

# Easy Appointments Scheduling - Multiuser

<https://www.youtube.com/embed/SfdGwj4ZHLQ>

## Setup a Non Root User

```
adduser <username>
```

Enter your desired password for this new user, then confirm it.

Feel free to enter any of the other requested info, then accept the information entered.

Now make your user part of the 'sudo' group.

NOTE: If you are using a RedHat, CentOS, Fedora spin, then add them to the 'wheel' group instead.

```
`usermod -aG sudo <username>`
```

Now you can log out from root, and log back in as the non-root user.

## Install Docker and Docker Compose

Next we'll install Docker-CE and Docker Compose on the system. This can usually be done with a simple one-line command on most Linux distributions. Enter the following:

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

Enter your super user password if prompted, then allow the installation to complete.

Next, let's add your user to the 'docker' group.

```
sudo usermod -aG docker <username>
```

Now, log out and back in, and you'll be able to run all the `docker` and `docker compose` commands without needing to use `sudo` .

# Setup compose.yaml file

Now we need to create a simple folder structure, and then a file called "compose.yaml".

First, we'll make a folder for our docker container and data to be stored in.

```
mkdir -p docker/easyappt
```

This command creates the 'docker' folder if it doesn't already exist, then it will create the 'easyappt' folder inside of that 'docker' folder.

Now we'll move into the folder we just created:

```
cd docker/easyappt
```

Let's create our 'compose.yaml' file inside of it.

```
nano compose.yaml
```

Copy the code block below, and paste it into your new 'compose.yaml' file.

You'll need to make a few changes in the code block after you paste it. I have marked the placeholder items to be changed by surrounding them with less than '<' and greater than '>'.

It was pointed out that I didn't discuss how the application emails you, or the person scheduling your time. It appears it's pre-configured for php-mail. That said, you can setup and use your smtp email server if you have one, or have the information for one. (e.g. I use Purelymail, so I setup that one for my instance). I'll add a second compose.yaml file below to show the changes needed for email via smtp if you'd like to set that up).

```
---
services:
  easyappointments:
    image: alextsellegidis/easyappointments
    environment:
      - BASE_URL=<http://sched.yourgreatdomain.org>
      - DEBUG_MODE=TRUE
      - DB_HOST=mysql
      - DB_NAME=easyappointments
      - DB_USERNAME=easyappt
      - DB_PASSWORD=<longstR0n6Pa5sw0rdTh4at1sUn1qu3>
    ports:
      - '8082:80' # you can change the left side port if needed.
```

```

# extra_hosts:
# - "<your.freeipa.server>:<your-server-local-ip"
mysql:
  image: mysql
  volumes:
    - ./data/mysql:/var/lib/mysql
  environment:
    - MYSQL_USER=easyappt
    - MYSQL_PASSWORD=<longstR0n6Pa5sw0rdTh4at1sUn1qu3>
    - MYSQL_ROOT_PASSWORD=<it-must-be-some-other-diffErent-paSsword>
    - MYSQL_DATABASE=easyappointments

```

In the above, make sure to change the password for the mysql user and root mysql user to something long and strong.

Also, if you need to use a port other than 80 on your server to reach the application, you can change the left side port number on the port mapping.

Finally, if you plan to use LDAP authentication, you may need to uncomment the lines for `extra_hosts:` and the line below it, filling in your ldap server hostname and domain and ip address appropriately.

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

## For Email via SMTP in the YAML

```

---
services:

  easyappointments:
    image: alextseligidis/easyappointments:latest
    restart: unless-stopped
    ports:
      - '8082:80'
    environment:
      - BASE_URL=https://sched.yourgreatdomain.com
      - DEBUG_MODE=FALSE
      - DB_HOST=mysql
      - DB_NAME=easyappointments
      - DB_USERNAME=root
      - DB_PASSWORD=<a-sup3r-complex-lon6-pas5woRd>

```

```
- MAIL_PROTOCOL=smtp
- MAIL_SMTP_DEBUG=0
- MAIL_SMTP_AUTH=1
- MAIL_SMTP_HOST=<smtp.your-email-provider.org>
- MAIL_SMTP_USER=appointments@yourgreatdomain.org>
- MAIL_SMTP_PASS=<a-l0ng-str0n6-pa5sw0Rd-for-your-email>
- MAIL_SMTP_CRYPT0=tls # might need to be ssl
- MAIL_SMTP_PORT=587 # if ssl, use 465
- MAIL_FROM_ADDRESS=<appointments@yourgreatdomain.org>
- MAIL_FROM_NAME=Appointment
- MAIL_REPLY_TO_ADDRESS=<appointments@yourgreatdomain.org>
extra_hosts:
  - "<ldap.your-ldap-domain.local>:<ip.for.ldap.server>"
mysql:
  image: mysql:8.0
  restart: unless-stopped
  environment:
    - MYSQL_ROOT_PASSWORD=<a-sup3r-complex-l0n6-pas5woRd>
    - MYSQL_DATABASE=easyappointments
  volumes:
    - ./mysql:/var/lib/mysql
```

In the above, you need to change the same information, surrounded by less than '<' and greater than '>' symbols. You also need to make sure to fill the proper details for your SMTP email provider. You can test the email is working by doing a 'Forgot Password' in the login screen. Enter the requested details, and it should successfully send you the new temporary password. This will be done, of course, after the initial setup screen has been completed.

