

COMP.SE.140

Session #3

Kari Systä
12.09.2023

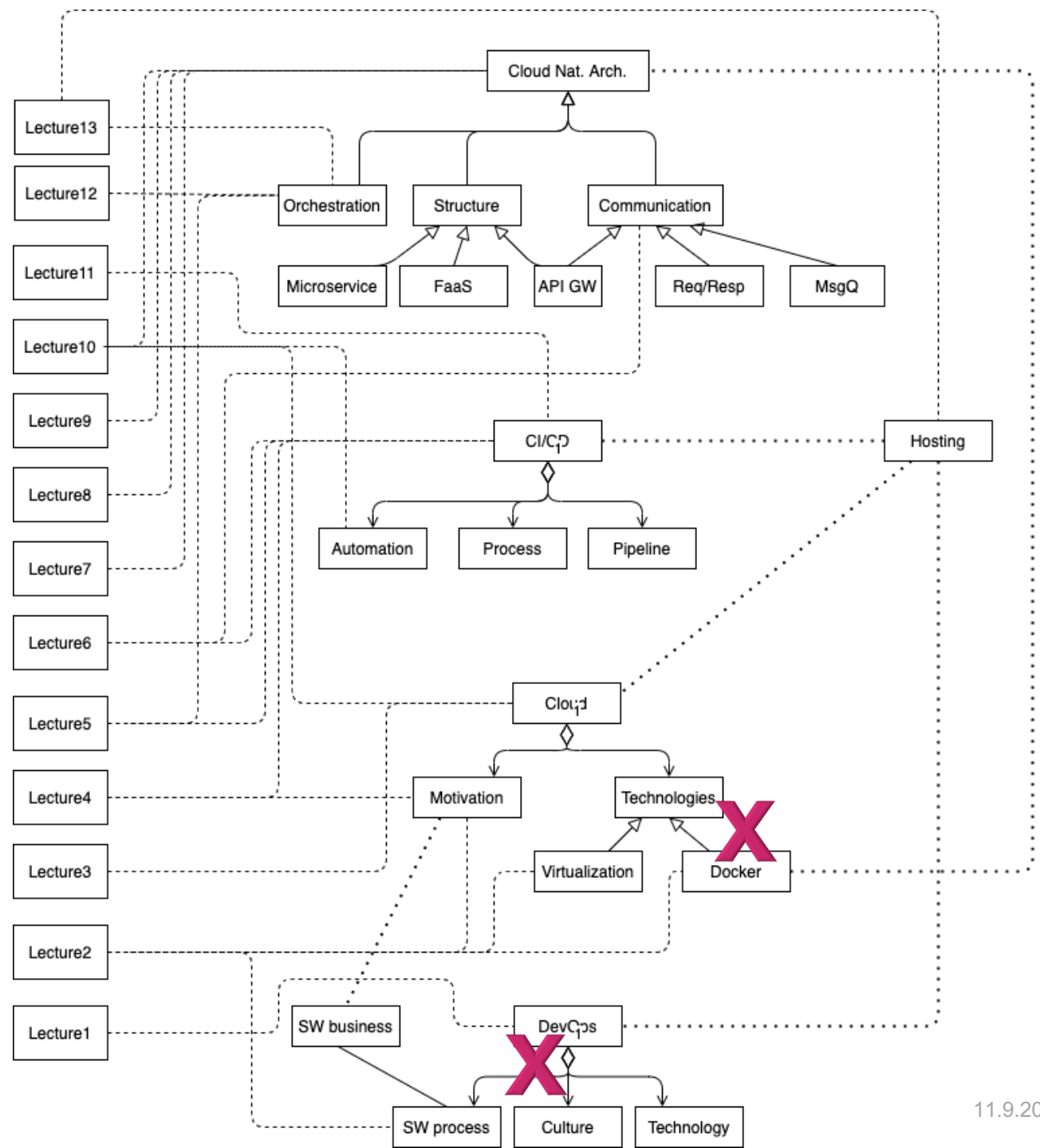
Today

- Course status
- Discussion about containers and docker
- Introduction to the first exercise

You in numbers

- SISU participants
 - 155 of which 19 stopped/skipped and 13 under discussion
=> 123 – 136
- In addition plus has 17 enrolment that are not in SISU

Content map



Discussion about containers

Your experience?

Why - benefits?

Why not - problems?

