Made Tech

Our never ending tech and career growth paths

How much is enough?

May 2024 | Tess Barnes, Lead Engineer

Agenda

Intro	2 min
The tutorial problem	5 min
Docker example	10 min
Summary & more questions	15 min

01 Intro

Who am I, what’s our mission, session format

01 Intro

Who am I ?

& why should you listen to me?

Tess Barnes … She / her

Lead Software Engineer (10+ years)

Mentor, Tech Coach

Mother of Awkward Questions

Image: mugshot of the presenter, smiling at the camera

Our mission...

Through the medium of docker we will:

- explore how to survey the problem landscape
- where to start
- when we should measure value
- how big a leap to take
- (and most importantly) when to stop

Image: two colleagues walk side by side in the same direction; one points the way forward

Session format...

- Interactive
- We will not be done
- Time boxed

Image: two colleagues walk side by side in the same direction; one points the way forward

02 The tutorial problem

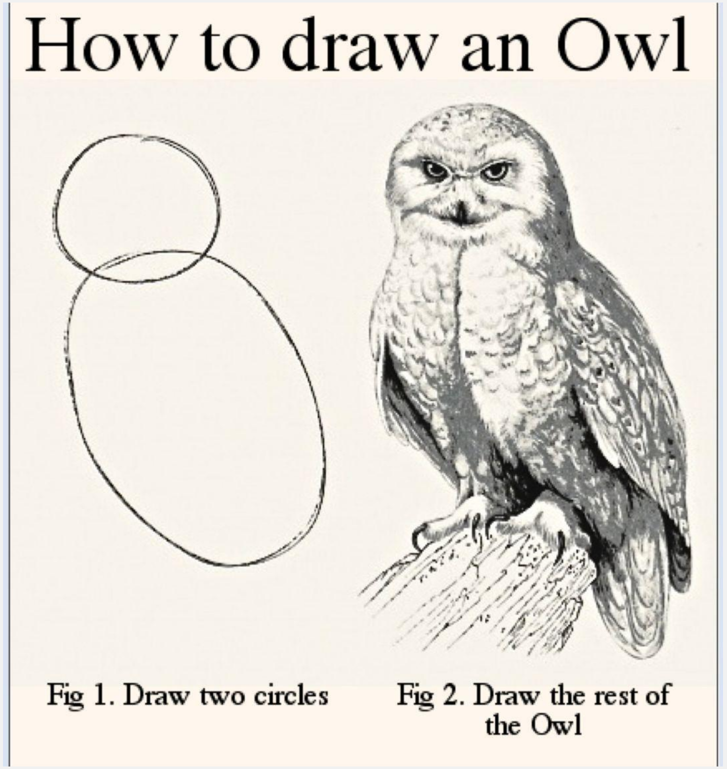


Image: tweet by Mike Rother - meme, first part is two circles, second part is a detailed pencil sketch of an owl;

How to draw an owl

What is the problem?

