Install Cytomine on Linux

This guide is designed for Linux Ubuntu 16.04+ (18.04 tested). Other distributions should be supported with slightly little modifications.

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.

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 Ubuntu. In particular, choose **Install using the repository**, set up the repository and install Docker CE.

You can optionally improve your Docker installation with the post installation steps for Linux.

Step 2 - Retrieve Cytomine bootstrap

To install Cytomine with the last features from ULiege research team, run these commands:

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

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.

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
```

Configure data paths

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/softwares/code
SOFTWARE_DOCKER_IMAGES_PATH=~/data/softwares/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

Once connected on http://localhost-core (if you kept `CORE_URL = localhost-core`), click on you name at the top right in the web interface, and go to **Account** to change the admin password.