Install and Configure Invoice Shelf

https://www.youtube.com/embed/GtFs0yJeFUU

A few years back I showed you how to setup and install Crater Invoice. It was a clean, simple, and powerful little invoicing software. After a while Crater sadly went into and unsupported state. I have checked back in occasionally hoping to see it revived by someone, and that persistence paid off. I saw a new note stating the project had been forked, and was being updated over at InvoiceShelf. Mark one in the win column for open source!

Now, the is very exciting for me. I run a small business on the side, and have been using InvoiceNinja for about a year now due to the lack of continued dwork on Crater, but I honestly loved Crater. It was exactly what I needed, and Invoice Ninja is a bit overkill in my book.

That said, if you're looking for a clean, simple, but elegant and powerful invoicing option, then InvoiceShelf may just be for you.

What you'll need

- A server, VM, or LXC, LXD, Incus Container to run the software on
- Docker and Docker Compose installed on the machine
- (optional) A domain or sub-domain if you want your clients to also be able to access their invoices and quotes
- (optional) A Reverse Proxy
- About 15 minutes of your time.

Install Docker and Docker compose

First, we need to install Docker and Docker Compose on our server. In most cases for Linux based servers, you can do this with a single command. Open your terminal, and enter the following:

```
curl https://get.docker.com | sh
```

When prompted, enter your super user password. This will attempt to install both Docker-CE (Community Edition) and Docker Compose for you.

If you are not using one fo the many distributions supported by this script, you'll need to look up how to install docker and docker compose on your system. In particular Windows supports docker both natively through the Docker Desktop app, as well as via WSL2 (the Windows Subsystem for Linux).

Add your user to the 'docker' group with:

`sudo usermod -aG docker $USER`

then log out and back into the system.

Setup our Domain / Subdomain Name and DNS A Record

In order to have a domain name (like invoice.mygreatdomain.com) take us to our install of InvoiceShelf, we need to

1. Own the domain we want to use (e.g. mygreatdomain.com)
2. Have access to create new DNS records for the domain (you can usually do this through your domain registrar's control panel)
3. Have a public IP address either for the server we are running our software on, that leads to our ISP given internet connection, or to a VPS (small server in the cloud) where we run a VPN (a private network that allows us to connect the VPS to our home / small business network).
4. Access for traffic to be allowed on ports 80 and 443 to that public IP address.

IF you have all of the above, then let's get your A Record setup. There are a few ways to do this. If you are only wanting to run InvoiceShelf and no other services or web applications from your Public IP, then you can setup a single A-Record. In your DNS settings on your registrar, you need to create a new DNS record, and select the Type as 'A'. Then enter the subdomain for your InvoiceShelf to be accessed from.

A sub-domain is the little part before the fully qualified domain name.

```
invoice.                    mygreatdomain.com
  ^                               ^

sub-domain                   fully qualified domain name (FQDN)
```

Next, enter in your public IPv4 address for the server / network where you'll be hosting the server.

Why Do I Need a Reverse Proxy?

A reverse proxy can provide a ton of benefits.

1. It can act as the traffic director from outside your LAN to machines inside your LAN.

2. It can be the only machine on your LAN exposed directly to the internet if needed.
3. You can setup a Reverse Proxy to help direct traffic to multiple services all sharing the same Public IPv4 Address.

If your host server is not directly available via a public IPv4 address to the web, then you'll likely want a reverse proxy.

I use the NGinX Proxy Manager reverse proxy, but feel free to use any you prefer. I have a video on NGinX Proxy Manager here, if you'd like to set this up as well.

In this case, we'll setup our sub-domain using our reverse proxy. We'll enter the subdomain name, give it the local (private) IPv4 address and port number we'll be pointing to for this particular application. In my case I'm going to setup InvoiceShelf on a machine with the private IPv4 address of 192.168.10.144, and it's going to run on port 8215 (a random number I chose above 8000, but not already in use by another service).

We'll tell our reverse proxy to block common exploits, and support websockts. We'll come back in a bit to get an SSL certificate for this entry, but for now, let's setup our InvoiceShelf service.

Install InvoiceShelf

On the host VM / Server you want to instaall InvoiceShelf on, let's create a directory structure to hold this application.

```
mkdir -p docker
```

Now we'll clone the InvoiceShelf repository, and pull down the files we need in order to run InvoiceShelf.

```
git clone https://github.com/InvoiceShelf/docker
```

This will create a new folder for us called 'docker'. Let's rename this to invoiceshelf to avoid confusion later.

```
mv docker invoiceshelf
```

Now let's move into the invoiceshelf folder, and configure the settings we need.

```
cd invoiceshelf
```

```
ls
```

First, let's choose which version we want to use. You'll see several files in this directory that start with 'docker-compose' and end with '.yml'. This is just different versions and methods of running the same application. IN my case I chose to run the version with a Postgres Database backend. You can choose any one you like, and the steps for setup should be similar to mine.

