

Running Clowder using Docker

Clowder consists of many pieces of software working together. Starting a new instance of clowder can be complex, especially to get all the pieces to work together. We highly recommend for users to use docker to get started. In docker all the components needed are packaged as containers. A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings. Using the containers makes it easier for the user to get started since all the software needed to run a piece is part of the container, no need to install libraries or additional software. To control the orchestration between these containers we leverage of docker compose. This will start the containers in the right order and makes sure the containers can talk with each other.

Installing Docker

You will need to have docker installed on your machine. You can find more information on how to get docker installed at: <https://docs.docker.com/install/>

You will also need to install docker-compose, this will be installed as part of docker on the Mac and Windows, for linux this is a separate install, see <https://docs.docker.com/compose/install/>

Starting Clowder

First download the clowder docker-compose file, <https://github.com/clowder-framework/clowder/blob/master/docker-compose.yml>. This file should allow you to get a complete docker environment up and running, including some extractors (audio, video, image, pdf). You do not need to make any changes to the docker-compose file, if you want to secure the system more, you can disable the ports used for rabbitmq, mongo and elasticsearch. We have left them open for developers but they will be mapped to random ports (see docker tips). The only port that needs to be mapped is port 9000 of clowder, which in the docker-compose file is mapped to port 9000 in the docker machine.

To start clowder you can run docker-compose up. This will show all the log files of clowder, if you want to run this in the background you can use docker-compose up -d. After about 1 minute clowder should be up and running and you can connect to the clowder instance at <http://localhost:9000/> (assuming you run it on your local machine).

When starting clowder using docker-compose you will see an error "WARNING: Some services (....) use the 'deploy' key, which will be ignored. Compose does not support 'deploy' configuration - use 'docker stack deploy' to deploy to a swarm." You can safely ignore this error, the docker-compose file can also be used to deploy clowder in a docker swarm or using later versions of docker in kubernetes (this however has not been tested).

Persistent data

The docker-compose file uses named volumes in docker. You can either create these volumes before you run docker-compose the first time, or you can use the volumes created by docker-compose. The volumes will need to be named the same way as the containers (see docker tips). When restarting docker will reuse these volumes making the data persistent across runs.

Creating Accounts

At this point you can create an account using the signup link. You will need to provide an email address when you sign up, this will be your account name. The docker system is not setup to send emails, you can see the email that would be send in the log files of clowder (see docker tips). In the logfiles you will see a line with "application - MOCK MAILER: send email", followed by some more text, and a link (usually something like http://localhost:9000/signup/large_random_string). If you copy paste this link in your browser you can create a new account. To create an account that has administrators rights you can by default use the admin@example.com email address.

Clowder Customization

There are many configuration values that you can override, for example the default admin email address. To do this you will need to edit the custom.conf file used by clowder. This file is located in the clowder-custom volume. You can either change the content of this folder directly (use docker volume inspect clowder_clowder_1 to find the actual location on disk), or you can exec into the running docker container (docker exec -ti clowder_clowder_1 /bin/bash) and goto the folder /home/clowder/custom. You can edit the custom.conf file to fit your needs. Once you finish making the changes you can exit the container using exit, this will not stop the container. You will need to restart the clowder container, which can be done using docker restart clowder_clowder_1.

Another way is to make any changes before you start the full stack. You can run just the clowder instance and make changes using: `docker-compose run --no-deps clowder /bin/bash` this will start just the clowder container and give you a bash prompt. You can make changes to the custom.conf and play. plugins and exit. If you now start the full stack it will use these updated changes when starting clowder.

A list of all values that can be changed can be found at <https://github.com/clowder-framework/clowder/blob/master/conf/application.conf>.

Docker Tips

When starting clowder using docker-compose up, the containers will be named <folder name>_<service>_1, for example if you download the docker-compose file in your clowder folder, the clowder container will be called clowder_clowder_1.

You can see a list of all docker containers that are running as part of the docker compose file using docker-compose ps, this will also show any port mappings that are done, allowing you to see what ports for example are used for rabbitmq.

To see the logfiles for a specific container, especially useful if you started docker-compose up -d, you can use docker logs <container name>, for example to see the log files for clowder you can use docker logs -f clowder_clowder_1 (this will follow the log file, to stop this you need to press ctrl -c).

To see the volumes and the actual location on disk you can use docker volume ls, as well as docker volume inspect. Both windows and mac will use a virtual machine, so the volumes will be located in that virtual machine.

If you run clowder using docker-compose up, you will have all the log files in the console where you started clowder. If you use docker-compose up -d, all containers will run in the background. You can