Dockerfile

```
#This is a sample Image  
FROM ubuntu  
MAINTAINER demousr@gmail.com  
RUN apt-get update  
RUN apt-get install -y nginx  
CMD ["echo", "Image created"]
```

kari@lepikko-nuc:~\$ docker history pingrelay

| IMAGE | CREATED | CREATED BY | | SIZE | COMMENT |
|--------------|-------------|---|-------------------------------|--------|---------|
| 52b9f6060548 | 2 years ago | /bin/sh -c #(nop) | CMD ["/bin/sh" "-c" "node... | 0B | |
| 20e97c5340b3 | 2 years ago | /bin/sh -c #(nop) | EXPOSE 8894 | 0B | |
| aed7842203d6 | 2 years ago | /bin/sh -c #(nop) | ENV PORT=8894 | 0B | |
| b42a225080f7 | 2 years ago | /bin/sh -c #(nop) | COPY dir:10a4bc704d33f8ef9... | 2.86kB | |
| d62176fb4127 | 2 years ago | /bin/sh -c #(nop) | USER node | 0B | |
| 8232a8b9c483 | 4 years ago | /bin/sh -c #(nop) | CMD ["node"] | 0B | |
| <missing> | 4 years ago | /bin/sh -c apk add --no-cache --virtual .bui... | | 3.56MB | |
| <missing> | 4 years ago | /bin/sh -c #(nop) | ENV YARN_VERSION=0.21.3 | 0B | |
| <missing> | 4 years ago | /bin/sh -c adduser -D -u 1000 node && ap... | | 45.4MB | |
| <missing> | 4 years ago | /bin/sh -c #(nop) | ENV NODE_VERSION=6.10.0 | 0B | |

52b9f6060548

20e97c5340b3

aed7842203d6

b42a225080f7

d62176fb4127

8232a8b9c483

[Empty box]

[Empty box]

[Empty box]

[Empty box]

| SIZE | COMMENT |
|--------|---------|
| 0B | |
| 0B | |
| 3.56MB | |
| 0B | |
| 45.4MB | |
| 0B | |
| 0B | |
| 4.81MB | |

Base images used in exercise 4

| | | | |
|------------------------------|----|----------------------------|---|
| • node:10 | 20 | • golang:alpine AS builder | 1 |
| • node:10-alpine | 3 | • golang:latest | 1 |
| • node:10.15.3-stretch | 1 | • python | 1 |
| • node:10.16.3-alpine | | • python:3 | 2 |
| • node:11 | | | 2 |
| • node:11-alpine | | | 2 |
| • node:12 | | | 2 |
| • node:12.2-alpine | | | 1 |
| • node:8.16.1-alpine | | | |
| • node:8.16.1-jessie-stretch | | | |
| • node:alpine | | | |
| • node:latest | | | |

node:<version>

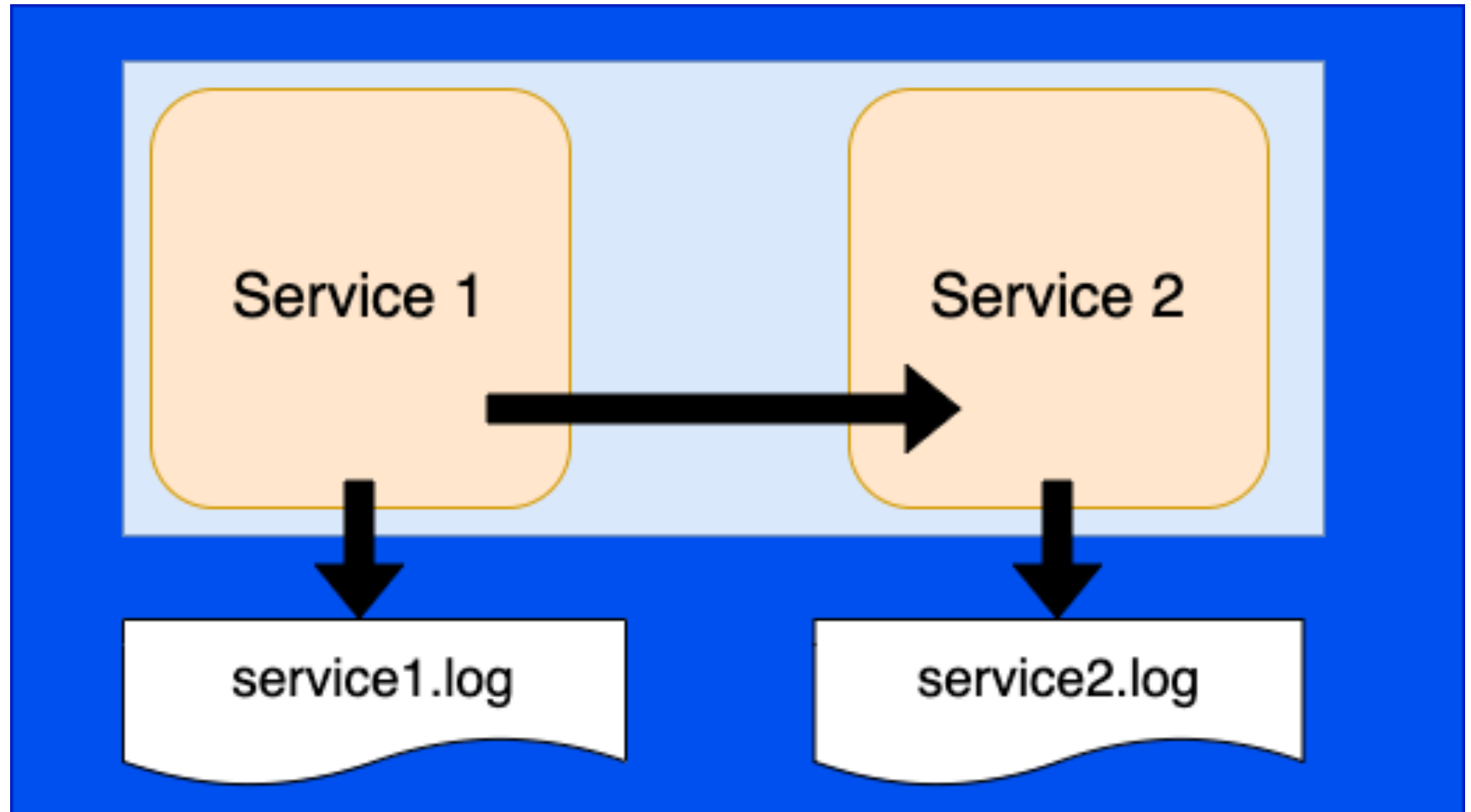
This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