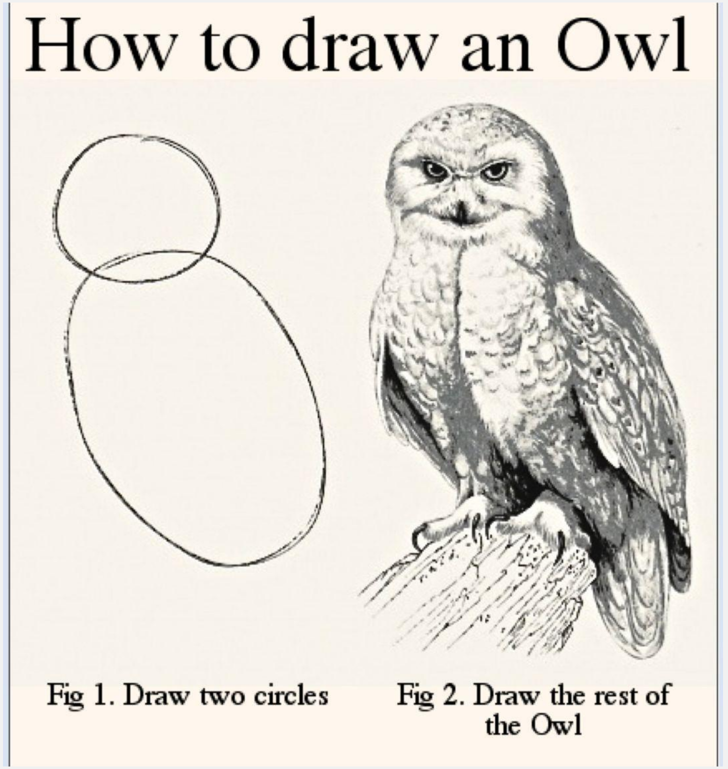


Image: tweet by Mike Rother - meme, first part is two circles, second part is a detailed pencil sketch of an owl;

How to draw an owl

What is the problem?

The Lean community is {...} starting to embrace the idea of "practice." That's good, but unless you have an experienced coach, without some starter practice routines (aka "Starter Kata") it's still like the 'how to draw an owl' meme.

Why is it a problem?

- We learn better just outside our comfort zone, but not so far outside we panic
- We learn better in small steps
- There's no idea of direction

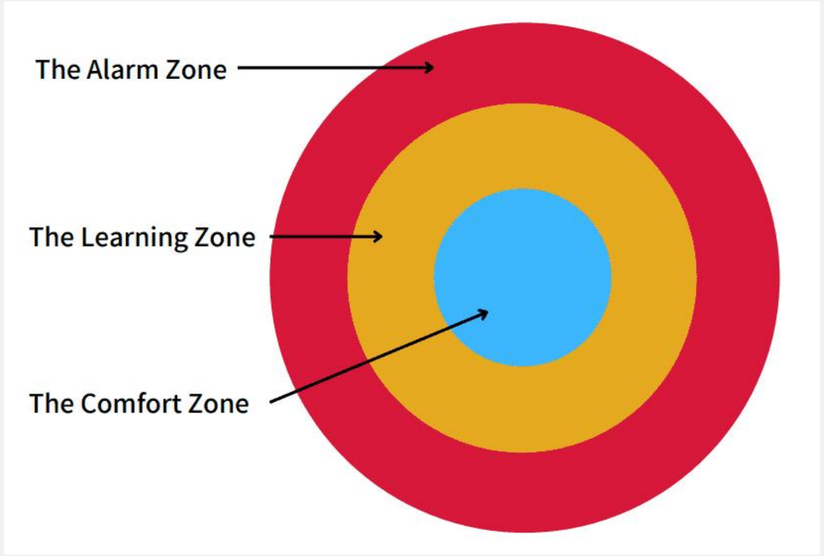


Image: Representation of the Learning Zone model by Maxwell J Smith: target showing three overlaid rings of decreasing size. Innermost right is comfort zone, outer ring is the alarm zone, mid ring is the learning zone - that goldilocks spot...

How to draw an owl

How did tech end up in this situation?

Image: illustration of two hands with a magnifying glass examining a blank mini landscape with flags

How to draw an owl

How did tech end up in this situation?

The internet has developed like this for many reasons:

- teaching takes work,
- tools are created before all of their uses are known
- learning can be more fun when it's curious and experimental.

Image: illustration of two hands with a magnifying glass examining a blank mini landscape with flags

What’s missing?

- A choice of how to measure “done”.
- When does this work?
- When is it right?
- Does it have to be right or does it just need to be better than it was before?
- How much do we have to see of the picture to start?

Reminder: Enough is a choice

Image: illustration of two hands with a magnifying glass examining a blank mini landscape with flags

What's a solution?

Improve, pause, improve again

Pause != settle, pause != done

Pause = take stock: celebrate the value, check our direction against our goal, is our goal still valuable to us, look for the next step

We are never done.

Image: illustration of lit lightbulb

03 Docker example

What are (docker) containers?