## Start our Docker Application

In the terminal enter the command:

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

The first part tells docker to bring up the containers in the background (detached or as a daemon), and the second part tells docker we want to follow the logs as they are generated while the application(s) start up.

The logs will likely scroll by very quickly, really you're just watching to see if you see any errors that may scroll past. We can always stop the logs and then scroll back up to read any errors if we see any. You shouldn't, but it does happen at times.

Once the logs slow down a bit, we can stop following them with CTRL + C.

Now let's go to our favorite modern browser, and open the IP and port we set for the application.

In my case I'll go to IP address 192.168.10.46, you'll likely have a different IP for your service. If you changed the left side of the port mapping in the compose.yaml file, then you'll want to put that after the IP in this form:

<http://192.168.10.46:8020> (for instance).

You should be greeted with the EasyAppointments first run wizard. We don't want to run through the wizard just yet, however. First we need to setup a domain or subdomain to point to our application so that our clients can access it from anywhere and schedule time with us.

## Setup a DNS A-record and / or Reverse Proxy Entry

In order to ensure our site name resolves to our server and service we will need a DNS A-record (if you are running this on a server with a public IP address and are running it on port 80, then this will be all you need technically, but since we also want to give it a LetsEncrypt certificate, we'll use a reverse proxy as well) and a Reverse Proxy.

First, you need to own your domain, or at least have access to add DNS records for the domain you are setting up EasyAppointments for. Add an A-Record for the domain or subdomain and tell it to point to the public IP address where your server is running. If it's a VPS, such as on Linode, Digital Ocean, Hetzner, etc, then it likely has it's own public IP. If you're running inside of a local area network, then your ISP may issue a public IP address to your main modem / router, and this is the address you'll want to use.

The record will be entered slightly differently for each registrar, so if you want to call it sched.mygreatdomain.com, you need the following:

Type	Domain / Subdomain	IPv4 Address
A	sched	21.22.23.24

Of course, using your actual public IP address instead of 21.22.23.24.

Once that is set, it may take a few minutes to a few hours to take effect fully, so be patient. In the meantime we'll setup our Reverse Proxy entry. I use NGinX Proxy Manager, but feel free to use any reverse proxy you prefer.

With NGinX Proxy manager, I'll add a new entry for sched.mygreatdomain.com, then I'll enter my private IP address (the LAN address of the server where I'm running the EasyAppointments application, and the port (in my case 80, but remember to set the right port if you changed it in the compose.yaml file). Next, I'll tick the options for **Block Common Exploits**, and **Websocket Support**.

Finally, we'll move to the SSL tab. We'll change **None** to **Request a New Certificate**. Next tick the options for **ForceSSL**, **HTTP/2 Support**, and both **HSTS** options.

Accept the Terms of Service, enter your email. Before you click Save, you need to be sure your A Record is now functional and pointint to your external IP address. You also need to have forwarded port 80 and 443 through your firewall / router to the server / machine where you are running your reverse proxy.

Request for sched.mygreatdomain.com --> DNS A Record points to 21.22.23.24 --> Router Firewall points to Reverse Proxy at 192.168.1.25 inside your LAN --> Reverse Proxy sends request to 192.168.1.55:8020 inside your LAN and your user is presented with Easy Appointments.

If we have setup everything correctly, we'll be presented with the Easy Appointments First Run Wizard. Fill in the form as completely as you can. Click Next. Let the system setup all the database and new entries, and then you'll be ready to configure Easy Appointments.

I go through the setup in detail in the video, so be sure to check that out for initial configurations.

## Plugins

EasyAppointments has several plugins in the 'Settings' area. You'll want to check those out. There may be one or more that you will find useful. I used the LDAP plugin to setup my FreeIPA system to allow Single Sign On for my Providers (what they call an employee essentially). I also setup calendar syncing with CalDav from NextCloud so as I add events they are synced to a specific NextCloud calendar, (but this will work with any CalDav), and when I add other appointments to my NextCloud calendar, they will sync to my EasyAppointments and block out that time.

Make sure to check out the video for more details on how to set this up. Be aware, when using OpenLDAP, or other LDAP systems, you should be familiar with your LDAP needs in order to properly configure your LDAP. If you are using FreeIPA, you can likely do the same as I did.

When setting up Caldav, you should also test first, then set it up for all your providers, or provide them detailed instructions on how to do it. It wasn't complicated, but testing first to make sure things work as expected is important.

