### **Docker and Docker Compose**

https://github.com/heig-vd-dai-course

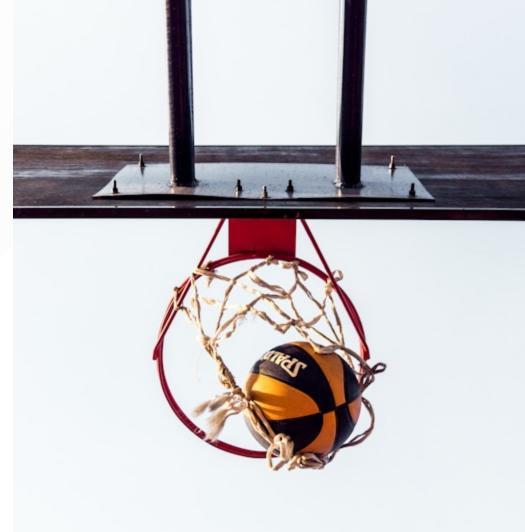
### Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

This work is licensed under the <u>CC BY-SA 4.0</u> license.

### **Objectives**

- Learn the differences between bare metal, virtualization and containerization
- Learn how the OCI specification defines images, containers, and registries
- Learn how to use Docker and Docker Compose to build, publish, and run applications in containers



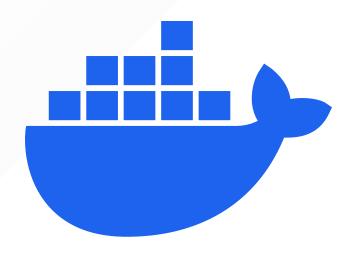
### Prepare and setup your environment

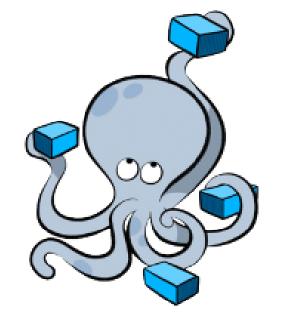
More details for this section in the <u>course material</u>. You can find other resources and alternatives as well.

### Install Docker and Docker Compose

- Install Docker and Docker Compose
- Configure Docker and Docker Compose to:
  - Run without sudo (root)
  - Start automatically at boot







# Check and run the code examples

- Check the code examples
- Run the code examples
- Helps to understand the concepts
- Modify/play with the code examples

