MongoDB Setup Guide

This guide explains how to set up a MongoDB container and configure the database server so they can be used by DISSECT-CF-Fog-WebApp and DISSECT-CF-Fog.

**Creating the MongoDB container**

* Download the [Docker Desktop](https://www.docker.com/) app and install it
* Pull the latest mongodb image using the following command:
  + *docker pull mongo:latest* command (latest version is 6.0.2. at the writing of this)
* The command below creates an image with the name of “dissect”, binds the 27017 port to the container’s 27017 port and runs the container:
  + *docker run -d -p 27017:27017 --name test-mongo mongo:latest*

**Configuring the MongoDB server**

In order to use DISSECT-CF-Fog and DiSSECT with MongoDB, it is necessary to create a database called "dissect", create the corresponding collections, upload files using GridFS and add their references to some collections.

Collections that has to be created:

* users: contains the users
* strategies:
  + Contains the different strategies that can be used during the configurations and simulations. For example: *Application-strategies.xml* and *Device-strategies.xml*
* simulator\_jobs:
  + Contains simulation jobs that can be processed by the simulator
* resources:
  + Contains the resources that can be used during the configurations and simulations
  + For example: *LPDS\_32.xml* and *LPDS\_16.xml*
* providers:
  + Contains the providers in the form of 1 or more xml files. For example: *providers.xml*
* fs.files:
  + One of the GridFS tables.
  + Registers the files that can be rebuilt from the chunks in the fs.chunks collection.
* fs.chunks:
  + The other table used by GridFS.
  + It contains the actual contents of the files broken down into chunks
* configurations:
  + Contains configurations submitted by users.
  + The *jobs* property of the documents in this table reference simulations from the simulator\_jobs

Files that has to be uploaded to the database using GridFS and their references need to be saved to the following collections:

* Providers: *providers.xml*
* Resources: *LPDS\_32.xml*, *LPDS\_16.xml*
* Strategies: *Application-strategies.xml*, *Device-strategies.xml*

**Configuring the MongoDB server programmatically**

You can use the *mongodb-setup.js* script to configure the MongoDB server programmatically.

In the script, you have to set the connection string as the value of the uri variable. This string specifies the key characteristics of the connection like the name of the database, the username and the password (if there is).

The script connects to the MongoDB server, then it creates the collections, uploads the files and creates the corresponding references of theirs in the right collections.

The files that get uploaded can be found in the *mongodb-setup-resources* folder, relatively to the *mongodb-setup.js* script. In order to be able to run the script, you should run the command *npm install* in the folder of the script, so the dependencies of the script will get installed.

**Notes**

* This script creates a minimalist database setup required to use DISSECT-CF-Fog simulator and the DISSECT-CF-Fog-WebApp. If you want to configure the database differently, feel free to change it.
* As with the script, you need to set the connection string for DISSECT-CF-Fog (*application.properties*) and DISSECT-CF-Fog-WebApp (*mongodb-service.js*).