

# Cloud computing - Service mesh and microservices networking!

## Table of contents

1. General info: .....	3
1.1. Timeframe: .....	3
1.2. What will i learn? .....	3
1.3. What tools will I need .....	3
1.3.1. Swarmlab.io hybrid .....	4
1.4. What students can take this course .....	4
1.5. How is the course going to take place .....	4
1.6. Will there be some kind of exam/certificate? What will i gain? .....	4
2. Course Description .....	4
2.1. Cloud & microservice .....	4
2.1.1. docker app .....	4
2.1.2. docker swarm .....	5
2.1.3. Orchestration .....	5
2.2. Administer and maintain a swarm of Docker Engines .....	5
2.2.1. manager nodes .....	5
2.2.2. Monitor swarm health .....	5
2.2.3. Scheduling Services on a Docker Swarm Mode Cluster .....	5
2.2.4. ansible .....	5
2.3. Create service on nodes .....	5
2.4. Monitoring - service applications communication .....	6
2.4.1. Real-Time data/Log Collection .....	6
2.5. create noSQL DB (mongo cluster) .....	6
2.5.1. create replicas .....	6
2.6. central web admin interface .....	6
2.6.1. vuejs .....	6

## Service mesh and microservices networking

We will be trying to create a swarm implementation that will allow communication between all of the members/nodes.

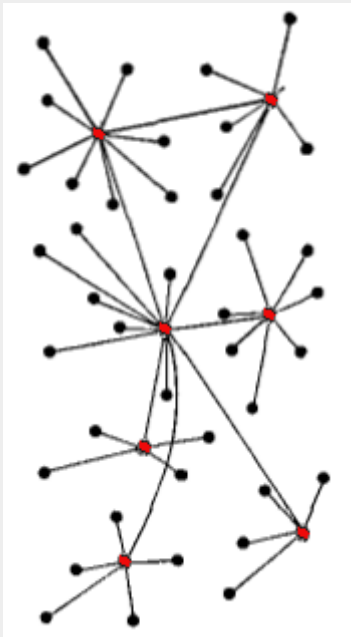
*Imaging a swarm*



To understand this better let's look at the picture below and imagine that red dots are IoT devices that can send and receive and black ones are clients that gather data.



*Architecture of swarm communication*



- Red Node: Server/Client and Gateway Role
- Black and Red Node: Client Role

**To make our life easier at this task we will be using the following tools...**

- Docker [↗](#)
- Ansible [↗](#)

- [NodeJS](#) ↗
- [VueJS](#) ↗
- [Redis](#) ↗
- [MongoDB](#) ↗

# 1. General info:

## 1.1. Timeframe:

This is a project that will be developed throughout the semester 2021.

*Courses Time: 11.00-13.00*

Date

- 2021-03-13, 2021-03-20, 2021-03-27
- 2021-04-03, 2021-04-10, 2021-04-17, 2021-04-24
- 2021-05-15, 2021-05-22, 2021-05-29, 2021-06-05
- 2021-06-12, 2021-06-19



- The link to participate will be **announced here**.
- You can also see detailed information about the sessions etc in the **calendar** (main page of the swarmlab client) after installation (see section "What tools will I need")
- For any questions you can visit our community page on [gitter](#)
- If none of the above solved your problem, you can always use good old-fashioned [email](#)!

## 1.2. What will i learn?

You will learn to code, coordinate and orchestrate a swarm of self-acting nodes.

## 1.3. What tools will I need

Internet and a PC

You should also have **installed swarmlab** to be able to recreate the steps and if you wish contribute to projects and communicate with your schoolmates.

### 1.3.1. Swarmlab.io hybrid

Follow the [instructions](#) to install swarmlab-hybrid

See also

[📺](#) | [vimeo](#)

## 1.4. What students can take this course

Any student with basic knowledge of networking and computer programming should be able to cope with the needs.

## 1.5. How is the course going to take place

The course will be divided into following parts

- A list of videos, asciinemas and instructions explaining the project
- lectures BASED ON THE VIDEOS for deeper analysis and questions
- and a [Gitter](#) for further conversations and answers to any of your questions

## 1.6. Will there be some kind of exam/certificate? What will i gain?

- The will NOT be an exam or certificate.
- You will gain contributions in form of commits and merge requests into larger projects, which you can then add to your C.V. and upgrade it.



Just to give some context, **contribution of code is regarded as the most important factor when choosing a software engineer**, thus making the course very helpful for future employment

## 2. Course Description

### 2.1. Cloud & microservice

#### 2.1.1. docker app

The section guides you through the following activities:

- Create a Dockerized Sample application
- Start an app container

### **2.1.2. docker swarm**

The section guides you through the following activities:

- initializing a cluster of Docker Engines in swarm mode
- adding nodes to the swarm
- deploying application services to the swarm
- managing the swarm once you have everything running

### **2.1.3. Orchestration**

The section guides you through the following activities:

- scale our containerized applications across clouds and datacenters

## **2.2. Administer and maintain a swarm of Docker Engines**

### **2.2.1. manager nodes**

### **2.2.2. Monitor swarm health**

### **2.2.3. Scheduling Services on a Docker Swarm Mode Cluster**

- Scheduling Preferences
- Rescheduling on Failure

### **2.2.4. ansible**

- Using ansible to perform operations on managed nodes aka Configurations, deployment, and orchestration/automation
- Deploying Docker Containers with Ansible

## **2.3. Create service on nodes**

This section includes Docker images and an application for Node development using containers.

Create Real-time Application with

- Node.js
- Express.js
- Socket.io
- Redis

## **2.4. Monitoring - service applications communication**

### **2.4.1. Real-Time data/Log Collection**

## **2.5. create noSQL DB (mongo cluster)**

A replica set is a group of mongod processes that maintain the same data set

### **2.5.1. create replicas**

- Replication in MongoDB
- Change Streams
  - work with the change stream cursor.
  - Watch Collection/Database/Deployment etc

## **2.6. central web admin interface**

Create a CRUD App

### **2.6.1. vuejs**

Create single-page application