

# Install Cytomine on MacOS

 This guide requires Apple Mac OS Yosemite 10.10.3 or above.

This page describes how to run a production-ready instance of Cytomine in a few steps, either on personal computers (such as laptops, desktops) or on larger servers. Once Cytomine's server is installed, your Cytomine instance can be accessed through a modern web browser or via clients using the RESTful API.

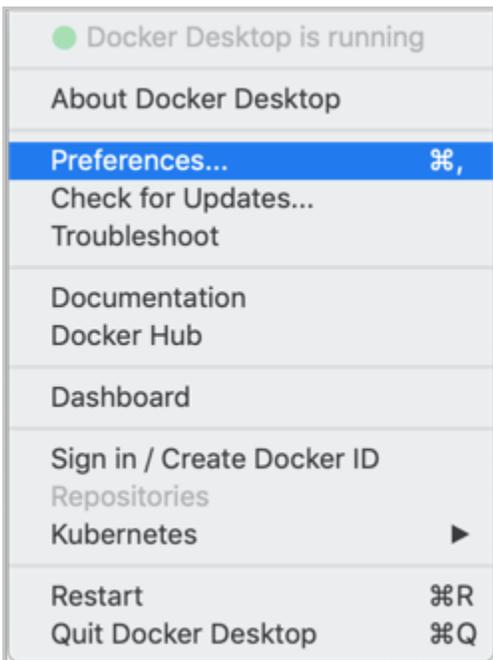
- 1 [Step 1 - Install requirements](#)
- 2 [Step 2 - Retrieve Cytomine bootstrap](#)
  - 2.1 Latest Cytomine ULiege release is available at: <https://github.com/Cytomine-ULiege/Cytomine-bootstrap/releases/latest>
  - 2.2 Latest Cytomine ULiege release can be downloaded here: <https://github.com/Cytomine-ULiege/Cytomine-bootstrap/releases/latest/download/Cytomine-bootstrap.zip>
- 3 [Step 3 - Configure your instance](#)
  - 3.1 [Configure URLs](#)
  - 3.2 [Configure data paths](#)
- 4 [Step 4 - Initialize your deployment](#)
- 5 [Step 5 - Deploy your instance](#)
- 6 [Step 6 - Check your running instance](#)

## Step 1 - Install requirements

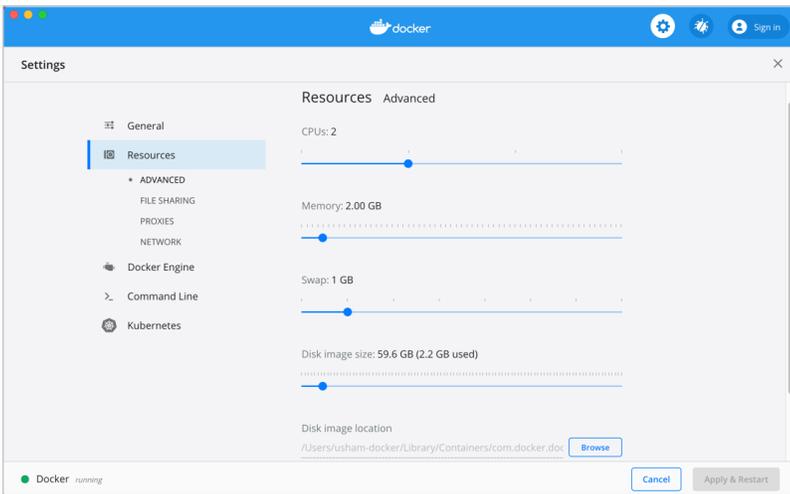
Cytomine runs in Docker containers, so that the only requirement is to install Docker.

[Check official Docker documentation to install Docker for Mac](#). In particular download Docker for Mac from Docker store and follows instructions. Docker for Mac requires Apple Mac OS Yosemite 10.10.3 or above.

Once Docker for Mac is running, Choose  **Preferences** from the menu bar and configure the runtime options described below.



On the Advanced tab, you can change resource limits available to Docker.



By default, Docker for Mac drastically limit the resources available to Docker. Cytomine requires more resources to run without issue.

