

Final Submission

Submission

- Upload all Docker Containers to Docker Hub ¹.
- Prepare a Docker Compose file (**Name it like your repository (gxy)**) which starts all Docker containers.
- Tag the final submitted commit, i.e. the one you want us to grade as **submission** and push the tag to the repository. You can do this using the following commands:
 - `git tag -a submission 'Final Submission'`
 - `git push origin -tags`
- Follow the instructions for your specific topic.

General Remarks

Please test your solution using your Docker Compose setup and images hosted on Docker Hub at least once to make sure it all works. Do not assume any preinstalled software on the host machine except for Bash, Docker and Vagrant. Additionally, do not make any changes to the host environment. If you use Vagrant, make sure that your solution does not require preconfigured images - use a Vagrantfile instead.

The main web GUI of your application must be reachable at `http(s)://localhost:8080/` and there must be a single command or script to execute for starting up your entire application, which should be documented in your README file.

Credentials and external services

Many solutions rely on external services that require some application credentials (e.g. API keys). Ideally your solution comes with these credentials, however if for whatever reason you don't want to hand us these credentials (e.g. because you used your personal Twitter or Dropbox account to develop your solution), please make sure you document the following things in your README file:

- Which **external services** your solution relies on (e.g., Indico, Twitter)
- **Locations (absolute paths) to configuration files** to put credentials, if required (e.g., for Twitter, Dropbox), to make your solution work

Contributions

All project members have to contribute in equal parts to the solution. Since not every contribution is equally well visible in the GitLab statistics, we urge you to write into your README file who worked on which parts and which main contributions were made by each group member. If you fail to do so we will be forced to rely solely on the GitLab statistics for your contributions.

¹<https://hub.docker.com/>

Topic specific setup

Topic 1 - Stream Processing

- Provide a Docker Compose file or shell/batch scripts that start
 - the dataprovider
 - the stream processing topology
 - the UI
- Provide a database or text files that are processed by the data provider

Topic 2 - Cloud-based RAID Storage

- Provide a Docker Compose file or shell/batch scripts that start your middleware and UI
- If your solution requires a specific authentication process (e.g., with Box) please document this in your README

Topic 3 - Docker-based Onion Routing

- The setup of your Docker containers should contain a pre-configured version of your originator software that allows us to use the onion routing after all other containers are started by means of a Docker Compose script.

Topic 4 - Docker-based Service Composition

- All services (Twitter service, analysis service, report service and failover service) should already be packaged in a Docker container
- Provide a Docker Compose file that starts the containers, including the service that hosts your BPMN engine and the frontend UI.