

AI-03096 Docker Inventree FINAL

👤 Owner	Ⓜ Rigel Arcayan
☰ Tags	Project
🕒 Created time	@October 21, 2025 8:54 PM

```
sudo mkdir -p
/home/dockerlab/{inventree,postgres,pgadmin,npm,phpmyadmin,redis,mariadb,portainer,npm}
sudo mkdir -p /home/dockerlab/inventree/data
sudo mkdir -p /home/dockerlab/inventree/logs
sudo mkdir -p /home/dockerlab/inventree/config
sudo mkdir -p /home/dockerlab/redis/data
sudo mkdir -p /home/dockerlab/mariadb/data
sudo mkdir -p /home/dockerlab/postgres/data
sudo mkdir -p /home/dockerlab/pgadmin/data
sudo mkdir -p /home/dockerlab/npm/{data,letsencrypt}
sudo mkdir -p /home/dockerlab/phpmyadmin/data
sudo mkdir -p /home/dockerlab/phpmyadmin/config
sudo mkdir -p /home/dockerlab/portainer/data
sudo mkdir -p /home/dockerlab/npm/data
sudo mkdir -p /home/dockerlab/npm/letsencrypt
```

Create an inventree docker deployment

use the following info:

provide a folder structure under /home/dockerlab directory.

provide persistent data inside /home/dockerlab

provide postgres init.sql for inventree

include inventree-db, inventree-cache

postgres host name: postgres

postgres user: postgres_user

postgres password: SecurePassword!

pgadmin user: rarcayan@gmail.com

pgadmin password: SecurePassword!

inventreedb: inventreedb

inventree user: inventree_user

inventree password: SecurePassword!

docker bridge network: dockerlab

```
# Docker Cleanup Command
sudo docker rm -f $(docker ps -aq) # Stops & removes all running/stopped containers
sudo docker volume prune -f # Removes unused volumes
sudo docker network rm dockerlab # Removes old dockerlab network (if exists)
sudo rm -rf /home/dockerlab # Deletes all data/configs for a full reset
sudo mkdir -p /home/dockerlab
sudo chown -R $USER:$USER /home/dockerlab
sudo chmod -R 755 /home/dockerlab
sudo mkdir -p /home/dockerlab/pgadmin/sessions
sudo chown -R 5050:5050 /home/dockerlab/pgadmin #inside docker pgadmin user
sudo chmod -R 700 /home/dockerlab/pgadmin
```

Post Ubuntu Setup

```
sudo apt install cockpit
sudo systemctl enable --now cockpit.socket
sudo apt update
sudo apt install -y wget apt-transport-https software-properties-common
wget -q http://www.webmin.com/jcameron-key.asc -O- | sudo tee /etc/apt/trusted.gpg.d/
webmin.asc
sudo add-apt-repository "deb http://download.webmin.com/download/repository sarge co
ntrib"
sudo apt update
sudo apt install -y webmin
```

```
sudo nano /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0: # Replace with your actual network interface name
      dhcp4: no
      addresses:
        - 192.168.1.3/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

```
sudo netplan apply
```

```
# Ubuntu Upgrade to the latest
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
sudo apt install update-manager-core
sudo nano /etc/update-manager/release-upgrades # set Prompt=normal
sudo do-release-upgrade -d
```

docker-compose.yml

```
# Docker compose recipe for a production-ready InvenTree setup, with the following containers:
# - PostgreSQL as the database backend
# - gunicorn as the InvenTree web server
# - django-q as the InvenTree background worker process
# - Caddy as a reverse proxy
# - redis as the cache manager (optional, disabled by default)

# -----
# READ BEFORE STARTING!
# -----

# -----
# Setting environment variables
# -----
# Shared environment variables should be stored in the .env file
# Changes made to this file are reflected across all containers!
#
# IMPORTANT NOTE:
# You should not have to change *anything* within this docker-compose.yml file!
# Instead, make any changes in the .env file!

