Install Invoice Ninja with Docker

https://www.youtube.com/embed/Uv11VPP4XP4

Invoice Ninja is a power open source offering that has been around for years now. It's only gotten better and better as time has gone by. I have been attempting to install it for a while now, but for whatever reason, just kept hitting little blocks that I couldn't quite get around. I have now gotten to the point that the install went very smoothly, and I wanted to share my steps and setup with all of you.

What You'll Need

- A server or machine to install Invoice Ninja on.
- Docker and Docker Compose installed on that machine
- (optional) A Domain Name you own and can create A or CNAME records for.
- (optional) a reverse proxy (I use NGinX Proxy Manager, but you can use any you like)
- (optional) SMTP email server for sending invoices through
- About 30 minutes of your time.

Installation of Docker and Docker Compose via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

```
chmod +x ./install_docker_nproxyman.sh
```

and then run the script with the command:

```
./install_docker_nproxyman.sh
```

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

In this case, you need to install Docker-CE and Docker Compose at a minimum, but you may also want to install NGinX Proxy Manager if you don't already have a reverse proxy on your network.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

The Domain Name

Domain names are a huge topic all their own. The short of it is, you can purchase a domain name fairly cheap these days. If you are invoicing people, you should definitely be doing it from a domain name that identifies your business well. Spend a few bucks and get one from anyone who'll give you access to set an A record. The A Record is a special DNS record that maps your domain (e.g. mygreatbusiness.com) or sub-domain (e.g. billing.mygreatbusiness.com) to your public IP address. Thus when someone tries to go to https://mygreatbusiness.com or https://billing.mygreatbusiness.com they get directed properly to your public IP address.

Installing Invoice Ninja

To begin with, make sure to check out the official documentation at the Invoice Ninja Dockerfiles site if you have any issues from this article. As these articles and videos age, the steps can sometimes be replaced with other processes that need to be done, or potentially can have processes removed that are no longer needed.

First, we need to make sure you have git, curl, and wget installed on your system. Use your systems package manager to install these.

For Debian / Ubuntu distros you can use:

```
sudo apt install git curl wget -y
```

For Fedora / Red Hat / CentOS, etc distros you can use:

```
sudo dnf install git curl wget -y
```

Once you have those installed, we'll pull down the git repository for Invoice Ninja Dockerfiles. I suggest putting your docker applications into a parent folder name "docker". To create that folder, use this command:

```
mkdir docker
```

Now, move into your 'docker' folder with

```
cd docker
```

and clone the repository:

```
 git clone https://github.com/invoiceninja/dockerfiles.git
```

Once cloned, you'll have a new directory called 'dockerfiles'. I personally don't like this name, so let's change it to something more identifiable with

```
mv dockerfiles invoiceninja
```

There, now we have a new folder called 'invoiceninja'. Let's move into it.

```
cd invoiceninja
```

In this folder, you'll find several files. We are interested in the files name 'env' and 'docker-compose.yml'. We'll be editing the contents of these files in order to get Invoice Ninja setup for our use. First, however, let's create a backup copy of each of the original files. This gives us the opportunity to make a mistake, and still have something we can revert back to if needed.

```
cp env env.bak
```

```
cp docker-compose.yml docker-compose.yml.bak
```

The two 'cp' commands copy the existing file to the backup file with the .bak extension. Now we can start editing the two original files as we see fit.

First, let's edit the 'env' file.

```
nano env
```

Feel free to make the changes manually based on my example file below, or delete all the contents from your file, then copy and paste my example below, and just change the items I have identified.

```
# IN application vars
APP_URL=http://in.localhost:8003    # <-- change this to your domain name, or the ip address
of your server
APP_KEY=<insert your generated key in here>  # <-- we'll generate this key with a command line
command
APP_DEBUG=false
REQUIRE_HTTPS=false
PHANTOMJS_PDF_GENERATION=false
PDF_GENERATOR=snappdf
TRUSTED_PROXIES='*'
```

```
QUEUE_CONNECTION=database

# DB connection
DB_HOST=db
DB_PORT=3306
DB_DATABASE=ninja
DB_USERNAME=ninja  # <-- you should change this to a user you like
DB_PASSWORD=ninja  # <-- you should make this a very long strong password

# Create initial user
# these credentials are what you'll use to login to the web interface
# DO NOT LEAVE THIS EMPTY!!!
IN_USER_EMAIL=      # <-- put the email of your admin user here
IN_PASSWORD=        # <-- put a very long strong password here for your login

# Mail options
# If you want email of invoices to work from the web interface
# you need to fill out this section. It's important that you
# know what details need to go here.
MAIL_MAILER=log
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS='user@example.com'
MAIL_FROM_NAME='Self Hosted User'

# MySQL
MYSQL_ROOT_PASSWORD=ninjaAdm1nPassword  # <-- You need to change this to a really long strong password
MYSQL_USER=ninja # <-- this needs to match the mysql user you created in the section near the top
MYSQL_PASSWORD=ninja # <-- this needs to match the strong password from the mysql section near the top
MYSQL_DATABASE=ninja

# V4 env vars
# DB_STRICT=false
```

```
# APP_CIPHER=AES-256-CBC
```

In the example above I have added comments (anything after a "#" symbol) so you know what you need to change.