Let's rename the 'docker-compose.pgsql.yml' to be just 'compose.yaml'. This allows docker-compose to know which file to run automatically, and shortens the startup command for us later.

```
mv docker-compose.pgsql.yml compose.yaml
```

Now we'll open our 'compose.yaml' file to edit it.

```
nano compose.yaml
```

In this file we need to change a few options very quickly.

1. In the section called 'invoiceshelf_db' we need to change the POSTGRES_PASSWORD value to something long, strong, and un-guessable. Use upper and lower case letters, and numbers only. Don't use symbols or spaces, as Postgres doesn't like that.
2. Next, in the 'volumes' sub-section, add a`./` to the front of 'postgres', sot he line looke like this:

```
  - ./postgres:/var/lib/postgressql/data
```

3. After this, we'll move down to the 'invoiceshelf' section. If you have port 90 in use, on your host server / system, then you can change the port number on the left side of the colon ':', but not the number on the right.

If you change that port, remember what you change it to.

4. Next, under the sub-section for 'environment', set your timezone for the two variables that have TZ in them. I am in America/Chicago, but you'll want to set your proper timezone.

5. For the DB_PASSWORD variable, make sure it matches the password you set in the first section. Copy / Paste that password from the first section to here to ensure they match exactly.

6. If you want to setup email, you need to uncomment the lines starting with a hashtag '#' sign, and then fill in the values for MAIL on each line. You need to know your SMTP values for this.

7. Finally change the port number at the end of the APP_URL and SANCTUM_STATEFUL_DOMAINS if you changed the port number in this section from 90 to a different port number. Make sure to put the port number you entered on these lines.

8. Vinally remove the lines at the bottom including the one that sys 'Volumes'.

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

Pull the InvoiceShelf Images

Now let's pull down our images for Invoice Shelf.

```
docker compose pull
```

You should see the images being pulled down from Docker Hub. If you get an error, make sure you have the spacing correct, and make sure you have all of the values correct. The errors can tell you what's wrong.

Start our Containers

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

You'll see the containers start up, and then you'll see the logs being output to the terminal. You're just watching for anything that looks liek an obvious error as it starts up.

Once it's all started, you can go to your favorite modern browser, and go to the private IPv4 address of your VM or Server, and then if you used a port other than 80, make sure to put a colon ':' then the port number after.

If you left it with Port 90, using my IP as an example we'd put "http://192.168.10.144:90".

If all has gone well, you should see the Setup Wizard for InvoiceShelf.

If you don't intend to have your clients access their invoices through this system, nor to access it from outside your network, you're done other than walking through the wizard to get things setup. You can access and use this system from this IP and port.

Finishing our Reverse Proxy Entry

If, in the start, you weren't sure what to do for the port number, you can go back and set that port in your reverse proxy. YOu can then use the SSL features of you rproxy to request a new certificate, then get it all setup for SSL encrypted access. This is important if you'll be using this outside of your private network (on the internet). You always want encrypted traffic when using anything over the internet, and it's free to sdo these days, so there's no reason not to do it.

Now try and access your site by the subdomain / domain name you wanted (for instance: https://invoice.mygreatdomain.com). You will hopefully again see the setup wizard in your browser, and now it's time to go through the wizard.

Most screens are self explanatory, however, when you get to the screen Site URL and Database, it's important you do this with your domain already setup if you are using a domain.

The Wizard defaults the values for a MySQL / MariaDB setup, so if you used that version, you'll be good other than on echange. If, however, you did Postgres like me, then we need to change a few values here.

1. Database Connection = pgsql
2. Database port = 5432
3. Database Name, Username, and Password you can get from yoru compose.yaml file. Copy / Paste is your friend.

4. Database Host = invoiceshelf_db

That last one is very important. You need to set this no matter which database backend you chose. If you don't, it will likely fail when trying to connect. Even if it makes it to a connection this time, it could fail in the future, so be sure to set it the way I have in number 4 above.

Now you can continue through the Setup Wizard.

Anything you set in the wizard can be changed in the application settings later, except for the currency type. **Make sure you choose your primary currency type properly to start with, as this cannot be changed later.**

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Buy me a Beer / Coffee:

https://paypal.me/BrianMcGonagill

Revision #2
Created 2025-03-12 11:28:32 UTC by Brian McGonagill
Updated 2025-03-12 14:50:46 UTC by Brian McGonagill