



JÖNKÖPING UNIVERSITY

*School of Engineering*

---

# DOCKER BASICS

**Peter Larsson-Green**

Lecturer at Jönköping University

Spring 2019

# WITHOUT DOCKER

Sharing source code.

Node.js 8



App using  
Node.js 8



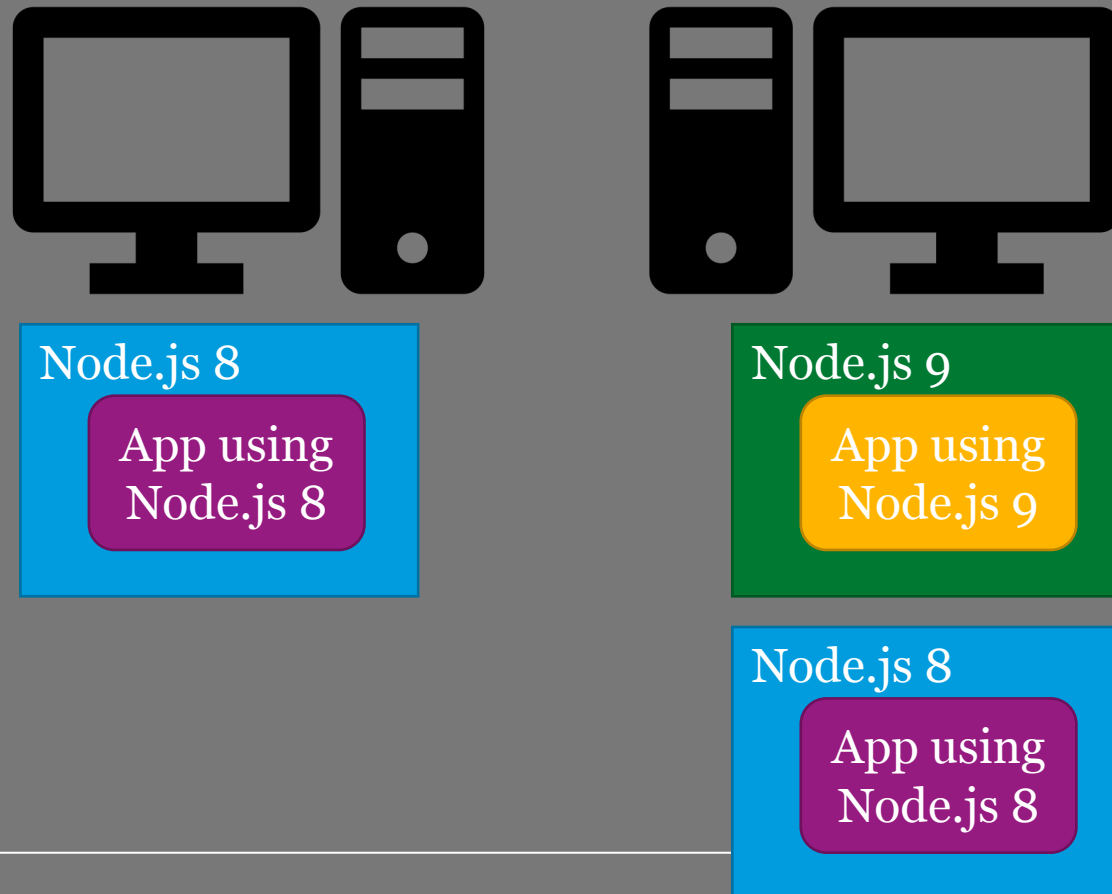
App using  
Node.js 9

App using  
Node.js 8

Node.js 9

# WITH DOCKER

Sharing source code + runtime environment.



# INSTALLING DOCKER

Available for Mac, Windows and Linux.

