

# Installing and populating BIAFLOWS locally

It is possible to install BIAFLOWS on a local server or a desktop computer. This might be useful to manage and analyse images locally. The procedure is described below and should take less than 30 minutes (UNIX based system recommended).

## Installing a local instance of BIAFLOWS

The procedure described in this section is for Linux Ubuntu but it should be possible to install BIAFLOWS on other platforms (not tested). Some specific details related to deployment on Mac OS can be found [online](#).

### 1/ Install requirement

BIAFLOWS runs in Docker containers, the only requirement is to install Docker.

[Check official Docker documentation to install Docker for Ubuntu](#). Choose Install using the repository, set up the repository and install Docker CE.

### 2/ Retrieve BIAFLOWS installation files

```
mkdir Biaflows/  
cd Biaflows/  
git clone https://github.com/Neubias-WG5/Biaflows-bootstrap.git  
cd Biaflows-bootstrap
```

### 3/ Configure the local instance

Edit configuration.sh and, if needed, update URLs (CORE\_URL, IMS\_URL, UPLOAD\_URL). Make sure to use URLs that are not already used by other applications (avoid localhost) to prevent conflicts. In **/etc/hosts** of the host machine, add the following lines, adapting them accordingly to chosen XXX\_URL in configuration.sh.

```
127.0.0.1    biaflows  
127.0.0.1    biaflows-ims  
127.0.0.1    biaflows-upload  
127.0.0.1    rabbitmq
```

If needed, update data path variables (IMS\_STORAGE\_PATH ...). All data paths must be valid and mappable in the Docker engine. If they don't exist, create all the directories (**mkdir**) corresponding to the following variables:

```
IMS_STORAGE_PATH  
IMS_BUFFER_PATH  
FAST_DATA_PATH  
PROXY_CACHE_PATH  
SOFTWARE_CODE_PATH  
SOFTWARE_DOCKER_IMAGES_PATH  
JOBS_PATH  
SERVER_SSHKEYS_PATH
```

A reference to these URLs and paths is provided here:

<https://doc.uliege.cytomine.org/display/PubOp/Cytomine+configuration+reference>

Configure `BIAFLOWS_WORKFLOWS_METRICS` to *true* or *false* depending if you want to perform benchmarking: ground truth annotations are then required for all images. Setting this flag to false is the valid option if you plan to manage and process local images.

#### 4/ Initialize the deployment

Run the installation script: `sudo bash init.sh`

#### 5/ Deploy the local instance

Run the generated deployment script: `sudo bash start.sh`

#### 6/ Check the running instance

When start up is finished, check that the application is running in your browser at the URL specified by `CORE_URL` (default: `http://biaflows`).

Three accounts with different access rights are automatically created (username: admin; password: admin; username: guest; password: guest; username: neubias; password: neubias). Passwords should be updated from the Account page (top right).

#### 7/ Install sample projects (images and ground-truth data)

After BIAFLOWS successfully installed locally, the local instance is empty. All projects available in BIAFLOWS online instance can be imported to the local instance. For this, get the public and private keys of the admin account (Account page), then run:

```
cd Biaflows-bootstrap
sudo bash ./inject_demo_data.sh ADMIN_PUBLIC_KEY ADMIN_PRIVATE_KEY
```

where `ADMIN_PUBLIC_KEY` and `ADMIN_PRIVATE_KEY` have been substituted by their respective values.

The script starts to download projects and import them in your local BIAFLOWS.

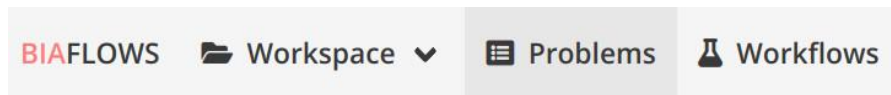
The list of imported projects can be tweaked by editing the file `Biaflows-bootstrap/configs/project_migrator/projects.txt`.

The whole data injection procedure can take several minutes, depending on your Internet connection and the number of projects being imported.

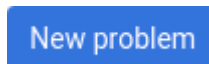
## Creating a new problem (project) in a local BIAFLOWS instance

To create a new problem, connect as regular user or admin.

