Swarmlab quick-start guide!

Πίνακας περιεχομένων

Welcome to the vast world of Swarmlab!

Swarmlab is a powerful collaboration tool that connects tutors and students in a unique all-in-one environment that gets work done fast and easily.

In other words Swarmlab is the way to make your learning fun while at the same time getting to your goals faster!

But let’s get you started…

Note: These instructions are general targeted at both students and tutors.
If you wish to skip to the specific instructions for tutors (lab creation, student invitation etc) click here.

> ⚠ "General info WILL NOT be repeated so if you have basic questions regarding registration or labroom usage please read on from here.

The first steps

Before you have all of the powerfull tools of Swarmalb at your grasp you will have to create an account.
To do that navigate to the home page at https://api-client.swarmlab.io:8088
You should have been greeted by our login screen!

[login] | */images/registration/login.png*

To register click on the 'login with sso' option and accept the terms.

[terms] | */images/registration/terms.png*

Now you should be redirected to the login screen and you should be able to use the 'register' option at the bottom to make a new account!

[login main] | */images/registration/login_main.png*

Now fill in the required details

[register details] | */images/registration/register_details.png*

and follow the remaining instructions to activate your account.

Now that your acount is ready, lets get to know Swarmlabs main page.

The Swarmlabs Platform

The main page

Once loaded the main page should look like this:

[main page] | */images/run/main_page.png*

On the left we can see that there are two areas inside swarmlab:

1. the dashboard where our services live
2. the 'mylab' tab from which we can control our labrooms.

The dashboard page is pretty straight forward:

1. The left list shows all the available services and allows you to run them.
2. The other shows the available lab rooms and allows you to interact with them (p.e. subscribe).

With that out of the way lets enter the heart of our platform, 'my lab'!

My Lab

Nested under 'mylab' are both creation tools (bootstrap) and execution tools (run).

1. The run menu

Inside the run menu there are three areas:

1. 'Rub lab instance'
2. 'Enter lab room'
3. the console

As the names suggest we can use the first to **run** one of our available lab instances, the second to **enter** a created labroom, and the third to **execute** scripts.
Lets look at the options one by one.
For the purpose of this walkthrough 'testmpi2' and 'test 18' were created.
Since the console is explained along with the first area we dont feel there is a need to have seperate walkthrough.

1.1. Run lab instance area

1.1.1. Creating a lab

To create a lab instance simply add a repo to your git!

You will then be able to locate that repo inside the instance list and run it.
For more info on this please refer to the Bootstrap chapter here !

1.1.2. Running a lab instance

First we have to select the lab we want to run. To find a specific lab we can use the search bar at the top.

[lab instance list] | */images/run/lab_instance_list.png*

After we have located the desired lab we can run it by using the cloud icon. [cloud icon]

To request more info about the lab we use the info icon [info icon] and the trashcan [trash icon] deletes the lab (forever).
Carefull, forever is a long time!

So lets go back to running the lab. After clicking on the according icon we should be redirected to the execution screen.

[execution screen] | */images/run/execution_screen.png*

Here we can use the command line on the bottom to execute the required commands. Our results will be visible in the console above.

[ls results console] | */images/run/ls_results_console.png*

Note that you can choose to use the editor of your liking from the drop down menu.

For more complicated tasks that may require responsivines from the machine you can use the terminal option to open a full-blown unix command line.

1.2. Enter lab room area

This is the area where you will find the available labrooms created by tutors and join the desired one.

1.2.1. Browsing labrooms

You can easily browse throught the available labrooms in the list using the search bar at the top.

[lab room list] | */images/run/lab_room_list.png*

You can get more info about a specific labroom by clicking the info button [info icon].

[info poppup] | */images/run/info_poppup.png*

Active labrooms are indicated using a green color while closed ones with orange.

1.2.2. Entering a labroom

Once you have found the labroom you were looking for you can enter by simply clicking on the

cloud icon [cloud icon].
You should see a screen as follows:

[password big] | */images/run/password_big.png*

The passwords are randomly generated and you can use them to connect to your lesson (see picture bellow).

[password user] | */images/run/password_user.png*

If the tutor is sharing his screen you can watch by clicking on the according button and using the respective password.

[password tutor screen] | */images/run/password_tutor_screen.png*

Now, under the password area there are three other options:

'Meeting' [meeting button]

Here you can join a meeting with your group (this will redirect you to a jitsi implementation).
For more visit https://jitsi.org/

'Gui' [gui button]

This loads a graphic unix environment.

'Console' [console button]

This loads a unix terminal.

By now it should propably be clear how to run a lab from the list or enter a labroom.
But you are propably wondering 'My list is empty how am i supposed to run labs from it?!'
Well here is where the final but most basic menu comes into play: **'Bootstrap'** !

