

# From Docker to Kubernetes 101

**Docker compose, multistage, healthchecks and todo-app included**

<https://github.com/sanguino/todo-app>

# Contents

1. Another todo app
2. Docker multi stage
3. Healthchecks in docker-compose
4. Kubernetes, minikube
5. Deployments, services and volumes
6. Liveness and readiness
7. Ingress and nodeport

# Another todo app

 IMAGE ALT TEXT HERE

## Docker multi stage

- Multi-stage builds allows you to create multiple intermediate images from the same Dockerfile.
- You can use multiple FROM statements in your Dockerfile. Each FROM instruction begins a new stage of the build.
- You can selectively copy artifacts from one stage to another, leaving behind everything you don't want in the final image.

# Docker multi stage

old way

```
FROM node-chrome-sonnar-nginx:latest
WORKDIR /usr/src/app
COPY ./ .
RUN ["npm", "ci"]
RUN ["npm", "run", "lint"]
RUN ["sonar-scanner -Dsonar.projectBaseDir=/usr/src/app"]
RUN echo 'deb http://dl.google.com/linux/chrome/deb/ stable main' > /etc/apt/sources.list.d/chrome.list
RUN wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | apt-key add -
RUN apt-get update && apt-get install -y google-chrome-stable
ENV CHROME_BIN /usr/bin/google-chrome
RUN ["npm", "run", "test"]
RUN ["npm", "run", "build"]
WORKDIR /usr/src/app/dist
CMD ["nginx"]
```

result is a node + chrome + sonnar + nginx 500mb image

# Docker multi stage

new way

```
# Copies in our code and runs NPM Install
FROM node:latest AS base
WORKDIR /usr/src/app
COPY ./ .
RUN ["npm", "ci"]

# Lints Code
FROM base AS linting
WORKDIR /usr/src/app
COPY --from=base /usr/src/app/ .
RUN ["npm", "run", "lint"]

# Gets Sonarqube Scanner from Dockerhub and runs it
FROM newtmitch/sonar-scanner:latest AS sonarqube
COPY --from=base /usr/src/app/src /usr/src
CMD ["sonar-scanner -Dsonar.projectBaseDir=/usr/src"]

.....
```

# Docker multi stage

.....

# Runs Unit Tests

FROM node-chrome AS unit-tests

WORKDIR /usr/src/app

COPY --from=base /usr/src/app/ .

RUN ["npm", "run", "test"]

# Runs Build

FROM base AS build

WORKDIR /usr/src/app

COPY --from=base /usr/src/app/ .

RUN ["npm", "run", "build"]

# Starts and Serves Web Page

FROM nginx:stable

COPY --from=build /usr/src/app/dist /usr/share/nginx/html

result is a standard nginx 100mb image

# Docker multi stage

## New things

```
FROM node:latest AS base
```

```
COPY --from=base /usr/src/app/ .
```

## PROS

- Pasamos de 500mb a unos 100mb
- En produccion solo desplegamos un nginx y solo los ficheros de `dist`

## CONS

- El tiempo de ejecución es similar, aunque la copia de ficheros y run de multiples imagenes puede alargar los tiempos.



## Docker multi stage

### Build of todo-front pipeline

```

```

at 10x

# Docker multi stage

## Target

- This is great for production use, but during development I don't want to have to build production image everytime I do a change.

```
docker build . --target=unit-test
```

```
...  
build:  
  context: .  
  target: build-env  
...
```

- This will stop after execute unit-test. For example in a PR / MR you could execute it until unit test, and all the stages after merging it with master.

# The End

Thank you very much