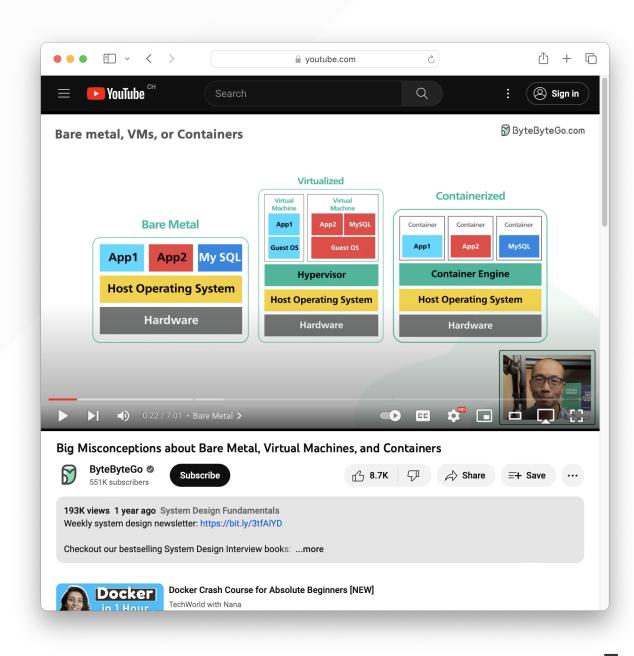
	dai-course-code-examples Q Type [] to sear	ch	+ • • • n @
Code ⊙ Issues 1 11 Pull reque		⊻ Insights 🕸 Se	
heig-vd-dai-course-code-e			-
B heig-va-dal-course-couc-c			
양 main ▾ 양 2 Branches ♡ 1 Tags	Q Go to file	+ <> Code -	About
() Iudelafo Update code examples for	Java IOs chapter (#4) 917c389 · 2 weeks ag	0 🕚 22 Commits	This repository contains the code examples related to the main repository.
D5-java-ios	Update code examples for Java IOs chapter (#4)	2 weeks ago	Readme
11-smtp-and-telnet	Add examples to chapters 11, 12 and 13	last year	-∿ Activity
12-ssh-and-scp	Add all explanation regarding web infrastructur	10 months ago	E Custom properties ☆ 2 stars
📘 13-java-tcp-programming	Update code examples for Java IOs chapter (#4)	2 weeks ago	<ul> <li>2 stars</li> <li>3 watching</li> </ul>
22-web-infrastructures	Update examples for web infrastructures	9 months ago	양 2 forks
23-practical-work-4	Small improvements to Traefik documentation	8 months ago	Report repository
🗋 .gitignore	Add examples to chapters 11, 12 and 13	last year	Releases
C README.md	Add all explanation regarding web infrastructur	10 months ago	🔁 1 tags
			Create a new release
🕮 README		Ø	create a new release
README		Ø	Contributors 3
	urse - Code examples	I	
HEIG-VD DAI Co	-	Ø	Contributors <sup>3</sup>
HEIG-VD DAI COU	examples related to the <u>main repository</u> .	D	Contributors 3
HEIG-VD DAI COU	examples related to the <u>main repository</u> . related to its specific chapter in the course.	D	Contributors 3 Judelafo Ludovic Delafontaine Adarylouis Hadrien Louis Cobora2001 Thomas V.
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	1	Contributors 3 Judelafo Ludovic Delafontaine hadrylouis Hadrien Louis Cobora2001 Thomas V. Languages
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	0	Contributors 3 Judelafo Ludovic Delafontaine Adarylouis Hadrien Louis Cobora2001 Thomas V.
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	0	Contributors 3 Judelafo Ludovic Delafontaine hadrylouis Hadrien Louis Cobora2001 Thomas V. Languages
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	0	Contributors 3 Judelafo Ludovic Delafontaine hadrylouis Hadrien Louis Cobora2001 Thomas V. Languages Java 100.0% Suggested workflows
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	0	Contributors 3 Judelafo Ludovic Delafontaine Cobora2001 Thomas V. Languages Java 100.0% Suggested workflows Based on your tech stack Madroid Cl Build an Android project with Grade.
HEIG-VD DAI Cou This repository contains the code Each directory contains examples	examples related to the <u>main repository</u> . related to its specific chapter in the course.	0	Contributors 3 ludelafo Ludovic Delafontaine hadrylouis Hadrien Louis Cobora2001 Thomas V. Languages Java 100.0% Suggested workflows Based on your tech stack Android CI Configure

## Bare metal, virtualization and containerization

More details for this section in the <u>course material</u>. You can find other resources and alternatives as well.

## Bare metal, virtualization and containerization

- Bare metal: software runs directly on hardware
- Virtualization: software runs on a virtual machine
- Containerization: software runs in a container



### **Bare metal**

- The traditional way to run software
- Software runs directly on hardware
- Software has full access to the hardware
- Security issues, hard to maintain, hard to migrate



### Virtualization

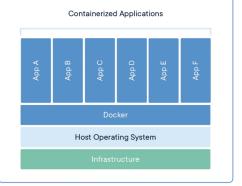
- Virtualization runs virtual machines
- A virtual machine is complete operating system
- A virtual machine is isolated from the host
- Virtual machines are heavy and use a lot of resources



### Containerization

- Containerization starts containers
- Containers contain all the dependencies to run the software
- Containers are isolated from each other
- Containers are lightweight and use the host kernel





Virtual Machine	Virtual Machine	Virtual Machine			
Арр А	Арр В	App C			
Guest Operating System	Guest Operating System	Guest Operating System			
Hypervisor					
Infrastructure					

#### CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

#### VIRTUAL MACHINES

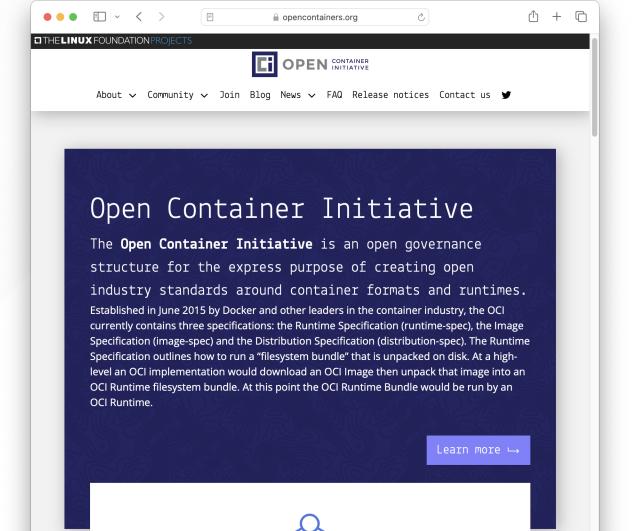
Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries – taking up tens of GBs. VMs can also be slow to boot.

