

InvoiceNinja

Neu auf x86

<https://github.com/invoiceninja/dockerfiles>

```
git clone https://github.com/invoiceninja/dockerfiles.git -b debian
cd dockerfiles/debian
```

docker-compose.yml:

```
services:
  app:
    build:
      context: .
    image: invoiceninja/invoiceninja-debian:${TAG:-latest}
    restart: unless-stopped
    env_file:
      - ./env
    volumes:
      - ./php/php.ini:/usr/local/etc/php/conf.d/invoiceninja.ini:ro
      - ./php/php-fpm.conf:/usr/local/etc/php-fpm.d/invoiceninja.conf:ro
      - ./supervisor/supervisord.conf:/etc/supervisor/conf.d/supervisord.conf:ro
      - public:/var/www/html/public
      - storage:/var/www/html/storage
    depends_on:
      mysql:
        condition: service_healthy
      redis:
        condition: service_healthy
    healthcheck:
      test: ["CMD", "php", "-v"]
      interval: 10s
```

```
timeout: 5s
retries: 3
start_period: 60s
```

nginx:

```
image: nginx:alpine
restart: unless-stopped
ports:
  - "8013:80"
volumes:
  - ./nginx:/etc/nginx/conf.d:ro
  - public:/var/www/html/public:ro
  - 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: ${DB_ROOT_PASSWORD}
volumes:
  - mysql:/var/lib/mysql
healthcheck:
  test:
    [
      "CMD",
      "mysqladmin",
      "ping",
      "-h",
      "localhost",
      "-u${MYSQL_USER}",
      "-p${MYSQL_PASSWORD}",
    ]
  interval: 10s
  timeout: 5s
```

```
    retries: 5
    start_period: 30s

redis:
  image: redis:alpine
  restart: unless-stopped
  volumes:
    - redis:/data
  healthcheck:
    test: ["CMD", "redis-cli", "ping"]

volumes:
  public:
  storage:
  mysql:
  redis:
```

für .env die var APP_KEY erzeugen mit

```
# If you haven't started the containers yet:
docker run --rm -it invoiceninja/invoiceninja-debian php artisan key:generate --show
# Or if your containers are already running:
docker compose exec app php artisan key:generate --show
```

Copy the entire string and insert in the .env file at `APP_KEY=base64....=`

! Außerdem noch IN_USER_EMAIL und IN_PASSWORD anpassen

.env

```
TAG=5.12.45

# IN application vars
#APP_URL=http://localhost:8012
APP_URL=https://invoiceninja.MEINEDOMAIN.it
APP_KEY=base64:+jpwe.....EGPGKkG/fxkEU=
```

```
APP_ENV=production
APP_DEBUG=true
REQUIRE_HTTPS=false
PHANTOMJS_PDF_GENERATION=false
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=ninja
DB_USERNAME=ninja
DB_PASSWORD=c94f68e.....6f0e9741
DB_ROOT_PASSWORD=d69f756.....11207d
DB_CONNECTION=mysql

# Create initial user
# Default to these values if empty
IN_USER_EMAIL=admin@example.com
IN_PASSWORD=changeme!
# IN_USER_EMAIL=
# IN_PASSWORD=

# Mail options
MAIL_MAILER=log
MAIL_HOST=smtplib.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=d69f756.....11207d
MYSQL_USER=ninja
MYSQL_PASSWORD=c94f68e12d.....0e6f0e9741
MYSQL_DATABASE=ninja

# 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
```

Alt ab hier:

Erster Versuch auf Fernnetz

```
### AUF EIS GELEGT, PROBLEME MIT MYSQL ENVs?

services:
  server:
    image: nginx
    restart: unless-stopped
    #env_file: env
    volumes:
      # Vhost configuration
      #-
      /var/lib/docker/volumes/invoiceninja/config/caddy/Caddyfile:/etc/caddy/Caddyfile:ro
      - /var/lib/docker/volumes/invoiceninja/config/nginx/in-vhost.conf:/etc/nginx/conf.d/in-
vhost.conf:ro
```

```
- /var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:ro
depends_on:
  - app
# Run webserver nginx on port 80
# Feel free to modify depending what port is already occupied
ports:
  - "8006:80"
  #- "443:443"
networks:
  - invoiceninja
extra_hosts:
  - "in5.localhost:192.168.0.124 " #host and ip

app:
  image: invoiceninja/invoiceninja:5.10
  #env_file: env
  restart: unless-stopped
  volumes:
    - /var/lib/docker/volumes/invoiceninja/config/hosts:/etc/hosts:ro
    -
/var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:rw,delegated
-
/var/lib/docker/volumes/invoiceninja/docker/app/storage:/var/www/app/storage:rw,delegated
  - /var/lib/docker/volumes/invoiceninja/config/php/php.ini:/usr/local/etc/php/php.ini
  - /var/lib/docker/volumes/invoiceninja/config/php/php-cli.ini:/usr/local/etc/php/php-
cli.ini

  depends_on:
    - db
  networks:
    - invoiceninja
  extra_hosts:
    - "in5.localhost:192.168.0.124 " #host and ip

db:
  image: mysql:8
# When running on ARM64 use MariaDB instead of MySQL
# image: mariadb:10.4
# For auto DB backups comment out image and use the build block below
# build:
```

```
# context: ./config/mysql
restart: unless-stopped
#env_file: env
volumes:
  - /var/lib/docker/volumes/invoiceninja/docker/mysql/data:/var/lib/mysql:rw,delegated

  # remove comments for next 4 lines if you want auto sql backups
  #- /var/lib/docker/volumes/invoiceninja/docker/mysql/bak:/backups:rw
  #- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.daily/daily:ro
  #- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.weekly/weekly:ro
  #- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.monthly/monthly:ro
networks:
  - invoiceninja
extra_hosts:
  - "in5.localhost:192.168.0.124 " #host and ip

# THIS IS ONLY A VALID CONFIGURATION FOR IN 4. DO NOT USE FOR IN 5.
# cron:
# image: invoiceninja/invoiceninja:alpine-4
# volumes:
# -
/var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:rw,delegated
# -
/var/lib/docker/volumes/invoiceninja/docker/app/storage:/var/www/app/storage:rw,delegated
# -
/var/lib/docker/volumes/invoiceninja/docker/app/public/logo:/var/www/app/public/logo:rw,delegated
ted
# entrypoint: |
# /bin/sh -c 'sh -s <<EOF
# trap "break;exit" SIGHUP SIGINT SIGTERM
# sleep 300s
# while /bin/true; do
# ./artisan ninja:send-invoices
# ./artisan ninja:send-reminders
# sleep 1d
# done
# EOF'
```

```
# networks:
#   - invoiceninja
#
```

```
networks:
  invoiceninja:
```