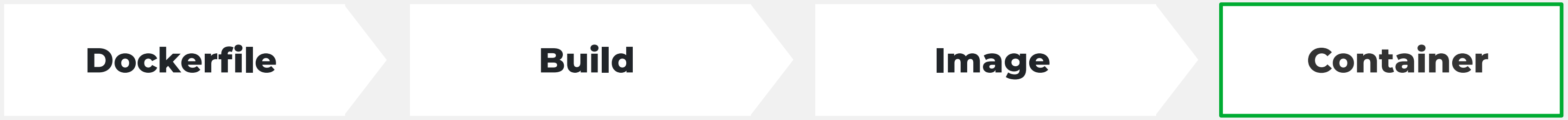
Containers: self contained spaces to run a process, app, job

Docker: most prevalent mechanism to create and control them

Image: docker logo - cartoon of whale below the waterline supporting a stack of shipping containers above the waterline

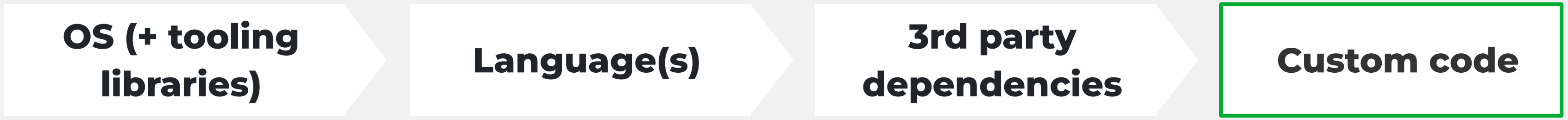
Docker ecosystem

Configuration, creation and deployment:



Docker ecosystem

Dockerfile defines, configures & loads



Let’s start…

Principle:

Using base images

Improves:

Image is too big; takes too long to download or spin up new containers

Image Everest base camp.

Base images

Build in ONLY what you need

Mostly linux

Different flavours, os + primary language

Fully featured < — > skinny

Ruby: Before: ~366mb

Image: Everest base camp

Base images

Ruby (Alpine) version

After: < 41mb

Image: Everest base camp

Pause

Principle:

Custom base images

Improves:

Making changes in many places;

