https://app.titra.io/try

Docker Guide

Deploying titra with Docker

You will need a **MongoDB container** to use titra. Here is an example to get started:

```
docker run --name mongodb -p 27017:27017 mongo
```

> Please note that this basic approach will store all database data in a transient Docker volume – this is not a good approach to run MongoDB in production! Checkout this guide for a more sophisticated approach on how to get started with running MongoDB in Docker.

Once the database is ready, titra needs to be aware of that container, an easy way to achieve this is using **–link**:

```
docker run --name titra -p 3000:3000 --link mongodb -e MONGO_URL=mongodb://mongodb/titra -e
ROOT_URL=http://localhost:3000 kromit/titra
```

In addition to the manual steps above, you can also use docker-compose to deploy the whole stack:

```
curl -L https://github.com/titraio/titra/raw/master/docker-compose.yml |
ROOT_URL=http://localhost:3000 docker-compose -f - up
```

> Please note that our standard Docker compose file does not map a volume for the database files and thus everything is stored in a transient Docker volume.

Congratulations! titra should now be up and running at http://localhost:3000

Persisting MongoDB database files

As mentioned above, the MongoDB container will fallback to using transient Docker volumes if not specified otherwise. If you are planning to run titra in production you should therefore specify a volume mapping for the container path /data/db. A ready-to-roll production yml file can also be found in the titra GitHub repository which maps the container path to /root/mongodb. This setup is used for the DigitalOcean one-click deployment app.

Revision #3

Created 2021-07-30 06:37:58 UTC

Updated 2026-04-03 03:52:36 UTC by Fabian Kromer