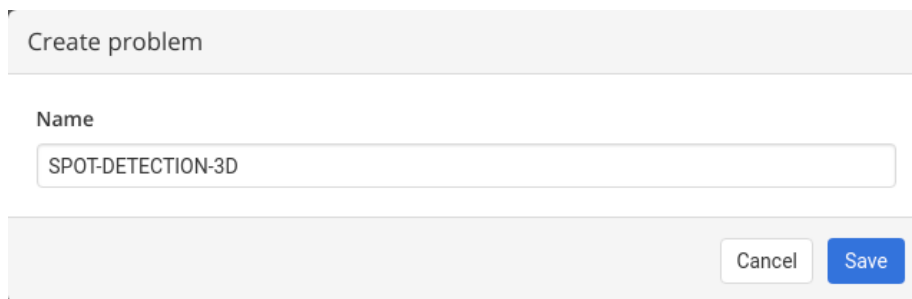
1/ Go to **Problems** tab



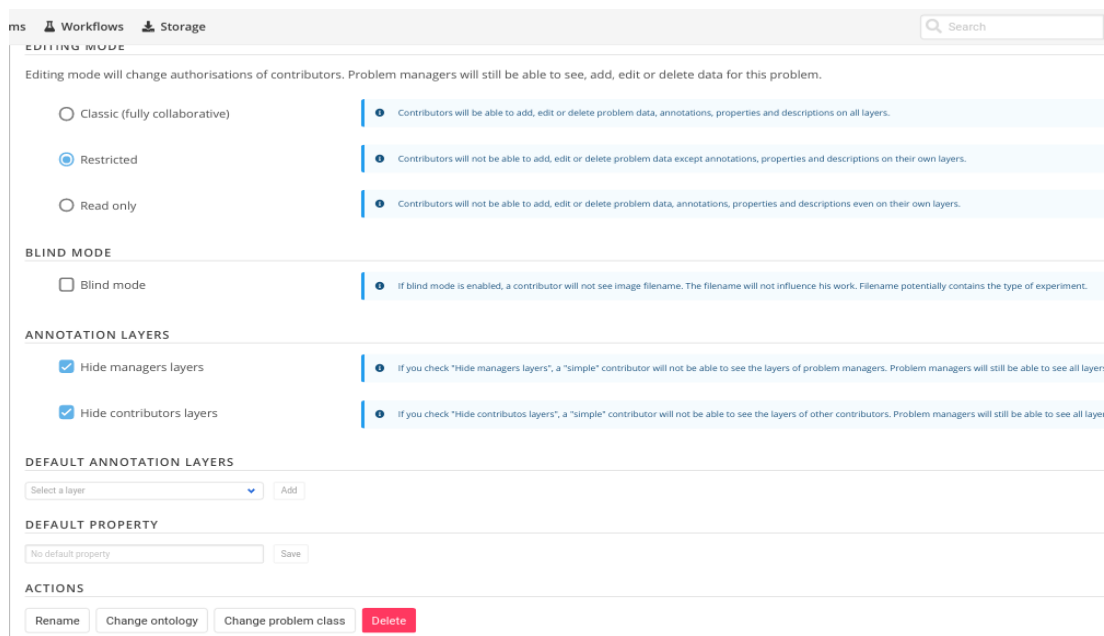
2/ Click **New Problem**



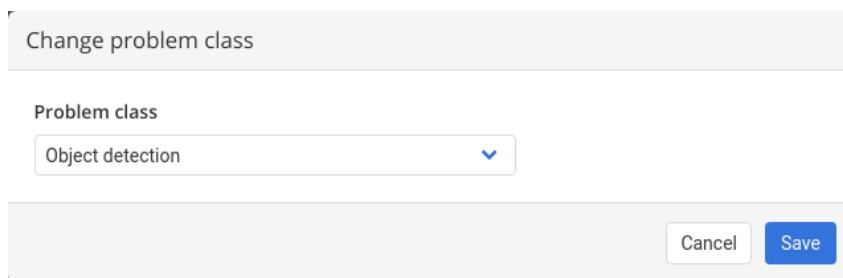
3/ Choose a meaningful problem name and save

The image shows a 'Create problem' dialog box. It has a title bar 'Create problem'. Below the title bar is a 'Name' label and a text input field containing 'SPOT-DETECTION-3D'. At the bottom right of the dialog are two buttons: 'Cancel' and 'Save'.