First real exercise

Simple Docker-compose application

Learning goals

- Learn some hands on with Docker and Docker Compose. This is needed in the next steps of the course.
- Understand the runtime context of Docker containers, for instance networks and volumes.
- See the value of virtualization for application development.



The files “service1.log” and “service2.log” stored in folder “logs” in the hosting Linux and made available to services with the “volume” directive in docker-compose file.

In the beginning of the execution both services should

- Create service1.log / service2.log if it does not exists
- Ensure that the files are empty in the beginning.

Service/application 1 should repeatedly 20 times in 2 second intervals:

- Compose a **text** from a counter (initialized to 1) current time and address+port of service2. Follow this example syntax precisely:
`1 2022-10-01T06:35:01.373Z 192.168.2.22:8000`
- Write the above text to file “service1.log” (each writing to a separate line)
- Send the text with HTTP protocol to service 1.
- If the sending fails, catch the exception and write the error message to “service1.log”
- Increase the counter with 1

After the 20 rounds

- Write “STOP” to file “service1.log”
- send STOP to service 2
- Close the file “service1.log”
- Exit the service

Service/application 2 should

- Wait for 2 seconds
- Establish an HTTP server that listens in port 8000
- As a response to incoming message create a new text that adds the remote address (address of service 1). An example:
`1 2022-10-01T06:35:01.373Z 192.168.2.22:8000 192.168.2.21:78390`
- Write that text to file “service2.log”
- If the received text was “STOP” close file “service2.log” and exit.

Submitting for grading

After the system is ready the student should return (in the git repository – in branch “exercise1”).

- Content of two Docker and docker-compose.yaml files
- Output of “**docker container ls**” and “**docker network ls**” (executed when the services are up and running.) in a text file
- Source codes of the applications.

You should prepare your system in a way that the course staff can test the system with the following procedure (on Linux):

- `$ git clone -b exercise1 <the git url you gave>`
- `$ docker-compose up -build`
- ... wait a minute
- `$ docker-compose down`
- `$ more logs/service1.log`

On using ChatGPT or similar AI (large language models - LLM) tools

- The university-level guidelines say:
 - *“If a student uses a language model in an assignment or a thesis, for example, as part of language editing, this must always be mentioned. When individual students describe their use of language models, we can share good practices. The use of a language model for language revision is justified, for example, to produce a grammatically or structurally fluent text (cf. proofreading and translation tools and similar tools).”*

In this exercise we interpret this as follows

In this exercise we interpret this as follows

- If language models are used, a separate report (~1page) must be written (included in the submission). This report includes:
 - Motivation/reason to use LLM
 - Text on how and why LLM helped
 - What kind of mistakes LLM did
 - What were things that LLM was not able to provide
- You allow course staff to use this report in grading and use of it (after anonymization) for teaching development/research.
- The course staff will investigate different ways to discover use of LLM – students using LLM without reporting it, will be discontinued from the course.

About Git

- I have created everybody a private repository in student-gitlab (you should have received an email)
- You can use other git-instances, too, but the teachers need to have an access.