Bootstrap

[bootstrap main] | */images/bootstrap/bootstrap_main.png*

Here we can create and add labs to our list. To do so we have to go through a series of steps, but lets first understand some basic things about how swarmlab works...

> Every swarmlab lab instance resides in a git repo created inside our git and owned by the lab creator. This repo then serves as a 'home' for all the work inside that lab.
> So for us to create and run a lab we have to link a service to a git repo.

Asuming we have an account (using the same email as our swarmlab account!) all our repos should be shown on the list in the top left corner.
To search for any public repo we can use the search bar.

[test search repo] | */images/bootstrap/test_search_repo.png*

From that list we can now select the desired repository that we want to use by clicking on the plug

icon. [plug icon]
We can use any repo we like, but we will be able to commit only to those that we have access to.
If we wish to edit or contribute to the development of a project we are not part of we can use the clone option as usual!

Now we have to select the service that our lab will be using. We just select the desired one from the list on the right and it will be automatically linked to the project.

[service select] | */images/bootstrap/service_select.png*

Finally we have to save the lab instance by filling in the required info at the bottom.

[save lab details] | */images/bootstrap/save_lab_details.png*

For Tutors

Tutors have access to an extended 'my lab/bootstap' page where they can create labrooms from existing lab instances.

To do so we have to scroll all the way down whilst on the Bootstrap page.
There we should find two menus, 'My labrooms' and 'Bootstrap Labroom'

1. 'My labrooms'

[my labrooms] | */images/for_tutors/my_labrooms.png*

Here we can see a list of all the labrooms programmed to run today. Again like in the run tab we can distinguish open from closed ones by color (green/orange).

[todays labrooms] | */images/for_tutors/todays_labrooms.png*

We can also get info using the info icon [info icon] or delete the labroom using the trashcan icon [trash icon] .
To search for a specific labroom we can use the searchbar at the top.

> ❗ The delete command is **FORCED** even if the labroom is running!!!

2. 'Bootstrap Labroom'

Now lets see how we can bootstrap a new labroom using one of our lab instances.
To do that lets navigate to the respective tab that should look like this:

[bootstrap labroom] | */images/for_tutors/bootstrap_labroom.png*

Here we can select the desired (existing!) lab instance that will be used to create the labroom using the plug icon [plug icon].

[lab instance choose] | */images/for_tutors/lab_instance_choose.png*

Then we have to specify starting date and time and also set the duration for which our lab room will be open.

[timeframe selection] | */images/for_tutors/timeframe_selection.png*

> 🔥 You can only start a labroom every whole or half hour (p.e. 12.00 or 14.30) and never inbetween so please plan ahead!!
> Also note that the duration cannot be greater than three hours since we consider that the maximum for any type of lesson!

Now using the searchbar that follows (you HAVE to search for students for them to come up) we search and select ([plug icon]) the students that we want to invite to the lesson.

Note that only students that have subscribed to our labroom will be available for invites to avoid confusion!

[student selection] | */images/for_tutors/student_selection.png*

After selecting all the desired students we have to fill in the last details at the bottom of the page. Please give concise titles and good descriptions to your labrooms!!

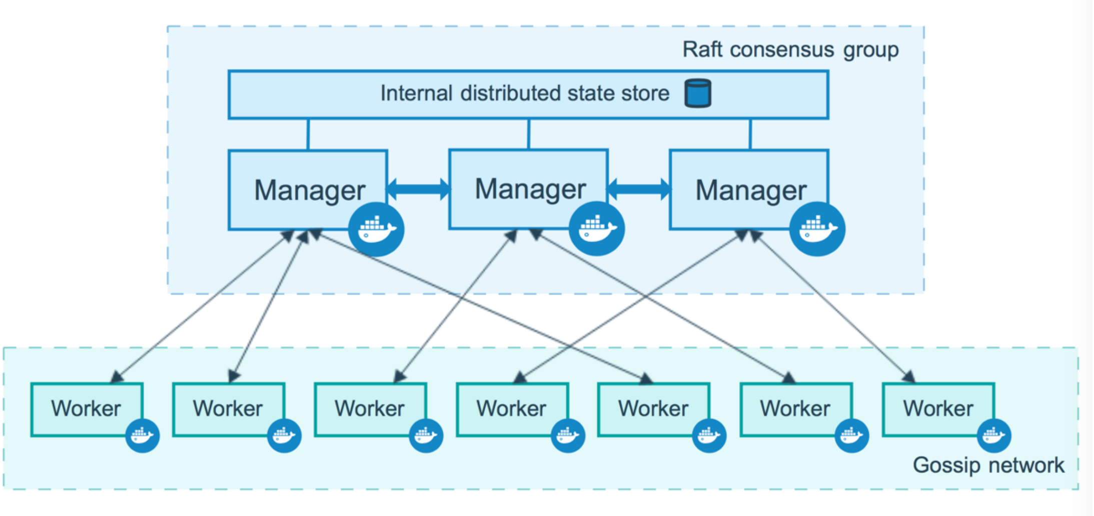
[bootstrap details] | */images/for_tutors/bootstrap_details.png*

Now we shall check and upon making shure that everything is right we click the 'bootstrap your lab' button!