4/ The problem is ready to be configured, the following configuration is recommended

The image shows the configuration page for a problem in BIAFLOWS. The page is titled 'EDITING MODE' and includes a search bar at the top right. Below the title bar, there is a warning message: 'Editing mode will change authorisations of contributors. Problem managers will still be able to see, add, edit or delete data for this problem.' The configuration is divided into several sections: 'Classic (fully collaborative)', 'Restricted', and 'Read only' (radio buttons); 'BLIND MODE' (checkbox); 'ANNOTATION LAYERS' (checkboxes for 'Hide managers layers' and 'Hide contributors layers'); 'DEFAULT ANNOTATION LAYERS' (dropdown menu and 'Add' button); 'DEFAULT PROPERTY' (text input and 'Save' button); and 'ACTIONS' (buttons for 'Rename', 'Change ontology', 'Change problem class', and 'Delete').

5/ Assign your problem to a problem class (see **Problem Class, Ground truth annotations and reported metrics**) by clicking on **Change problem class**. The problem class specifies the format of ground truth annotations (and workflow outputs), as well as the associated benchmark metrics to be computed (if benchmark is enabled).



6/ Configure project members. If you work alone, you can leave contributors and project managers to default user. This can be done from the “Members” tab in the problem configuration.

7/ The problem can be fully configured to display or hide panels / tabs / tools in the user interface. This is achieved from the **Custom UI** tab in the problem configuration.

8/ A description of the problem can optionally be added from the **Information** (left sidebar). The description is displayed in **Problems** list.

### Uploading images to a local BIAFLOWS instance

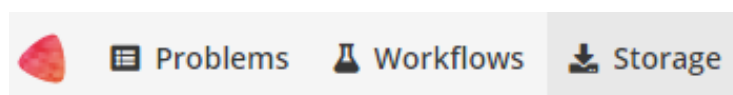
To upload new images, connect as regular user or admin.

#### Supported formats

- **2D images:** 8-bit/16-bit TIFF (or OME-TIFF files)
- **Multi-dimensional images (Z, C, T):** single file 8-bit/16-bit OME-TIFF

**Note:** The text string `_lbl` should not be used in image names since it is a reserved field for ground truth annotation images.

1/ Go to **Storage** section



2/ Select the **Problem** to which the images should be associated with (**Link with problem**)

Storage

Link with problem

Files *No file*

**Note:** If a problem is not in the list, make sure you are a member for this problem

3/ Click on **Add files...** and select the files from the file browser

4/ Start upload with **Start upload** and wait until completion

The status can be:

- **DEPLOYED/CONVERTED:** The image is correctly imported to BIAFLOWS
- **ERROR FORMAT:** The file format is not supported
- **ERROR EXTRACTION:** Something went wrong during metadata extraction
- **ERROR CONVERSION:** Something went wrong during the conversion of the image into the BIAFLOWS internal image format
- **ERROR DEPLOYMENT:** Something went wrong during the communication with BIAFLOWS API. It can be due to access rights, or other unexpected error

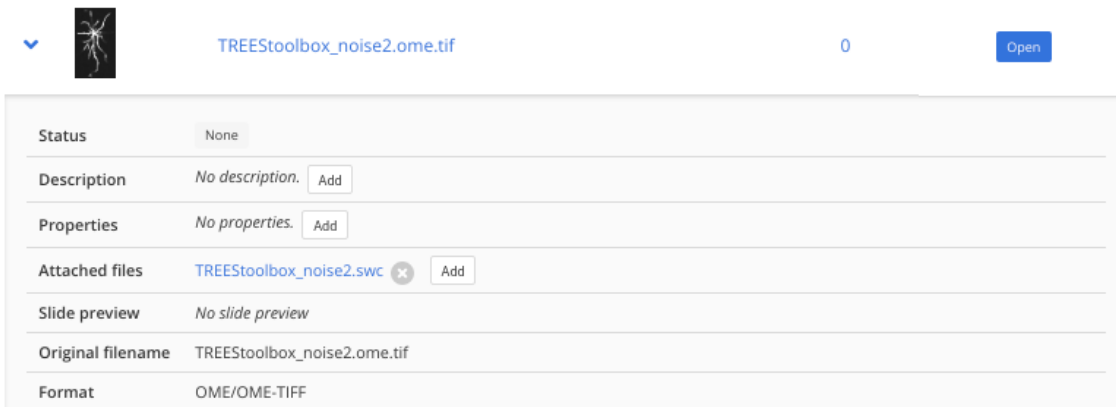
**Note:** Images uploaded to storage can also be associated to a Problem after upload (Problem: **Add image**). This can be useful to associate the same image to several Problems.

