Install and Setup Docuseal

https://www.youtube.com/embed/AFSxjJ4nyDI

If you've ever had to sign an electronic document in the last 10 years, you've likely been sent something through Docusign. It's a really great service, and provides some great tooling to ensure that your electronic signature or initials are all on the document, and in the right places.

Docuseal is an open source, and powerful alternative to Docusign, and I wanted to go through getting this setup and running in your open source lab, business, or MSP if you're following along on that series. This is a huge thing to have when selling services, so you can detail what service you'll provide, and getting client sign off that they understand and agree.

What you'll need

- A server or machine to host the service
- Docker-CE installed on that machine (I have a script to help with this)
- (optional) Reverse Proxy so you can setup your own domain name / url for the service (required for external users to your network)
- About 20 minutes of your time

Install Docker-CE and Reverse Proxy 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 -O install-docker.sh 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.sh`

and then run the script with the command:

`./install-docker.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. In this case, you may want to answer 'y' to NginX Proxy Manger if you don't already have a reverse proxy running for your services. If you just want to use Docuseal on your local network to start, then the reverse proxy isn't necessary.

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.

Install Docuseal

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 confrim, 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
```

This will pull down the necessary images, and startup the containers as defined in the yaml file.

Once back at a normal terminal prompt, you can open your favorite modern browser, and go to the IP address of the host machine you've setup docuseal on, or if you are setting it up on the machine you are working on directly, you can enter "localhost", then put a colon ":", and after that enter the port number you entered on the left side of the port mapping in the compose.yaml file.

In my case I went to http://192.168.10.157:8022.

You should be greeted with the initial setup screen for Docuseal.

You should STOP here! Please consider whether you want to access this instance of Docuseal from your local network only, or whether you want to use a domain name / URL. If you want to use a domain / URL, you need to first setup your A-record for DNS (a record that tells DNS at what public IP address your domain name should resolve), and / or your reverse proxy.

In my case, I own the domain sysmainit.com, so I have setup a DNS A-record that tells DNS servers to go to the public IP address of my server whenever anyone enters docuseal.sysmainit.com in their browser. Once it reaches the server, I have an entry in my reverse proxy that routes the request to the correct private IP inside my local network.

browser ----> Request DNS (docuseal.sysmainit.com) ----> public IP of server ----> reverse proxy ----> server / port running docuseal

Check out the accompanying video to see how to setup a reverse proxy entry easily using NGinX Proxy Manager.

Once you've setup that up, make sure you set your page to use SSL with a proper certificate (e.g. from LetsEncrypt). Next, make sure you can reach your site using the https version of your URL without any certificate warnings.

Once at the initial setup page with the proper https URL, start working through the setup Wizard, as this will default the site URL to the domain name you want.

You're done. Start using Docuseal by setting up your outgoing email, notifications, settings, and uploading your first PDF to get signed.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Buy me a Beer / Coffee:
https://paypal.me/BrianMcGonagill

Revision #2
Created 2024-07-03 19:32:52 UTC by Brian McGonagill
Updated 2024-07-04 15:18:28 UTC by Brian McGonagill