### **OCI**, images, containers, and registries

More details for this section in the <u>course material</u>. You can find other resources and alternatives as well.

## OCI, images, containers, and registries

- Image: read-only template for container creation
- Container: runnable instance of an image
- Registry: service storing images



### **Docker Hub**

- The official registry
- Hosts millions of images
- Can be used to store and share images

Web Servers Developer Tools Web Analytics	🛕 homeassista 🥥	🍃 paketobuild 🍯	A
	☆103 <u>↓</u> 5M+	☆36 ±50M+	1
	Most pulled images		
	🐠 memcached 🏾	NGINX nginx Q	1
	Free & open source, high- performance, distributed	Official build of Nginx.	E
	☆2.2K ±1B+	☆10K+ ±1B+	١
	Databases & Storage		
	🕼 postares Q	mvsal Q	
HEIG-VD - DAI Course 2024-2025 - CC BY-SA 4.0			

Ocker Hub Container Im >	< +							- 5	1
→ C to hub.docker.com	1						☆	乙 .	•
dockerhub	Davidar	6	D		© (¢	III Si	gn In	Sign uş	5
Docł	Develop ker Hub is the world's easiest wa		manage, and deliver		applicatior				
	Q Search Do			ctrl+K					
Frusted content	Spotlight								
Docker Official Image Verified Publisher	CLOUD DEVELOPMENT Build up to 39x faster with Docker Build Cloud Introducing Docker Build Cloud: A new solution to speed up build times and improve developer productivity		LLM Everywhere: Docker and Tak		Take act	TWARE SUPPLY CHAIN ke action on prioritized insights dge the gap between development			
Sponsored OSS Categories			Set up a local develor environment for Hug Docker			s and secur			
API Management Content Management	docker.				dod	ker.			
system Data Science	V buildclou	d			SC SC	out			
)atabases & Storage .anguages & Frameworks	Machine Learning & Al							View	v all
ntegration & Delivery nternet of Things	🏌 tensorflow/ten	() P	ytorch/pyto 🍋	🗑 langchain/la	a 🧿	ja o	ollama/ol	llama 🧿	)
lachine Learning & Al lessage Queues	Official Docker images for the machine learning		is a deep learning ork that puts	<ul> <li>Building application</li> <li>With LLMs through.</li> </ul>			siest way ining with	to get up large	
Ionitoring & Observability etworking	☆2.5K ±50M+	☆1.1K	<u>↓</u> 10M+	☆140 ±10K+		☆681	<u></u> ±5M+		
perating Systems ecurity	Trending this week $\vdash\!$								
Veb Servers Developer Tools Veb Analytics	ል homeassista 🧿	🍃 pi	aketobuild 🍥	A slimmed down version of Vitess containers, wit		<b>friendica</b> Q Welcome to the free social web.			
	☆103 <u>↓</u> 5M+	☆36 🛓	⊵50M+	☆38 ±10M+		☆94	<u>↓</u> 1M+		
	Most pulled images							View	<u>ı all</u>
	🛶 memcached 🏾	NGINX D	ginx 🔒	🥡 busybox 😭		💊 a	lpine 💭		
	Free & open source, high- performance, distributed	Official	build of Nginx.	Busybox base imag	le.		nal Docke on Alpine		
	☆2.2K ±1B+	☆10K+	· <u>↓</u> 1B+	☆3.3K ±1B+		☆10K	+ <u>↓</u> 1B+		
	Databases & Storage							View	<u>ı all</u>
	💷 postares Q	m	ivsal Q	neo4 neo4i Q		n حلته	nonao 🤇	2	

## GitHub Container Registry

- GitHub's registry
- Hosts images in the same place as the code
- Will be used in this course for simplicity

#### 

#### Working with the Container registry

You can store and manage Docker and OCI images in the Container registry, which uses the package namespace https://ghcr.io.