### Uploading ground truth annotations to an existing BIAFLOWS problem

If you plan to perform benchmarking, ground truth annotations should also be uploaded and associated to every image of a problem. The format of these annotations depends on the associated problem class (see **Problem Class, ground truth annotations and reported metrics**).

Image annotations (e.g. binary masks) should be uploaded as 16-bit TIFF (or OME-TIFF) for 2D images and as single file 16-bit OME-TIFF for multidimensional (C,Z,T) images. They should be uploaded by following the procedure described in the previous section and by setting the same name as their corresponding image + **\_lbl** suffix (e.g. **AnImage.ome.tif** and **AnImage\_lbl.ome.tif**).

Other required annotations (e.g. SWC, division text file) should be added to the images as attached files. To do so, expand the image (blue arrow) in the list and click on **Add** next to **Attached files**.



## Adding existing workflows from trusted sources to a local BIAFLOWS instance

It is possible to integrate existing BIAFLOWS workflows to any BIAFLOWS instance. This operation requires configuring an external trusted source made of:

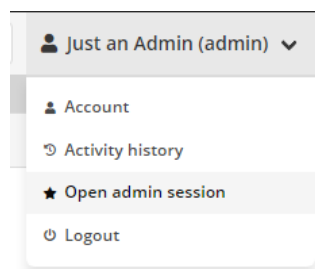
1. A source code registry (typically a Github user space)
2. An execution environment registry (typically a DockerHub user space)

If your workflow repositories are mixed with other repositories in your user space, you can specify a prefix to distinguish workflows repositories. For instance, all bioimage analysis workflows developed by NEUBIAS are prefixed by **W\_** and available from this user space: <https://github.com/Neubias-WG5>.

Some information regarding trusted sources is given below.

To manage trusted sources, you need to be administrator.

1/ Connect as administrator by clicking Open admin session:



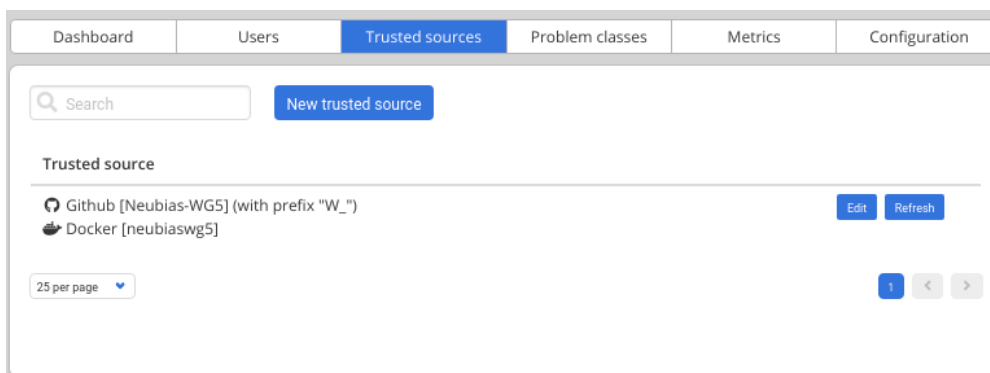
2/ In administration page, go to **Trusted sources** tab and click **Add trusted source**

3/ Fill the form and **Save**

For instance, to add NEUBIAS curated set of workflows, the trusted source has to be configured as follows:

- **Source code provider:** github
- **Source code provider username:** Neubias-WG5
- **Environment provider:** docker
- **Environment provider username:** neubiaswg5
- **Prefix:** W\_

4/ Trusted sources are periodically checked (about every 5/10 minutes) to automatically add new versions of existing workflows or new workflows, but you can also click on **Refresh** to trigger the check.



5) Once a workflow is imported, it has to be linked to a BIAFLOWS **Problem**. This can be performed in the Configuration panel of the **Problem (Workflows tab)** by toggling **Enable** for that workflow as illustrated below:

