# Homebox Inventory Management

- Install and Configure Homebox

# Install and Configure Homebox

There are a lot of great open source options for inventory management out there.  I want to start with one that I found recently called Homebox.  It is really a great conceptual bit of software, and it has some really great features that can make it very easy to get started with inventory management.

When I think of home inventory, I think of a list of the items I have in my home, from furniture, to electronics, to photo-albums, antiques, and so much more.  When you think about why you might want to catalog these items, consider the worst case scenario where you have a destructive occurrence such as a fire, flood, hurricane, or tornado.  Perhaps its a disastrous criminal occurrence where you are burglarized, and everything is taken or destroyed.

Your insurance company is going to want a list of all of your assets. They don't want to just write you a giant check when all you owned was a couple of bricks, a plank of wood, and a television set.  You want to be able to catalog the items you had, the value to replace them, and serial numbers for the law enforcement agencies to try and track them down.  These days, they can enter the serial numbers into a database that pawn shops can subscribe to in order to save themselves money. The pawn shop can identify something as stolen before lending money or buying an item.

Let's get down to the nuts and bolts of this system, and get it setup.

## What You'll Need

- Docker-CE and Docker-Compose installed on a system where you want to run your inventory server.
- (optional) NGinX Proxy Manager (or a reverse proxy of your choice) to get outside access to your server from the internet.
- (optional) A domain name that you can add / edit A-records for.
- About 20 Minutes of your time.

# Installing Docker-CE, Docker-Compose, and NGinX-Proxy-Manager

# Installation 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.

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.

## Installing Homebox in Docker

Now that you have Docker and Docker-Compose installed, you'll want to setup a folder structure that will be very handy for any future applications you may want to run in docker on this same machine.

First we'll create a parent "docker" directory, and we'll want a "homebox" directory inside of that.

```
mkdir -p docker/homebox
```

This command looks to see if the "docker" folder already exists, and if it does, it just creates the "homebox" folder inside of it. If it doesn't, then it creates both folders at once.

Now, we want to move into our new "homebox" directory, and create our "docker-compose.yml" file.

```
cd docker/homebox
```

```
nano docker-compose.yml
```

Now that you have the nano text editor open, you'll copy the docker-compose yaml text from the box below, and paste it into your new file.

```
version: "3.4"

services:
  homebox:
    image: ghcr.io/hay-kot/homebox:latest
    container_name: homebox
    restart: unless-stopped
    environment:
    - HBOX_LOG_LEVEL=info
    - HBOX_LOG_FORMAT=text
    - HBOX_WEB_MAX_UPLOAD_SIZE=10
    - HBOX_OPTIONS_ALLOW_REGISTRATION=true
    volumes:
      - ./homebox-data:/data/
    ports:
      - 3100:7745

volumes:
   homebox-data:
     driver: local
```

From the yaml code above, know that it is space (not tab) sensitive. You must use spaces with yaml files, and it is space dependent, which means the number of spaces before an item is important.

If you need to change anything, it will be the port number of the left side (left side only) of the port mapping.  Should the server (machine) you are using already have port 3100 in use, then change that number to a port number not in use on the host machine.  If you change it, keep that number in mind as we move forward in the tutorial.

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

Now, we just need to run the docker-compose file, and get our application up and running.  We'll use two commands combined with the '&&' symbols in order to save a little effort. First we'll run the 'up' command to bring up the server, then we'll runt he 'logs' command to view the logged output and make sure we don't see any errors.

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

If you don't see any errors in the output after you see the 'done' message, then your server should be up and ready to access.  Use the CTRL + C hotkey combination to stop viewing the log output.

Now open your favorite modern browser, and navigate to the IP address of your host machine, and the port you put on the left side of the port mapping in the docker-compose.yml file.  In my case I left it as 3100, so I went to:

`http://192.168.10.42:3100`

**You would, of course, want to use your host machines IP address.**

**You should be greeted with a login page, and you should see a 'Register' button on that page.  Click 'Register', and fill out the form to create the first (admin) account.  Now, if you don't want to allow anyone else to register, you can go back to the "docker-compose.yaml" file, and make a simple change to remove the Register button.**

**Go back to your terminal and enter the command**

`nano docker-compose.yml`

**Once in the editor, arrow down tot he line that says**

`- HBOX_OPTIONS_ALLOW_REGISTRATION=true`

and change that line to read

`- HBOX_OPTIONS_ALLOW_REGISTRATION=false`

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

Now run the command:

`docker-compose up -d`

Running the 'docker-compose up -d' command while the container is running does a few things.

1. It stops the docker-container briefly.
2. Checks for any new configurations or images available to run against (especially useful when using the "latest" tag)
3. Starts the container up with the new configurations / images.

This all happens very fast, and is a really easy way to make changes / updates with very little downtime.

Now, if you go back to your web-page, 'Sign Out' (bottom-left), then refresh, you'll notice the 'Registration' button is gone.  Log back in, and check out the Homebox user interface. It's pretty straight forward to use.  You can create new locations and labels (tags) for your inventory, and you can even create a tree structure for your locations by putting multiple sub-locations under a parent level location.  This can be especially useful for multiple locations that are physically separate, or for businesses with more than one office, or even for houses that have several rooms that are all bedrooms, but you want to identify which bedroom has which items in it.

Additionally, check out the Import functions available under the settings of the application.  You can create a spreadsheet, save it as a CSV (comma separated value) file, and import your items in bulk.  Sometimes the spreadsheet is the right starting place as it provides the ability to copy / paste with similar items, as well as a way to more quickly add items with out a lot of clicks to constantly start a new form.

You are now up and running on Homebox.  I hope you'll like it, and definitely keep an eye out for other great inventory management solutions coming in the future!

# Support my Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://patreon.com/awesomeopensource