (v6)
# Nginx Full (v6)	ALLOW	Anywhere (v6)

Завантажую сам BookStack та налаштовую
Створить кореневу папку для програми BookStack.

```
sudo mkdir -p /var/www/bookstack
```

Роблю власником себе (поточного юзера з якого працюю), щоб не працювати постійно з sudo та для nginx

```
sudo chown -R oleksandr-d:www-data /var/www/bookstack
```

Переходжу в створену папку, та клоную репозиторій з BookStack (крапка в кінці треба, щоб розпакувалось в поточну папку. а не створювалась нова).

```
cd /var/www/bookstack  
git clone https://github.com/BookStackApp/BookStack.git --branch=release --single-branch .
```

Встановлюємо Composer для bookstack

```
composer install --no-dev
```

Копіюю файл, та налаштовую дані для додатку

```
cp .env.example .env
```

Відкрийте файл для редагування.

```
nano .env
```

Заповніть URL-адресу програми та дані бази даних. Якщо ви збираєтеся використовувати функції електронної пошти, заповніть дані SMTP або видаліть їх із файлу.

```
APP_URL=https://doc.rtb-system.online
```

```
DB_HOST=localhost
```

```
DB_DATABASE=bookstack
```

```
DB_USERNAME=bookstackuser
```

```
DB_PASSWORD=*****
```

```
# Can be 'smtp' or 'sendmail'
```

```
MAIL_DRIVER=smtp
```

```
MAIL_FROM_NAME="RTB Documentation"
```

```
MAIL_FROM=my@email.com
```

```
MAIL_HOST=smtp.gmail.com
```

```
MAIL_PORT=587
```

```
MAIL_USERNAME=my@email.com
```

```
MAIL_PASSWORD="**** *  
**** *  
**** *  
**** *"
```

```
MAIL_ENCRYPTION=tls
```

Далі генерую ключ для artisan який запишеться в файл .env

```
php artisan key:generate
```

Налаштовую додаткові права для юзера до каталогу з bookstack

```
sudo chown -R oleksandr-d:www-data /var/www/bookstack
```

```
sudo chmod -R 755 /var/www/bookstack
```

```
sudo chmod -R 775 /var/www/bookstack/storage /var/www/bookstack/bootstrap/cache  
/var/www/bookstack/public/uploads
```

```
sudo chmod 640 /var/www/bookstack/.env
```

Роблю artisan міграцію

```
php artisan migrate
```

Далі встановимо та налаштуємо certbot та конфігурацію nginx

Спочатку зупинимо nginx

```
sudo systemctl stop nginx.service
```

Встановлюємо Certbot

```
sudo apt install snapd
```

```
sudo snap install core
```

```
sudo snap refresh core
```

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Перевірити версію

```
certbot --version
```

Генеруємо сертифікат

```
sudo certbot certonly
```

Тепер запускаю та вмикаю nginx

```
sudo systemctl start nginx.service  
sudo systemctl enable nginx.service
```

Налаштовую конфігурацію для nginx

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

```
# HTTP — редірект на HTTPS  
server {  
    listen 80;  
    listen [::]:80;  
    server_name notes.bistats.site;  
  
    location / {  
        return 301 https://$host$request_uri;  
    }  
}  
  
# HTTPS — конфігурація SSL  
server {  
    listen 443 ssl;  
    listen [::]:443 ssl;  
    server_name notes.bistats.site;  
  
    ssl_certificate /etc/letsencrypt/live/notes.bistats.site/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/notes.bistats.site/privkey.pem;  
  
    root /var/www/bookstack/public;  
    index index.php index.html index.htm;  
  
    client_max_body_size 100M;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;
```

```
}

location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
}
```

```
sudo ln -s /etc/nginx/sites-available/bookstack /etc/nginx/sites-enabled/
```

```
sudo systemctl restart nginx.service
```

Була помилка додав ще раз права на необхідні папки.

Revision #3

Created 24 June 2024 11:59:01 by Admin

Updated 9 December 2024 13:17:41 by Admin