1.5.3 InvoiceShelf

- 1.5.2.1 Installation
- 1.5.2.3 Email Templates
- 01. Installation

1.5.2.1 Installation

Links

1. Github Invoiceshelf
2. Invoice Shelf PVE community script

We will install Invoiceshelf as a docker compose file but it can also be installed as proxmox LXC container. At the time of writing invoiceshelf is still in heavy development and the full functionality is not added yet. However, I try to keep it as up to date as possible ;)

1. Create a folder called invoiceshelf with

```
mkdir invoiceshelf
```

2. Then the first thing we need to do is get the docker compose files by using the following commands according to the documentation page

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

2. This will download a collection of docker compose files and some other files. Now you can choose 3 different databases and I chose for postgresql for no particular reason

3. In my case I like to deploy my docker compose files with Dockge and hence I will copy the contents of the **docker-compose.pgsql.yml** file and paste it into my dockge dashboard for a new container. However, you can also grab the file rename it to compose.yml and then save it in your preferred location.

4. Next up is the docker compose file. What is important to change here are 4 things out of which three are already obvious the mysql database, user and password fields. This is clearly added as a note in the compose file

5. Next to that you need to have a public domain name s.t customers can actually access your invoice platform and pay their bills. This domain name needs to be added 3 times at the following places

1. APP_URL=https://invoice.example.com
2. SESSION_DOMAIN=invoice.example.com
3. SANCTUM_STATEFUL_DOMAINS=invoice.example.com

6. Finally we will add mail to it as well with the following environment variables

- MAIL_DRIVER=SMTP
- MAIL_HOST=smtp.example.com
- MAIL_PORT=587
- MAIL_USERNAME=email@email.org
- MAIL_PASSWORD=emailpassword
- MAIL_ENCRYPTION=starttls

7. Finally, you can decide if you keep docker volumes or mounts. I did not get it to work with mounts and hence I stayed with volumes. Since, the data quantity is pretty low it did not really matter anyways since we are not attaching an nfs share or something to it.

8. Then it is ready to sping up the docker compose file

9. The installation wizard is pretty simple however you need to make sure you get one thing right at the **SITE URL & DATABASE** section. When it asks for the *DATABASE HOST*. In this case you need to fill in the name you gave the database container. If you have not changed anything this will be **invoiceshelf-pgsql**.