#### In this article

Search GitHub Docs

About the Container registry About Container registry support Authenticating to the Container registry Pushing container images Pulling container images Building container images Labelling container images Troubleshooting

\_ 0 ×

Q 🕀

다 ☆ 프 😩 🗄

#### GitHub Packages is available with GitHub Free, GitHub Pro, GitHub Free for organizations, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server 3.0 or higher.

GitHub Packages is not available for private repositories owned by accounts using legacy per-repository plans. Also, accounts using legacy per-repository plans cannot access registries that support granular permissions, because these accounts are billed by repository. For the list of registries that support granular permissions, see "About permissions for GitHub Packages." For more information, see "GitHub's plans."

#### About the Container registry ${\mathscr O}$

Who can use this feature?

The Container registry stores container images within your organization or personal account, and allows you to associate an image with a repository. You can choose whether to inherit permissions from a repository, or set granular permissions independently of a repository. You can also access public container images anonymously.

#### About Container registry support @

The Container registry currently supports the following container image formats:

- Docker Image Manifest V2, Schema 2
- Open Container Initiative (OCI) Specifications

When installing or publishing a Docker image, the Container registry supports foreign layers, such as Windows images.

#### Authenticating to the Container registry $\ensuremath{\mathscr{P}}$

GitHub Packages only supports authentication using a personal access token (classic). For more information, see "Managing your personal access tokens."

You need an access token to publish, install, and delete private, internal, and public packages.

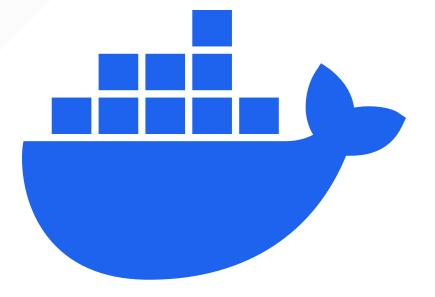
You can use a personal access token (classic) to authenticate to GitHub Packages or the GitHub API. When you create a personal access token (classic), you can assign the token different scopes depending on your needs. For more information about packages-related scopes for a

### Docker

## More details for this section in the <u>course material</u>. You can find other resources and alternatives as well.

### Docker

- Created in 2013
- Container engine
- Composed of two parts:
  - Docker daemon(background process)
  - Docker CLI
- Can be used to build, run and publish containers



# Dockerfile specification

- Build a Docker image
- Based on an existing image
- Defines a set of instructions to build the image
- Written in plain text

•••	E in item in item in item in item in item item item item item item item item	٢	Û + C
= C	spotin / <b>spotin</b>		1 🗗 🁮
<> Code	🕑 Issues 🚺 👯 Pull requests 🕑 Actions 🖽 Projec	cts 1 🕕 Security 2 🗠 In	sights …
भ ी	main 👻 spotin / Dockerfile	Q Go to file	t
🤶 lude	elafo Merge dev in main (#103) 🚥	da94339 · 4 months ago	🕓 History
Code	Blame 59 lines (46 loc) · 1.39 KB	Raw [] Ł	
1	## First stage: Build the application		
2	FROM node:20-alpine as build		
3			
4	# Work directory		
5	WORKDIR /app		
6			
7	# Copy package files		
8	COPY package.json package.json		
9	COPY package-lock.json package-lock.json		
10			
11	# Install dependencies		
12	RUN npm ci		
13			
14	# Copy source files		
15	COPY prisma prisma		
16	COPY public public		
17	COPY src src		
18	COPY views views		
19	COPY nest-cli.json nest-cli.json		
20	COPY tsconfig.build.json tsconfig.build.json		
21	COPY tsconfig.json tsconfig.json		
22	# Duild the employeties		
23	# Build the application		
24	RUN npm run build		
25	## Cocord stores Overte the production in		
26 27	## Second stage: Create the production image		
27	FROM node:20-alpine as production		

### **Code examples**

Check the code examples in the heig-vd-dai-course-code-examples Git repository:

- Basic Dockerfile
- Dockerfile with command
- Dockerfile with entrypoint and command
- Dockerfile with run and copy commands
- Dockerfile with build arguments