# -----
# InvenTree Image Versions
# -----
# By default, this docker-compose script targets the STABLE version of InvenTree,
# image: inventree/inventree:stable
#
# To run the LATEST (development) version of InvenTree,
# change the INVENTREE_TAG variable (in the .env file) to "latest"
```

```

#
# Alternatively, you could target a specific tagged release version with (for example):
# INVENTREE_TAG=0.7.5
#

# -----
# Docker compose customization
# -----
# If you wish to customize the docker-compose script, you should only do so if you understand the stack!
# Do not expect support for customizations that are not part of the standard InvenTree set up!

services:
  # Database service
  # Use PostgreSQL as the database backend
  inventree-db:
    image: postgres:17
    container_name: inventree-db
    expose:
      - ${INVENTREE_DB_PORT:-5432}/tcp
    environment:
      - PGDATA=/var/lib/postgresql/data/pgdb
      - POSTGRES_USER=${INVENTREE_DB_USER:?You must provide the 'INVENTREE_DB_USER' variable in the .env file}
      - POSTGRES_PASSWORD=${INVENTREE_DB_PASSWORD:?You must provide the 'INVENTREE_DB_PASSWORD' variable in the .env file}
      - POSTGRES_DB=${INVENTREE_DB_NAME:?You must provide the 'INVENTREE_DB_NAME' variable in the .env file}
    volumes:
      # Map 'data' volume such that postgres database is stored externally
      - ${INVENTREE_EXT_VOLUME:?You must specify the 'INVENTREE_EXT_VOLUME' variable in the .env file!}:/var/lib/postgresql/data/:z
    restart: unless-stopped

  # redis acts as database cache manager
  inventree-cache:
    image: redis:7-alpine
    container_name: inventree-cache
    env_file:
      - .env

```

```

expose:
  - ${INVENTREE_CACHE_PORT:-6379}
restart: always

# InvenTree web server service
# Uses gunicorn as the web server
inventree-server:
  # If you wish to specify a particular InvenTree version, do so here
  image: inventree/inventree:${INVENTREE_TAG:-stable}
  container_name: inventree-server
  # Only change this port if you understand the stack.
  expose:
    - 8000
  ports:
    - ${INVENTREE_GUNICORN_PORT:-8000}:8000
  depends_on:
    - inventree-db
    - inventree-cache
  env_file:
    - .env
  volumes:
    # Data volume must map to /home/inventree/data
    - ${INVENTREE_EXT_VOLUME}:/home/inventree/data:z
  restart: unless-stopped

# Background worker process handles long-running or periodic tasks
inventree-worker:
  # If you wish to specify a particular InvenTree version, do so here
  image: inventree/inventree:${INVENTREE_TAG:-stable}
  container_name: inventree-worker
  command: invoke worker
  depends_on:
    - inventree-server
  env_file:
    - .env
  volumes:
    # Data volume must map to /home/inventree/data
    - ${INVENTREE_EXT_VOLUME}:/home/inventree/data:z
  restart: unless-stopped

# caddy acts as reverse proxy and static file server

```

```

# You can adjust the ports that the proxy listens on via the .env file
# https://hub.docker.com/_/caddy
inventree-proxy:
  container_name: inventree-proxy
  image: caddy:alpine
  restart: always
  depends_on:
    - inventree-server
  ports:
    - ${INVENTREE_HTTP_PORT:-80}:80
    - ${INVENTREE_HTTPS_PORT:-443}:443
  env_file:
    - .env
  volumes:
    - ./Caddyfile:/etc/caddy/Caddyfile:ro,z
    - ${INVENTREE_EXT_VOLUME}/static:/var/www/static:z
    - ${INVENTREE_EXT_VOLUME}/media:/var/www/media:z
    - ${INVENTREE_EXT_VOLUME}:/var/log:z
    - ${INVENTREE_EXT_VOLUME}:/data:z
    - ${INVENTREE_EXT_VOLUME}:/config:z
portainer:
  image: portainer/portainer-ce:latest
  container_name: portainer
  restart: unless-stopped
  #networks:
  #- dockerlab    # ← Same network as InvenTree
  ports:
    - "9000:9000"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - /home/dockerlab/portainer/data:/data
pgadmin:
  image: dpage/pgadmin4:latest
  container_name: pgadmin
  restart: unless-stopped
  #networks:
  #- dockerlab    # ← Same network as InvenTree
  ports:
    - "5050:80"
  environment:

```

```
PGADMIN_DEFAULT_EMAIL: rarcayan@gmail.com # adjust if different
PGADMIN_DEFAULT_PASSWORD: SecurePassword! # adjust if different
volumes:
  - /home/dockerlab/pgadmin:/var/lib/pgadmin
```

```
#networks:
#dockerlab:
#driver: bridge
```

Caddyfile

```
# Example Caddyfile for InvenTree
# The following environment variables may be used:
# - INVENTREE_SITE_URL: The upstream URL of the InvenTree site (default: inventree.local
host)
# - INVENTREE_SERVER: The internal URL of the InvenTree container (default: http://invent
ree-server:8000)
#
# Note that while this file is a good starting point, it may need to be modified to suit your s
pecific requirements
#
# Ref to the Caddyfile documentation: https://caddyserver.com/docs/caddyfile

