Containerized MADS

Paolo Bosetti

2025-06-02

The base agents for setting up a MADS network are available as a ontainerized environment.

Table of contents

Contents

Setting up a MADS network requires a minimum of a broker, a MongoDB server, and the logger agent. This guide explains the easiest and more portable solution to have them up and running in minuts in a Docker environment.

Prerequisites

You need to have Docker installed on your machine. If you don't have it yet, you can follow the official installation guide. Follow that guide thoroughly, and make sure you can run this command in your terminal:

```
docker run hello-world
```

Running the MADS network

The MADS Network is actually available as a **compose** network of three containers: the broker, the MongoDB server, and the logger agent. Docker has a `docker compose` command that allows to build and deploy multiple, synchronized containers properly sharing network communications.

To find more on `docker compose`, have a look at the documentation.

Step 1. Clone the repo

All you need is in the MADS_container repository. You can clone it with the following command:

```
git clone --depth 1 https://github.com/MADS-NET/MADS_container.git
```

Step 2. Build contaners

In the `MADS_container` directory, you can build the containers with the following command:

```
docker compose build
```

> **Note**
>
> This is only to be done once.

Step 3. Run the containers

You can now run the containers with the following command:

```
docker compose up
```

This starts the three containerized processes: MogoDB, the broker, and the logger agent **in terminal attached mode**: this means that you can read the logs and stop all three containers at once by pressing `Ctrl-C`.

Once you are sure that everything works as expected, you may start the containers in **detached mode** with the following command:

```
docker compose up -d
```

> **Note**
>
> You can stop the containers at any time with the command `docker compose down`.

Now the broker and the database are also accessible as if they were processes running on your host machine.