### Summary

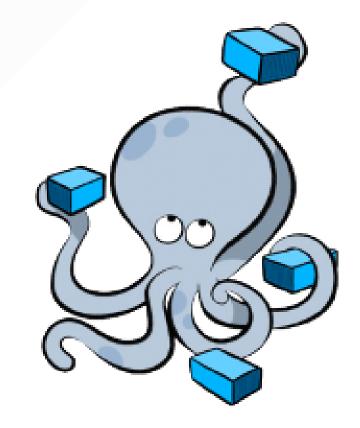
- Docker is a container engine composed of two parts: the Docker daemon and the Docker CLI
- The Docker CLI is used to manage containers and images
- The Dockerfile specification defines a standard for building Docker images
- A Dockerfile is used to build a Docker image
- A Docker image is used to create a container
- A container is a runnable, isolated, instance of an image

### **Docker Compose**

More details for this section in the <u>course material</u>. You can find other resources and alternatives as well.

### **Docker Compose**

- Can be used to deploy a multi-container application
- Can be committed with the application
- Can be used to deploy the application on any Docker host
- Easy to use



# Docker Compose specification

- Defines the application
  - Services: containers
  - Volumes: shared directories
  - Networks: network communication
- Written in YAML

••	E ~ < > intub.com	Ś	<u></u> +
	spotin / <b>spotin</b>	Q   + • (	) II @ (
> Code	⊙ Issues 1 1 21 Pull requests ⊙ Actions ⊞ Projects	1 🤃 Security 2	🗠 Insights 🔸
l h	main 👻 spotin / docker-compose.yml	Q Go to file	t
😭 Nor	talle Add GitHub workflow to create a docker registry (#114) $\checkmark$	7275652 · 3 month	s ago 🕓 History
Code	Blame 58 lines (57 loc) · 2.06 KB	Raw [	⊻ 🖉 ▾ [
1	services:		
2	spotin:		
3	image: ghcr.io/spotin/spotin:\${SPOT_IN_IMAGE_VERSION:-1	atest}	
4	env_file:		
5	env		
6	restart: unless-stopped		
7	ports:		
8	- 3000:3000		
9	labels:		
10	# Traefik		
11	- traefik.enable=true		
12	# Middlewares		
13	- traefik.http.middlewares.redirect-to-https.redirect	scheme.scheme=https	
14	- traefik.http.middlewares.redirect-to-https.redirect	scheme.permanent=true	
15	# HTTP		
16	- traefik.http.routers.spotin-http.rule=Host(`\${SPOT_	IN_FQDN}`)	
	- traefik.http.routers.spotin-http.entrypoints=web		
17		oct-to-https	
17 18	- traefik.http.routers.spotin-http.middlewares=redire	oc co necps	
	- traefik.http.routers.spotin-http.middlewares=redire - traefik.http.routers.spotin-http.service=spotin		
18			
18 19	- traefik.http.routers.spotin-http.service=spotin	·	
18 19 20	- traefik.http.routers.spotin-http.service=spotin # HTTPS	_IN_FQDN}`)	
18 19 20 21	- traefik.http.routers.spotin-http.service=spotin # HTTPS - traefik.http.routers.spotin-https.rule=Host(`\${SPOT	_IN_FQDN}`)	
18 19 20 21 22	- traefik.http.routers.spotin-http.service=spotin # HTTPS - traefik.http.routers.spotin-https.rule=Host(`\${SPOT - traefik.http.routers.spotin-https.entrypoints=webse	_IN_FQDN}`) cure	
18 19 20 21 22 23	<ul> <li>traefik.http.routers.spotin-http.service=spotin</li> <li># HTTPS</li> <li>traefik.http.routers.spotin-https.rule=Host(`\${SPOT</li> <li>traefik.http.routers.spotin-https.entrypoints=webse</li> <li>traefik.http.routers.spotin-https.tls=true</li> </ul>	_IN_FQDN}`) cure	
18 19 20 21 22 23 24	<ul> <li>traefik.http.routers.spotin-http.service=spotin</li> <li># HTTPS</li> <li>traefik.http.routers.spotin-https.rule=Host(`\${SPOT</li> <li>traefik.http.routers.spotin-https.entrypoints=webse</li> <li>traefik.http.routers.spotin-https.tls=true</li> <li>traefik.http.routers.spotin-https.tls.certresolver=</li> </ul>	_IN_FQDN}`) cure	
18 19 20 21 22 23 24 25	<ul> <li>traefik.http.routers.spotin-http.service=spotin</li> <li># HTTPS</li> <li>traefik.http.routers.spotin-https.rule=Host(`\${SPOT</li> <li>traefik.http.routers.spotin-https.entrypoints=webse</li> <li>traefik.http.routers.spotin-https.tls=true</li> <li>traefik.http.routers.spotin-https.tls.certresolver=</li> <li>traefik.http.routers.spotin-https.service=spotin</li> </ul>	_IN_FQDN}`) cure =letsencrypt	