10. Finally, also make sure you provide the right details for your email address. This will be important since from that email emails will be send to customers.

```
#-----------------------------------------
# InvoiceShelf PostgresSQL docker-compose variant
# Repo : https://github.com/InvoiceShelf/docker
#-----------------------------------------

services:
  database:
    container_name: invoiceshelf-pgsql
    image: postgres:15
    environment:
      - POSTGRES_PASSWORD=sdfij2309rrjlkkfj0ejfsjdflkjsdfojwefkjHD # ⚠ Change DB password
before proceeding
      - POSTGRES_USER=invoiceshelf # ⚠ Change DB username before proceeding
      - POSTGRES_DB=invoiceshelf # Change DB name before proceeding
    volumes:
      - invoiceshelf_postgres:/var/lib/postgresql/data
    networks:
      - invoiceshelf
    restart: unless-stopped
  webapp:
    image: invoiceshelf/invoiceshelf:nightly
    container_name: invoiceshelf-variant-pgsql
```

```
    environment:
      - APP_NAME=InvoiceShelf
      - APP_ENV=production
      - APP_DEBUG=false
      # APP_URL should have:
      # 1. Protocol http or https if you are accessing it via reverse proxy with
      #    SSL, then it will be https, if via local IP, then http.
      # 2. IP address of the server or the domain name
      # 3. (optional) If you are using a custom port such as :8080 or another.
      # REF: https://github.com/InvoiceShelf/InvoiceShelf/issues/213#issuecomment-3026790543

      - APP_URL=https://invoice.rfeyn.org
      # SESSION_DOMAIN should have:
      # 1. IP address of the server or the domain name
      # 2. (optional) If you are using a custom port such as :8080 or another.
      # REF: https://github.com/InvoiceShelf/InvoiceShelf/issues/213#issuecomment-3026790543
      - SESSION_DOMAIN=invoice.rfeyn.org
      # SANCTUM_STATEFUL_DOMAINS should have:
      # 1. IP address of the server or the domain name
      #    1.1 (optional) If you are using a custom port such as :8080 or another.
      # 2. (optional) If you are planning to access the application from different
      #    domains, you must add a comma between each domain
      # REF: https://github.com/InvoiceShelf/InvoiceShelf/issues/213#issuecomment-3026790543
      - SANCTUM_STATEFUL_DOMAINS=invoice.rfeyn.org
      - DB_CONNECTION=pgsql
      - DB_HOST=database
      - DB_PORT=5432
      - DB_DATABASE=invoiceshelf # ⚠ Change DB name before proceeding
      - DB_USERNAME=invoiceshelf # ⚠ Change DB username before proceeding
      - DB_PASSWORD=sdfij2309rrjlkkfj0ejfsjdflkjsdfojwefkjHD # ⚠ Change DB password before
proceeding
      - DB_PASSWORD_FILE=
      - CACHE_STORE=file
      - SESSION_DRIVER=file
      - SESSION_LIFETIME=240
      - AUTORUN_ENABLED=true
      - AUTORUN_LARAVEL_MIGRATION=false
      - AUTORUN_LARAVEL_OPTIMIZE=false
      - PHP_OPCACHE_ENABLE=1
      - MAIL_DRIVER=SMTP
```

```
      - MAIL_HOST=smtp.porkbun.com
      - MAIL_PORT=587
      - MAIL_USERNAME=dp@p2pnode.org
      - MAIL_PASSWORD=PuV9#@pUca49C70e2V46&keg
      - MAIL_ENCRYPTION=starttls
    volumes:
      - invoiceshelf_storage:/var/www/html/storage/
      - invoiceshelf_modules:/var/www/html/Modules/
    ports:
      - 8090:8080 # 8090 is the public port.
    networks:
      - invoiceshelf
    restart: unless-stopped
    depends_on:
      - database
networks:
  invoiceshelf: null
volumes:
  invoiceshelf_postgres: null
  invoiceshelf_storage: null
  invoiceshelf_modules: null
```

drawing-1-1779878580.png

Edit this Page >>

1.5.2.3 Email Templates

Nederlands

Factuur

Beste {BILLING_ADDRESS_NAME},

- Er staat een factuur voor u klaar van □□ **{COMPANY_NAME}** voor het uitvoeren van horeca werkzaamheden.
- Vind de factuur als PDF bestand in onderstaande link
- Mochten er vragen zijn, twijfel dan niet en bericht me meteen □□
- Uiterste betaal datum **{INVOICE_DUE_DATE}** □□
- Ontvang een betalings bevestiging □□
- □□iDeal -> {CUSTOM_INVOICE_BETAALLINK}

Met vriendelijke groet,

DPlancer
Dennis Paassen
KVK: {CUSTOM_INVOICE_KVK}

Betaling

□□□□Wohoooo!! Bedankt voor je betaling!!

U kunt uw betalings bevestiging vinden door op onderstaande knop te drukken

Met vriendelijke groet,

Dennis Paassen
□□**DPlancer**

English

Invoice

Dear {BILLING_ADDRESS_NAME}

- An invoice from □□ **{COMPANY_NAME}** is ready for you for hospitality services
- Find the invoice as a PDF file in the link below
- If you have any questions, please don't hesitate to contact me □□
- Invoice due date **{INVOICE_DUE_DATE}** □□
- You will receive a payment confirmation □□
- □□**iDeal** -> {CUSTOM_INVOICE_BETAALLINK}

