
CoSS Documentation

Release 0.1.1

Participation Systems

Jun 26, 2018

Contents

1	Installation	3
1.1	Dependencies	3
1.2	Building CoSS	4
1.3	Populate database	4
1.4	Running coss	5
2	Indices and tables	7

Community Support Software [\[Source Code\]](#)

Contents:

CoSS development environment can be installed using **docker**. This way we run the web app and all its dependencies as docker containers. [Here](#) you can find more info about what docker is.

1.1 Dependencies

1. You need to install docker in your system. The [installation guide](#) covers many operating systems but for now we only support Linux and Mac OS X. *Version required: 1.3.1 or higher.*
2. We are using an orchestration tool for docker called [docker-compose](#) that helps us automate the procedure of initiating our docker containers required for development. Installation instructions can be found in [Compose's documentation](#). *Version required: 1.0.1 or newer.*

1.1.1 Running Docker on Mac

Here are some notes for running Docker on Mac.

- Docker cannot run natively on Mac because it is based on a Linux kernel specific feature called LXC.
- When running docker in Mac via **boot2docker** you are running a lightweight Linux VM in Virtualbox that hosts the docker daemon and the LXC containers.
- We are running docker client in our host system that connects to the docker daemon inside boot2docker VM.
- We are using docker's *volume sharing* feature in order to share the source code with the Coss container. This is not directly supported in Mac. As a workaround boot2docker implements this feature by sharing the folder with Virtualbox first.
- The extra layer that we are adding using Virtualbox might cause some performance issues. This is a trade-off for having an easily reproducible stack without installing everything manually.

More information regarding boot2docker can be found in [the documentation](#).

Here are some extra steps in order to run CoSS on Mac:

1. Make sure *boot2docker* is initialized:

```
$ boot2docker init
```

2. Make sure *boot2docker* VM is up and running:

```
$ boot2docker up
```

3. Export *DOCKER_HOST* variables using the following command:

```
$ $(boot2docker shellinit)
```

Note: You need to make sure to run `$(boot2docker shellinit)` in each new shell you are using, or export it globally in order not to repeat this step every time you are working on CoSS.

1.2 Building CoSS

You only need to follow these steps once.

1. Fork the main [CoSS repository](#).
2. Clone your fork to your local machine:

```
$ git clone git@github.com:YOUR_USERNAME/coss.git coss
(lots of output - be patient...)
$ cd coss
```

3. Configure your local coss installation:

```
$ cp env-dist .env
```

4. Start PostgreSQL container:

```
$ docker-compose up -d db
```

5. Build the app container (this will take some time):

```
$ docker-compose build web
```

1.3 Populate database

You only need to follow these steps once.

1. Create the database tables and run the migrations:

```
$ docker-compose run web python manage.py migrate --noinput
```

2. Create a superuser:


```
$ docker-compose run web python manage.py createsuperuser

#. Load http://127.0.0.1:8000 or (for Mac users only) ``<IP>:8000`` where ``<IP>
→`` is the one returned by ``boot2docker ip`` command.
#. Stop the server with ``Ctrl^C``.
```

1.4 Running coss

1. Run coss:

```
$ docker-compose up web
```

2. Develop!

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`