**CPUs:** By default, Docker for Mac is set to use half the number of processors available on the host machine. **Set the number of CPUs available to a number greater than 75% of the number of processors on the host machine.**

**Memory:** By default, Docker for Mac is set to use 2 GB runtime memory, allocated from the total available memory on your Mac. **Increase RAM to a number at least greater than 4GiB. It is highly recommended to set it to at least 75% of the total available memory on your Mac.**

**Swap:** Configure swap file size as needed. The default is 1 GB. **Set it to 3GiB.**

## Step 2 - Retrieve Cytomine bootstrap

**i** Latest Cytomine ULiege release is available at: <https://github.com/Cytomine-ULiege/Cytomine-bootstrap/releases/latest>

Latest Cytomine ULiege release can be downloaded here: <https://github.com/Cytomine-ULiege/Cytomine-bootstrap/releases/latest/download/Cytomine-bootstrap.zip>

Download the archive and extract it.

```
wget https://github.com/Cytomine-ULiege/Cytomine-bootstrap/releases/latest/download/Cytomine-bootstrap.zip
unzip Cytomine-bootstrap.zip
```

To install Cytomine ULiege latest developments, continuous builds or pre-releases, you can clone the Git repository:

```
git clone https://github.com/Cytomine-ULiege/Cytomine-bootstrap.git
```

## Step 3 - Configure your instance

Open `configuration.sh` file in your favorite text editor such as `nano`.

**✓** Have a look at [Cytomine configuration reference](#) to configure your instance.

```
cd Cytomine-bootstrap
nano configuration.sh
```

## Configure URLs

To install Cytomine locally on your machine, you can keep the `XXX_URL` variables unchanged. Be sure to use URLs that are not already used by other applications (avoid `localhost`) to prevent conflicts.

Due to the isolated nature of Docker containers, some changes on your Mac network configuration are required to allow to communicate with Cytomine directly from your Mac (such as your browser).

Add the `XXX_URL` variable values into the `/etc/hosts` of your host machine.

```
sudo nano /etc/hosts
```

In the `/etc/hosts`, add the following lines and don't forget to adapt them with values you chose for `XXX_URL` variables.

```
127.0.0.1      localhost-core
127.0.0.1      localhost-ims
127.0.0.1      localhost-upload

# if you enabled IRIS plugin
127.0.0.1      localhost-iris

# if you enabled RETRIEVAL plugin
127.0.0.1      localhost-retrieval

# if you enabled SOFTWARE plugin
127.0.0.1      rabbitmq
```

Then, flush your Mac's DNS cache, so it doesn't get confused by any changes you've made to the hosts file.

```
sudo killall -HUP mDNSResponder
```

## Configure data paths

Data managed by Cytomine such as images are stored in directories mapped with your host machine. By default the `/Users` directories (and thus sub-directories) can be mapped in the Docker engine. Check  **Preferences** from the menu bar and then File Sharing tab to change location.

All paths referenced in variables `XXX_PATH` must exist and be mappable in the Docker engine. To create the data in your `$HOME`, set paths with

```
IMS_STORAGE_PATH=~/.data/images
IMS_BUFFER_PATH=~/.data/_buffer
FAST_DATA_PATH=~/.data/images

# if you enabled SOFTWARE plugin
SOFTWARE_CODE_PATH=~/.data/software/code
SOFTWARE_DOCKER_IMAGES_PATH=~/.data/software/images
JOBS_PATH=~/.data/jobs
SERVER_SSHKEYS_PATH=~/.data/ssh

# if you enabled RETRIEVAL plugin
RETRIEVAL_PATH=~/.data/thumb

# if you enabled BACKUP plugin
BACKUP_PATH=~/.data/backup
```

Don't forget to create all these directories (`mkdir`) if they don't exist.

## Step 4 - Initialize your deployment

Generate your installation script with the command

```
sudo bash init.sh
```

## Step 5 - Deploy your instance

Run the generated deployment script with the command

```
sudo bash start.sh  
# OR, in old versions: sudo bash start_deploy.sh
```

Deployment can take some time, especially if it's the first time you install Cytomine on your host machine. All Docker container images have to be downloaded.

## Step 6 - Check your running instance

When start up is finished, check the application is running in your browser on the URL specified in your `CORE_URL` variable (by default: localhost-core).

An admin account is created by default. To connect,

- **Username:** admin
- **Password:** retrieve it in `Cytomine-bootstrap/configs/core/cytomineconfig.groovy`. Search the line with `grails.adminPassword`. For example, in `grails.adminPassword='5366E276-7574-4155-84C1-E1EA5ECB1039'`, the password is `5366E276-7574-4155-84C1-E1EA5ECB1039`