

DocuSeal

[Pi-hole website](#)

Let's create a folder for our Docuseal install. I like to create a parent level folder for 'docker', then put each docker application within that folder.

```
mkdir -p docker/docuseal
```

Now we'll move into that folder, and create a new compose.yaml file where we'll define how we want Docuseal to be configured.

```
cd docker/docuseal
```

```
nano compose.yaml
```

Inside the new compose.yaml file, we need to add the block of yaml code below.

```
services:
  docuseal:
    depends_on:
      postgres:
        condition: service_healthy
    image: docuseal/docuseal:latest
    ports:
      - 3000:3000
    volumes:
      - ./data:/data
    environment:
      #- FORCE_SSL=${HOST}
      - DATABASE_URL=postgresql://postgres:this-needs-to-be-a-long-strong-
password@postgres:5432/docuseal

  postgres:
    image: postgres:15
    volumes:
      - './pg_data:/var/lib/postgresql/data'
    environment:
      POSTGRES_USER: postgres
```

```
POSTGRES_PASSWORD: this-needs-to-be-a-long-strong-password # <-- should match the
password in the DATABASE_URL above
POSTGRES_DB: docuseal
healthcheck:
  test: ["CMD-SHELL", "pg_isready -U postgres"]
  interval: 5s
  timeout: 5s
  retries: 5
```

In the file, you may need to make a change to the port mapping. `3000` is a very common port for many applications. If you already have an application using port `3000` on your host machine, you can change the port number on the left side of the port mapping. In my case I changed it to `8022`, so that line looked like

```
- 8022:3000
```

Next, in two places, there is a password being set. First in the `DATABASE_URL` environment variable in the `docuseal` section. I have it defaulted to "this-needs-to-be-a-long-strong-password", but you should change this to an actual long, strong password. Once set, you need to copy it, and paste it in as the value for `POSTGRES_PASSWORD` in the `postgres` section. Those two passwords must match for the application to function.

Finally, I have commented out the environment variable line for `- FORCE_SSL=${HOST}`, but if you are setting this up without a reverse proxy, then in the original file found here, you will see a 'caddy' section as a built in reverse proxy. You will need to uncomment that line by removing the hash tag, and then make sure the 'caddy' section is added back.

Once you've made those changes, you can save your `compose.yaml` file with `CTRL + O`, then press `Enter` to confirm, and use `CTRL + X` to exit.

We are ready to start up the application. We can do that with the command:

```
docker-compose up -d
```

Revision #4

Created 2025-03-13 07:10:04 UTC by Johan Odendaal

Updated 2025-03-13 07:20:15 UTC by Johan Odendaal