> ⚠ Bootstraping is final and no further edits are allowed!!

Platform Architecture

Architecture overview



1. Storage

1.1. IPFS Cluster 🌍

Automated data availability and redundancy on IPFS.

IPFS Cluster provides data orchestration across a swarm of IPFS daemons by allocating, replicating and tracking a global pinset distributed among multiple peers.

It is being used for both tutors and students to publish content on a **decentralized** manner!
Information will be **auto-deleted** after approximatelly **24 hours** of not being used.

1.2. minio cluster 🌍

High Performance, cloud Native Object Storage.

Amazon's S3 API is the defacto standard in the object storage world.
MinIO is the defacto standard for S3 compatibility and was one of the first to adopt the API and the first to add support for S3 Select.
More than 750 organizations, including Microsoft Azure, use MinIO's S3 Gateway - more than the rest of the industry combined.

It is being used for both tutors and students to save content on a **centralized** manner!
Information storage is **permanent**.

1.3. MariaDB Galera Cluster

MariaDB Galera Cluster is a virtually synchronous multi-master cluster for MariaDB.
Provide **Active-active** multi-master topology.

Used for user/admin authorization and authentication.

1.4. A replica set in MongoDB

MongoDB is a document database designed for ease of development and scaling.

Used for labroom and swarmlab services administration and scheduling.

1.5. ShareDB

ShareDB is a realtime database backend based on Operational Transformation (OT) of JSON documents

2. Devel

2.1. slim

Slim is a PHP micro framework that helps you quickly write powerful APIs.

Used for main api of swarmlab.io
Slim was used for a fast and powerful router that maps route callbacks to specific HTTP request methods and URIs.

2.2. nodejs

Nodejs is an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.

Used for event-driven programming and for Sockets and WebSockets.

2.3. vue.js

Vue is a progressive JavaScript framework for building user interfaces.
It is designed from the ground up to be incrementally adoptable, and can easily scale between a library and a framework depending on different use cases.
It consists of an approachable core library that focuses on the view layer only, and an ecosystem of supporting libraries that helps you tackle complexity in large Single-Page Applications.
Used for the main user interface of swarmlab.io.

2.4. python 🌍

Python is an interpreted, high-level, general-purpose programming language.
Python’s design philosophy emphasizes code readability with its notable use of significant whitespace.
Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
Used along with ansible for maintenance and administration of the machines making up the cluster of swarmlab.io.
se sinrgasia me tin ansible gia tin diachirissi ton michanon

2.5. php 🌍

PHP is a popular general-purpose scripting language that is especially suited to web development.
Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

Used with slim for creation of the main API.

2.6. codemirror 🌍

CodeMirror is a versatile text editor implemented in JavaScript for the browser.

Used for the online editors for interraction with tutors and students.

2.7. oauth2 🌍

OAuth 2.0 is the industry-standard protocol for authorization.
OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices.
This specification and its extensions are being developed within the IETF OAuth Working Group.

This protocol is being used for the connection of users and services.

3. System admin and devel

3.1. docker 🌍

Docker provides the ability to package and run an application in a loosely isolated environment called a container.

Used for all the services implementations.

3.2. docker-compose

Compose is a tool for defining and running multi-container Docker applications.
Used for development, configuration and testing of the services before deployment to the swarm.

3.3. Bash

A lot of bash scripting was used (low-level) tasks concerning the Linux operating system.

3.4. ansible

Ansible is an open-source software provisioning, configuration management, and application-deployment tool.

Used for administration of the machines of the general cluster of swarmlab.io as well as its subclusters.

3.5. keycloak

Open Source Identity and Access Management

Used as backend for authentication to provide secure services in our sso implementation.

4. Deploy

4.1. Docker swarm

Docker swarm is a container orchestration tool, its allow to manage multiple containers deployed across a cluster.

Used for orchestration of swarmlab.io services.

5. Colaboration

5.1. gittea

A self-hosted Git service used for the git-repo management used by students and tutors.

5.2. jitsi

A secure, flexible, and completely free video conferencing.

Used for the online virtual meetings attended by students and tutors.

6. service access

6.1. noVNC

A open source VNC client - it’s is both a VNC client JavaScript library as well as an application built on top of that library.
noVNC runs in any modern browser including mobile browsers (iOS and Android).

Used to access a unix graphic environment for the services provided through a normal browser.

6.2. ttyd

Terminal over the web.

Used to access a unix command line for the services provided through a normal browser.