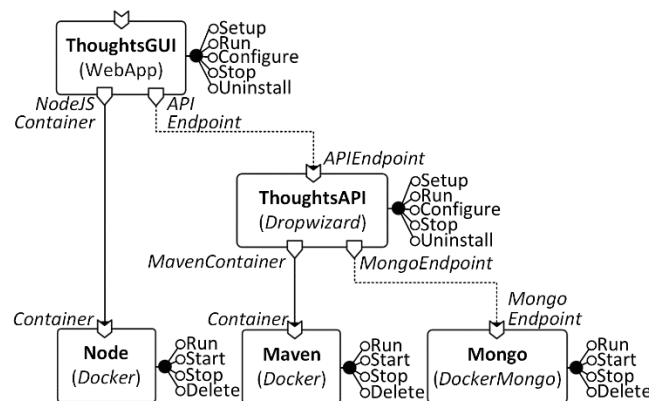


The *Thinking* application

Thinking is an open-source web application that allows end-users to share what they are thinking about, so that all other end-users can read it. It is composed by three main components:

1. an instance of *MongoDB* that is exploited to permanently store the collection of thoughts shared by end-users,
2. *ThoughtsApi*, which is a Dropwizard-based REST API that permits accessing the collection of shared thoughts, and
3. *ThoughtsGui*, which is a web-based graphical user interface that interacts with *ThoughtsApi* to permit retrieving and adding thoughts to the shared collection.

The *MongoDB* instance is obtained by instantiating a *Mongo* Docker container, while *ThoughtsApi* and *ThoughtsGui* are made concrete by hosting them on a *Maven* Docker container and on a *Node* Docker container, respectively.



The resulting application topology is shown above, where each node is also associated with the operations that it offers to manage its lifecycle. To understand how to use such operations and their effects please refer to the online documentation available at <https://github.com/di-unipi-socc/thinking>.

Barrel

You can access a running instance of Barrel at <https://di-unipi-socc.github.io/barrel/>. (Please set “Hard recovery to ON”).

You can download the input needed for running all analyses in Barrel at <https://github.com/di-unipi-socc/barrel/blob/master/examples/thinking-app-with-fault-handlers.csar>.

Test for evaluating the design-time support offered by Barrel

User profile

The test is completely anonymous. We would ask you to kindly fill the below form just for profiling purposes.

Group

Are you in the group using Barrel?

☐ Yes

☐ No

Personal Information

Which is your age?

☐ 21-24

☐ 25-28

☐ 29-32

☐ 32+

Which is your gender?

☐ Male

☐ Female

Which is your current degree?

☐ BSc in Computer Science (or equivalent)

☐ MSc in Computer Science (or equivalent)

☐ PhD in Computer Science (or equivalent)

Experience

How many years of experience as application developer do you have? (Overall, please do not consider periods of inactivity)

☐ Less than one

☐ Between one and three

☐ More than three

Have you ever developed a multi-component application?

☐ Yes

☐ No

How many years of experience as system/application administrator do you have? (Overall, please do not consider periods of inactivity)

☐ Less than one

☐ Between one and three

☐ More than three

Have you ever deployed a multi-component application on a cloud?

☐ Yes

☐ No

Have you ever used containers (like LXC, Docker or rkt containers)?

☐ No, and I do not know what they are

☐ No, but I know them

☐ Yes

Have you ever used configurator management systems (like Chef or Puppet)?

☐ No, and I do not know what they are

☐ No, but I know them

☐ Yes

1. Suppose that we wish to orchestrate the deployment of *Thinking*. Which of the following deployment plans are valid? (If you mark a plan as not valid, please also explain why such plan is not valid.)

Start Time __ : __ – End Time __ : __

Plan 1.1

Run Mongo -> Run Maven -> Run Node -> Setup GUI -> Run GUI -> Setup API -> Config API -> Run API -> Config GUI

Valid? [Y] [N] Why: _____

Plan 1.2

Run Node -> Setup GUI -> Run GUI -> Run Maven -> Setup API -> Run API -> Run Mongo -> Config API -> Config GUI

Valid? [Y] [N] Why: _____

Plan 1.3

Run Node -> Run Maven -> Setup GUI -> Setup API -> Config API -> Config GUI -> Run Mongo -> Run API -> Run GUI

Valid? [Y] [N] Why: _____

Plan 1.4

Run Maven -> Run Node -> Setup API -> Setup GUI -> Run GUI -> Run Mongo -> Config API -> Run API -> Config GUI

Valid? [Y] [N] Why: _____

2. Suppose that an instance of *Thinking* is up and running (viz., all containers are running, and both the API and the GUI are effectively working). Which are the effects of the following management plans?

Start Time __ : __ – End Time __ : __

Plan 2.1

Stop GUI -> Uninstall GUI -> Stop API -> Uninstall API -> Stop Maven -> Delete Maven -> Stop Node -> Delete Node
-> Stop Mongo -> Delete Mongo

Effects: _____

Plan 2.2

Stop API -> Uninstall API -> Setup API -> Config API -> Config GUI -> Run API

Effects: _____

Plan 2.3

Stop Node -> Delete Node -> Stop Maven -> Delete Maven

Effects: _____

3. Suppose that an instance of *Thinking* is up and running (viz., all containers are running, and both the API and the GUI are effectively working). Please answer to the following questions.

Start Time __ : __ – End Time __ : __

3.1 What happens if we stop the *Mongo* Docker container?

3.2 What should we do after stopping *Mongo* to have *Thinking* up and running again (viz., to go back to the situation where all containers are running, and both the API and the GUI are effectively working)?

4. Suppose that, while deploying an instance of *Thinking*, the *API* component unexpectedly crashes (viz., it becomes unresponsive due to an unexpected error, and none of its management operations can be effectively performed). Such instance hence got stuck with the Docker containers running, the *API* crashed, and the *GUI* not installed yet. Can we still complete our installation (by only exploiting the management operations offered by the components of *Thinking*)? Please motivate your answer.

Start Time __ : __ – End Time __ : __