# Logging configuration for Caddy
(log_common) {
  log {
    output file /var/log/caddy/{args[0]}.access.log
  }
}

# CORS headers control (used for static and media files)
(cors-headers) {
  header Allow GET,HEAD,OPTIONS
  header Access-Control-Allow-Origin *
  header Access-Control-Allow-Methods GET,HEAD,OPTIONS
  header Access-Control-Allow-Headers Authorization,Content-Type,User-Agent

  @cors_preflight{args[0]} method OPTIONS
```

```

handle @cors_preflight{args[0]} {
    respond "" 204
}
}

# The default server address is configured in the .env file
# If not specified, the proxy listens for all http/https traffic
# If you need to listen on multiple addresses, or use a different port, you can modify this section directly
{$INVENTREE_SITE_URL:"http://, https://"} {
    import log_common inventree

    encode gzip

    request_body {
        max_size 100MB
    }

    # Handle static request files
    handle_path /static/* {
        import cors-headers static

        root * /var/www/static
        file_server
    }

    # Handle media request files
    handle_path /media/* {
        import cors-headers media

        root * /var/www/media
        file_server

        # Force download of media files (for security)
        # Comment out this line if you do not want to force download
        header Content-Disposition attachment

        # Authentication is handled by the forward_auth directive
        # This is required to ensure that media files are only accessible to authenticated users
        forward_auth {$INVENTREE_SERVER:"http://inventree-server:8000"} {
            uri /auth/

```

```

    }
}

# All other requests are proxied to the InvenTree server
reverse_proxy {$INVENTREE_SERVER:"http://inventree-server:8000"} {

    # If you are running behind another proxy, you may need to specify 'trusted_proxies'
    # Ref: https://caddyserver.com/docs/json/apps/http/servers/trusted_proxies/
    # trusted_proxies ...
}
}

```

.env

```

# InvenTree environment variables for docker compose deployment
# For a full list of the available configuration options, refer to the InvenTree documentation:
# https://docs.inventree.org/en/stable/start/config/

# Specify the name of the docker-compose project
COMPOSE_PROJECT_NAME=inventree

# InvenTree version tag (e.g. 'stable' / 'latest' / 'x.x.x')
INVENTREE_TAG=stable

# InvenTree server URL - update this to match your server URL
#INVENTREE_SITE_URL="http://inventree.localhost"
INVENTREE_SITE_URL="http://192.168.1.7:8000" # You can specify a local IP address here
#INVENTREE_SITE_URL="https://inventree.my-domain.com" # Or a public domain name
(which you control)

# InvenTree proxy forwarding settings
INVENTREE_USE_X_FORWARDED_HOST=True
INVENTREE_USE_X_FORWARDED_PORT=True
INVENTREE_USE_X_FORWARDED_PROTO=True

# Specify the location of the external data volume
# By default, placed in local directory 'inventree-data'
INVENTREE_EXT_VOLUME=/home/dockerlab/inventree/data

# Ensure debug is false for a production setup

```

```
INVENTREE_LOG_LEVEL=WARNING

# Enable custom plugins?
INVENTREE_PLUGINS_ENABLED=True

# Run database migrations automatically?
# Note: This does not negate the need to run "invoke update"
INVENTREE_AUTO_UPDATE=True

# InvenTree superuser account details
# Un-comment (and complete) these lines to auto-create an admin account
INVENTREE_ADMIN_USER=admin
INVENTREE_ADMIN_PASSWORD=SecurePassword!
INVENTREE_ADMIN_EMAIL=rarcayan@gmail.com

# Database configuration options
# DO NOT CHANGE THESE SETTINGS (unless you really know what you are doing)
INVENTREE_DB_ENGINE=postgresql
INVENTREE_DB_NAME=inventree
INVENTREE_DB_HOST=inventree-db
INVENTREE_DB_PORT=5432

# Database credentials - These should be changed from the default values!
# Note: These are *NOT* the InvenTree server login credentials,
# they are the credentials for the PostgreSQL database
INVENTREE_DB_USER=pguser
INVENTREE_DB_PASSWORD=SecurePassword!

# Redis cache setup
# Refer to the documentation for other cache options
INVENTREE_CACHE_ENABLED=True
INVENTREE_CACHE_HOST=inventree-cache
INVENTREE_CACHE_PORT=6379

# Options for gunicorn server
INVENTREE_GUNICORN_TIMEOUT=90
INVENTREE_GUNICORN_PORT=8000
```