Once you've changed it save with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Now, let's quickly create that APP_KEY and get it put into the 'env' file as well. For this we'll be running a docker command. The command will pull down the Invoice Ninja image we need later, start a container from the image, run a special command inside the temporary container, and spit out an APP_KEY. We'll then copy that key and paste it into our 'env' file.

Run this command to create your key:

```
docker run --rm -it invoiceninja/invoiceninja php artisan key:generate --show
```

You'll get some output on the screen. This is normal, so be patient while everything works. Once it's done you should see a line toward the bottom of the output that starts with "base64:". You need to copy that entire line (including the 'base64:' part, and paste it into the 'env' file for the APP_KEY value. So, copy it, then do

`nano env`

to open the file, and remove the text inside of the angle brackets '<' & '>', as well as the angle brackets themselves. When you're done pasting your line in the 'env file should look something like this.

`APP_KEY=base64:RGK672GY56Df#$SR3998JHKYVBE45=`

Now, save the file with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Next, we need to open our docker-compose.yml file, and make one change. To do this, enter

`nano docker-compose.yml`

Now, find the 'ports' where it read `- "80:80"`.

We want to change the port on the left side of the colon to some other port than 80. This is a very common port for webservers, and shouldn't be used. Let's change it to port 8035. So the mapping should now look like `- "8035:80"`.

This maps port 8035 on the host machine to port 80 inside the docker container, and allows us to access the Invoice Ninja web user interface on port 8035.

Save your file, with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Before we start up our containers, we need to do a couple of more things. The folders where data and configurations will be stored need to have certain ownership (specifically by the www-data user) and permissions. So let's run the following two commands to get that setup.

```
chmod 755 docker/app/public
```

```
sudo chown -R 1500:1500 docker/app
```

Now, we're ready to start our server running. Let's use the "docker compose" command to bring up our services, and then log out the status as it's starting to the terminal.

`docker compose up -d && docker compose logs -f`

Now, sit back while the images are pulled down, and the containers are started. You may see a couple of MySQL connection errors, but the system should keep re-trying until it connects. Once the output stops flowing for about a minute, you should see an INFO message with some text about things being successful.

At this point, you should be ready to go. Type CTRL + C to stop the log output in the terminal, and move over to your favorite modern browser. Enter the local IP address of your server host, and the port 8035. This should open up to the Invoice Ninja login page.

I went to http://192.168.10.50:8035

and was greeted by the login page. If you see the login page, and you only intend to run Invoice Ninja on your local network, then you can stop here and start logging in and setting up Invoice Ninja for your use. If, however you're running this for a buinsess, and you want to have a domain attached to this install, then keep moving forward with me.

Setting up our Reverse Proxy

A Proxy is a machine you have to go through to reach the internet from inside of certain networks. A revese proxy is a machine that outside devices go through in order to reach services running inside your private network. There needs to be a port foward happening on your Firewall / Router so that ports 80 adn 443 point to the host where your reverse proxy is running. That's a bit beyond the scope of this tutorial, but there are tons of resources on the internet to help you with that process on your router model iif needed.

Next, we'll login to our reverse proxy, and create a new Proxy Host. In NGinx Proxy Manager (NPM from this point on), we need to enter the domain or sub-domain name we setup with an A record in our DNS. Press 'Tab' to make it a chip, then move to the IP address field, and enter the private IP address of your host machine. Now, move to the port field, and enter the port you mapped in your docker compose file (in this case I used 8035). Enable the options for 'Block Common Explits' and 'Websocket support', then click 'Save'.

Now test the domain in your browser to make sure you are forwarded to your Invoice Ninja login page. If so, AWESome! We are moving along nicely. If not, you'll want to start troubleshooting. Make sure you setup your Domain Name, DNS A Record, Firewall port forwarding all correctly. Then start trying to ping to see where the requests go, and so on. Hopefully everything is up and running for you though.

Now, we need to get an SSL certificate so we can reach our site with full encryption. Go back to NPM, and click the 3-dot icon at the right end of your Invoice Ninja entry. Select 'Edit' from the pop-up menu, and move to the SSL tab in the window that opens. In the 'None' drop-down, select 'Request a New Certificate', then enable 'Force SSL', HTTP/2 Support', and both HSTS options. Make sure you email is filled in, then enable to accept the LetsEncrypt terms of service. Click 'Save'. The spinner will spin while LetsEncrypt challenges your domain to make sure it can reach it on port 80, and if everything is working properly, the pop-up will just close on it's own with no errors. You should see the LetsEncrypt indicator on the Invoice Ninja row now. You can again go to your invoice ninja domain and you should now be accessing it using https and full encryptiong. You can now login and begin the setup of your Invoice Ninja install.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Revision #2
Created 2024-01-05 22:34:52 UTC by Brian McGonagill
Updated 2024-01-05 23:58:19 UTC by Brian McGonagill