Zweiter Versuch auf Heimnetz

```
### AUF EIS GELEGT, PROBLEME MIT VOLUMES FÜR DATEIEN (zb hosts)
```

```
services:
```

```
  server:
```

```
    image: nginx
```

```
    restart: unless-stopped
```

```
    #env_file: env
```

```
    volumes:
```

```
      # Vhost configuration
```

```
      #-
```

```
/var/lib/docker/volumes/invoiceninja/config/caddy/Caddyfile:/etc/caddy/Caddyfile:ro
```

```
  - /var/lib/docker/volumes/invoiceninja/config/nginx/in-vhost.conf:/etc/nginx/conf.d/in-
```

```
vhost.conf:ro
```

```
  - /var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:ro
```

```
  depends_on:
```

```
    - app
```

```
  # Run webserver nginx on port 80
```

```
  # Feel free to modify depending what port is already occupied
```

```
  ports:
```

```
    - "8006:80"
```

```
    #- "443:443"
```

```
  networks:
```

```
    - invoiceninja
```

```
  extra_hosts:
```

```
    - "in5.localhost:192.168.0.124 " #host and ip
```

```
app:
```

```
  image: invoiceninja/invoiceninja:5.10
```

```
  #env_file: env
```

```
  restart: unless-stopped
```

```
volumes:
  - /var/lib/docker/volumes/invoiceninja/config/hosts:/etc/hosts:ro
  -
/var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:rw,delegated
  -
/var/lib/docker/volumes/invoiceninja/docker/app/storage:/var/www/app/storage:rw,delegated
  - /var/lib/docker/volumes/invoiceninja/config/php/php.ini:/usr/local/etc/php/php.ini
  - /var/lib/docker/volumes/invoiceninja/config/php/php-cli.ini:/usr/local/etc/php/php-
cli.ini

depends_on:
  - db

networks:
  - invoiceninja

extra_hosts:
  - "in5.localhost:192.168.0.124 " #host and ip

db:
  image: mysql:8
#   When running on ARM64 use MariaDB instead of MySQL
#   image: mariadb:10.4
#   For auto DB backups comment out image and use the build block below
#   build:
#     context: ./config/mysql
  restart: unless-stopped
#env_file: env
volumes:
  - /var/lib/docker/volumes/invoiceninja/docker/mysql/data:/var/lib/mysql:rw,delegated

# remove comments for next 4 lines if you want auto sql backups
#- /var/lib/docker/volumes/invoiceninja/docker/mysql/bak:/backups:rw
#- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.daily/daily:ro
#- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.weekly/weekly:ro
#- /var/lib/docker/volumes/invoiceninja/config/mysql/backup-
script:/etc/cron.monthly/monthly:ro

networks:
  - invoiceninja

extra_hosts:
```

```
- "in5.localhost:192.168.0.124 " #host and ip

# THIS IS ONLY A VALID CONFIGURATION FOR IN 4. DO NOT USE FOR IN 5.
# cron:
# image: invoiceninja/invoiceninja:alpine-4
# volumes:
# -
/var/lib/docker/volumes/invoiceninja/docker/app/public:/var/www/app/public:rw,delegated
# -
/var/lib/docker/volumes/invoiceninja/docker/app/storage:/var/www/app/storage:rw,delegated
# -
/var/lib/docker/volumes/invoiceninja/docker/app/public/logo:/var/www/app/public/logo:rw,delegated
ted
# entrypoint: |
# /bin/sh -c 'sh -s <<EOF
# trap "break;exit" SIGHUP SIGINT SIGTERM
# sleep 300s
# while /bin/true; do
#     ./artisan ninja:send-invoices
#     ./artisan ninja:send-reminders
#     sleep 1d
# done
# EOF'
# networks:
# - invoiceninja
#

networks:
  invoiceninja:
```

Revision #8

Created 2024-10-17 09:31:53 UTC

Updated 2026-01-21 20:15:54 UTC