Kind regards,

DPlancer
Dennis Paassen
KVK: {CUSTOM_INVOICE_KVK}

Payment

□□□□Wohoooo!! Thanks for your payment

You can find your payment confirmation by clicking on the lick below

Kind regards,

Dennis Paassen
□□**DPlancer**

01. Installation

```
services:
  app:
    build:
      context: .
    image: invoiceninja/invoiceninja-debian:latest
    restart: unless-stopped
    env_file:
      - ./.env
    volumes:
      - ./app_public:/var/www/html/public
      - ./app_storage:/var/www/html/storage
    depends_on:
      mysql:
        condition: service_healthy
      redis:
        condition: service_healthy
  nginx:
    image: nginx:alpine
    restart: unless-stopped
    ports:
      - 8333:80
      - 6443:443
    volumes:
      - ./nginx:/etc/nginx/conf.d:ro
      - ./app_public:/var/www/html/public:ro
      - ./app_storage:/var/www/html/storage:ro
    depends_on:
      app:
        condition: service_healthy
  mysql:
    image: mysql:8
    restart: unless-stopped
    environment:
      MYSQL_DATABASE: ${DB_DATABASE}
      MYSQL_USER: ${DB_USERNAME}
```

```
      MYSQL_PASSWORD: ${DB_PASSWORD}
      MYSQL_ROOT_PASSWORD: ${DB_ROOT_PASSWORD}
    volumes:
      - ./mysql_data:/var/lib/mysql
    healthcheck:
      test:
        - CMD
        - mysqladmin
        - ping
        - -h
        - localhost
        - -u${MYSQL_USER}
        - -p${MYSQL_PASSWORD}
  redis:
    image: redis:alpine
    restart: unless-stopped
    volumes:
      - ./redis_data:/data
    healthcheck:
      test:
        - CMD
        - redis-cli
        - ping
networks: {}
```

```
# IN application vars
APP_URL=https://invoice.rfeyn.org
APP_KEY=base64:oUr5FRUWNrM42GeyksHF33ntNtlh29KW5vg2J2brnJ4=
APP_ENV=production
APP_DEBUG=true
REQUIRE_HTTPS=false
PHANTOMJS_PDF_GENERATION=true
PDF_GENERATOR=snappdf
TRUSTED_PROXIES='*'



CACHE_DRIVER=redis
QUEUE_CONNECTION=redis
SESSION_DRIVER=redis
```

```
REDIS_HOST=redis
REDIS_PASSWORD=null
REDIS_PORT=6379

FILESYSTEM_DISK=debian_docker

# DB connection
DB_HOST=mysql
DB_PORT=3306
DB_DATABASE=ninjadplancer
DB_USERNAME=ninja
DB_PASSWORD=sf9823fsjlfkjwioefjlskfh0923jfklsdjf
DB_ROOT_PASSWORD=sdpoifj293fj09030rjnsnedfo3498t
DB_CONNECTION=mysql

# Create initial user
# Default to these values if empty
IN_USER_EMAIL=dp@p2pnode.org
IN_PASSWORD=nvQi5W14blOnpv
# IN_USER_EMAIL=
# IN_PASSWORD=

# Mail options
MAIL_MAILER=log
MAIL_HOST=smtp.porkbun.com
MAIL_PORT=587
MAIL_USERNAME=dp@p2pnode.org
MAIL_PASSWORD=PuV9#@pUca49C70e2V46&keg
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS='dp@p2pnode.org'
MAIL_FROM_NAME="DPLancer Invoices"

# MySQL
MYSQL_ROOT_PASSWORD=sdpoifj293fj09030rjnsnedfo3498t
MYSQL_USER=ninja
MYSQL_PASSWORD=sf9823fsjlfkjwioefjlskfh0923jfklsdjf
MYSQL_DATABASE=ninjadplancer

# GoCardless/Nordigen API key for banking integration
NORDIGEN_SECRET_ID=
```

```
NORDIGEN_SECRET_KEY=

IS_DOCKER=true
SCOUT_DRIVER=null
#SNAPPDF_CHROMIUM_PATH=/usr/bin/google-chrome-stable
```