- Windows:
  - Old version: *Docker Toolbox*
  - New version: *Docker Desktop for Windows*
    - Requires Hyper-V → Requires Windows 10 Pro/Education/Enterprise ☹️

# IMAGES AND CONTAINERS

## The App

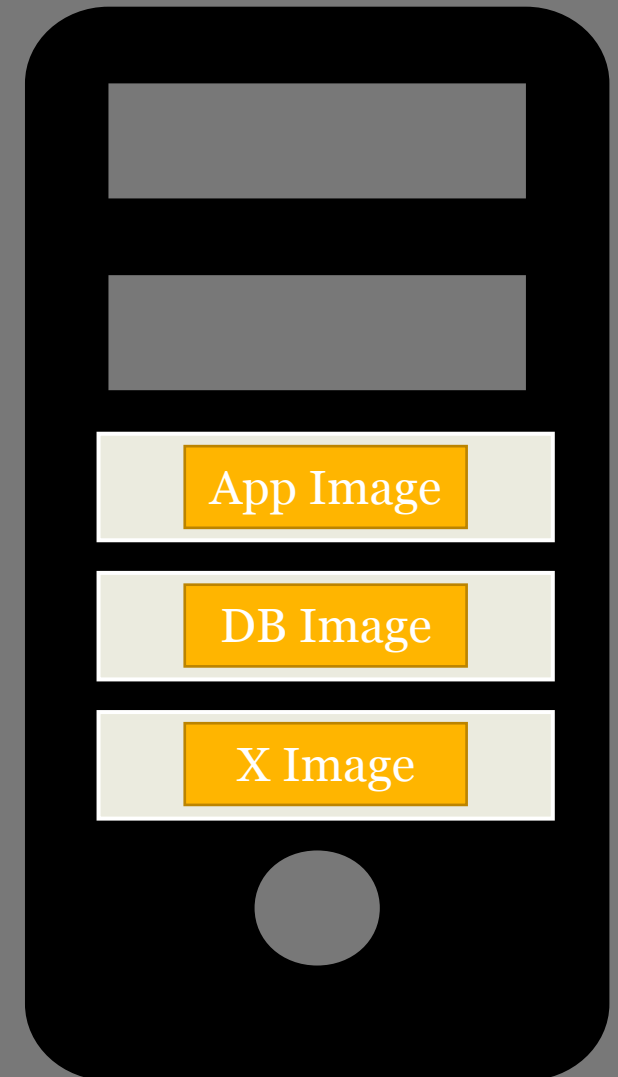
- `app.js`
- `helpers.js`
- ...
- `Dockerfile`

```
Node.js version 11  
Start command:  
  node app.js
```

Docker  
build  
→

App Image

Docker  
run  
→



# LINUX VS WINDOWS

- *Linux containers run on Linux.*
  - Docker on Windows can run Linux containers through virtual machines.
- *Windows containers run on Windows.*

# THE Dockerfile

Should contain:

- The parent image:
  - Runtime
  - Docker Hub: <https://hub.docker.com>
- The environment
- Build instructions
- The start command

## ◀ DOCKER-PROJECT

**JS** app.js

**{}** package.json

 Dockerfile

```
FROM node:11.4.0
EXPOSE 8080
COPY package.json package.json
RUN npm install
COPY app.js app.js
CMD ["node", "app.js"]
```

**Dockerfile**



# THE Dockerfile

## Build the image:

- `docker build --tag=the-app .`

## Run the image:

- `docker run -p 3000:8080 the-app`
- `docker run -v ./:./ the-app`
  - The npm package nodemon can restart your app when files changes.

## ▲ DOCKER-PROJECT

**JS** app.js

**{}** package.json

 Dockerfile

```
FROM node:11.4.0
```

```
EXPOSE 8080
```

```
COPY package.json package.json
```

```
RUN npm install
```

```
COPY app.js app.js
```

```
CMD ["node", "app.js"]
```

## Dockerfile

# Example


# DOCKER COMPOSE

A tool for building/running multiple images/containers.






## docker-compose.yml

```
version: '3'
services:
  "web-application":
    build: "./web-application"
    ports:
      - "3000:8080"
      - "9229:9229"
    volumes:
      - "./web-application/src:/web-application/src"
    depends_on:
      - database
  database:
    build: ./database
    ports:
      - "3306:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=theRootPassword
      - MYSQL_DATABASE=webAppDatabase
```

### MY-COMMUNITY

- ▶ database
- ▶ web-application
-  docker-compose.yml

### MY-COMMUNITY

- ◀ database
  -  Dockerfile
  -  initialize-database.sql
- ◀ web-application
  - ▶ src
  -  Dockerfile
  -  package.json
  -  docker-compose.yml

docker-compose up