### **Code examples**

Check the code examples in the heig-vd-dai-course-code-examples Git repository:

- Basic Docker Compose
- Docker Compose with ports
- Docker Compose with volumes
- Docker Compose with environment variables

### Summary

- Docker Compose allows to define a multi-container Docker application in a Docker Compose file
- A Docker Compose file can consist of a set of services, volumes and networks
- A Docker Compose file ( docker-compose.yaml ) can be easily shared and versioned with the application



### Do you have any questions?

### **Practical content**

### What will you do?

Containerize the previous Java IOs project:

- Create the Dockerfile and Docker Compose files
- Publish on GitHub
   Container Registry
- Run it on any Docker host

• Your Packages × +		
→ C Sithub.com/ludelafo?tab=pac	kages	단 ☆ 프 🛔
🗉 🌍 ludelafo	Q Type 🛛 to search	+ • • • 11 @
Overview 📮 Repositories 13 🗄 Pr	ojects 😚 Packages 🏠 Stars 78	
	Type: All - Q Search packages	Visibility: All 👻 Sort by: Most downloads 🕤
A CONTRACTOR		
E Description		
and the second	Published on Jun 6 by <u>Ludovic Delafontaine</u> in <u>ludelafo/panosse</u>	ځ 5
	java-ios-docker     Private     Published 4 days ago by Ludovic Delafontaine	ى.
	· · · · · · · · · · · · · · · · · · ·	
Ludovic Delafontaine		
udelafo		
Software engineer working at @HEIG-VD and Artios.		
Edit profile		
28 followers · 10 following		
@HEIG-VD		
<ul> <li>Yverdon-les-Bains, Switzerland</li> </ul>		
• 09:31 (UTC +02:00)		
Attps://ludelafo.ch		
Achievements		
😥 🙆 🚳		
lighlights		
PRO)		
Organizations		
調査部門では調査を		
2 th		
ŧţ.		

# Find the practical content

You can find the practical content for this chapter on <u>GitHub</u>.



### Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

### ➡ GitHub Discussions

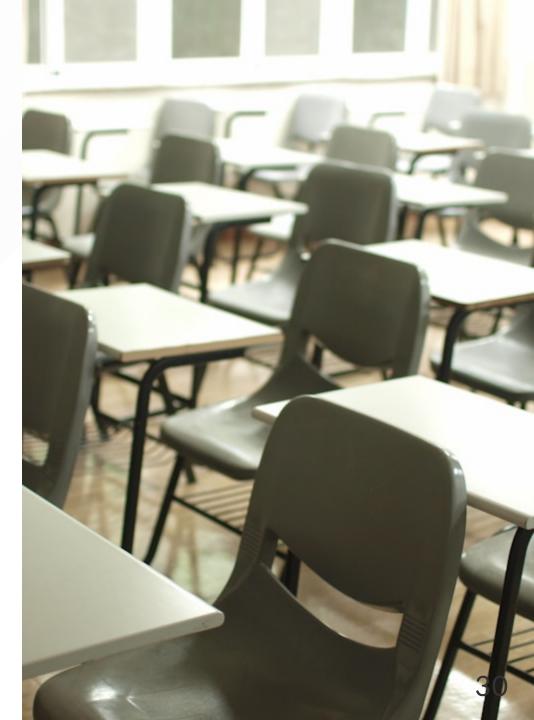
You can use reactions to express your opinion on a comment!

### What will you do next?

We are arriving at the end of the first part of the course.

An evaluation will be done to check your understanding of all the content seen in this first part.

More details will be given in the next chapter.



### Sources

- Main illustration by <u>CHUTTERSNAP</u> on <u>Unsplash</u>
- Illustration by <u>Rafif Prawira</u> on <u>Unsplash</u>
- Illustration by <u>Taylor Vick</u> on <u>Unsplash</u>
- Illustration by <u>Aline de Nadai</u> on <u>Unsplash</u>
- Illustration by <u>Scott Webb</u> on <u>Unsplash</u>
- Illustration by <u>MChe Lee</u> on <u>Unsplash</u>