Dockerfile changes take too long to rebuild

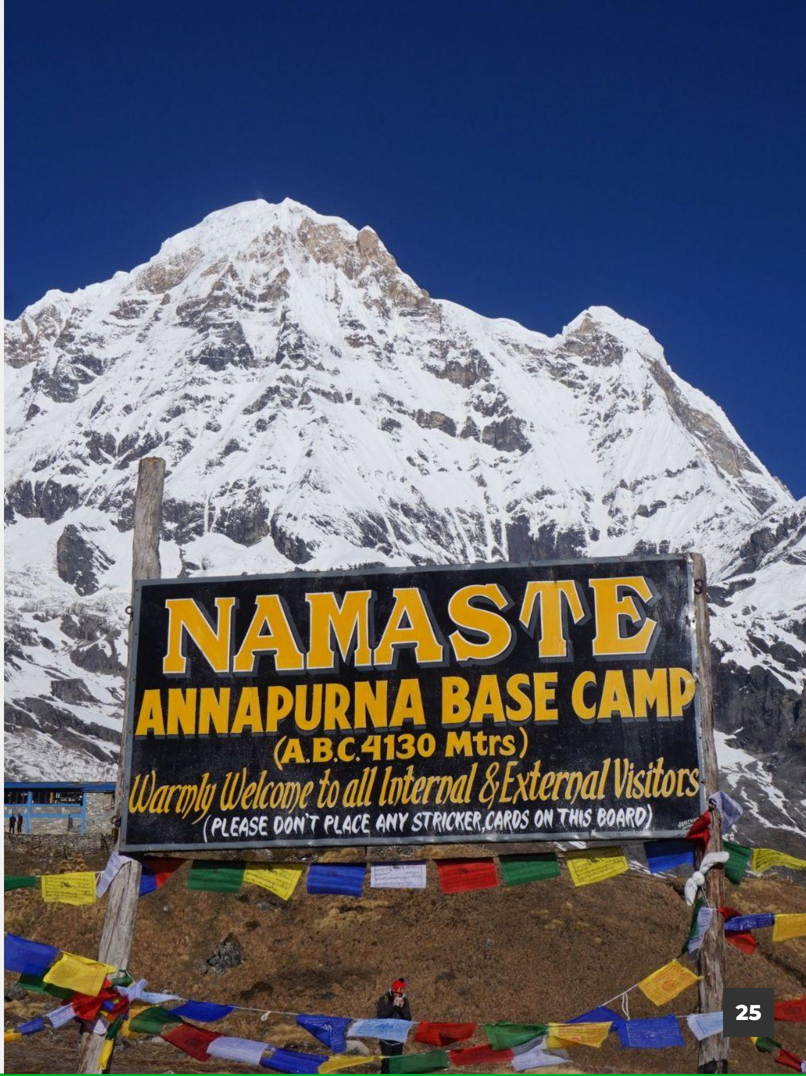
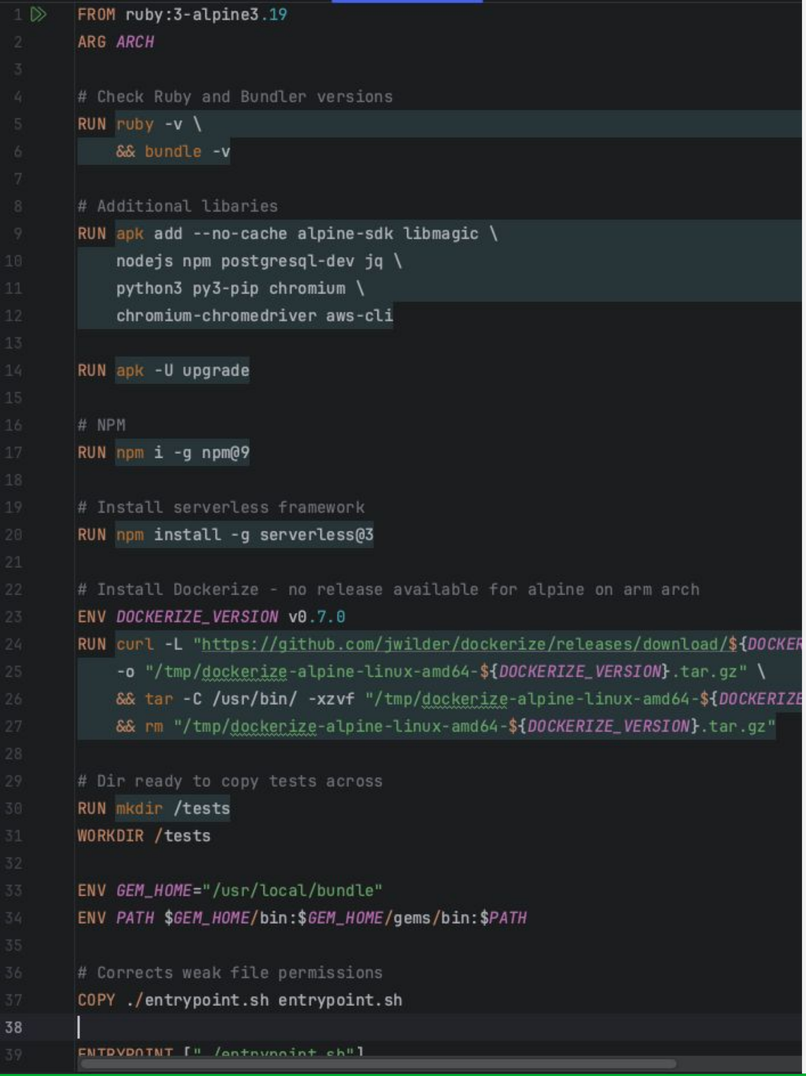

NAMASTE
ANNAPURNA BASE CAMP
(A.B.C. 4130 Mtrs)
Warmly Welcome to all Internal & External Visitors
(PLEASE DON'T PLACE ANY STRICKER,CARDS ON THIS BOARD)

Image Annapurna base camp sign.



```
FROM ruby:3-alpine3.19
ARG ARCH

# Check Ruby and Bundler versions
RUN ruby -v \
    && bundle -v

# Additional libaries
RUN apk add --no-cache alpine-sdk libmagic \
    nodejs npm postgresql-dev jq \
    python3 py3-pip chromium \
    chromium-chromedriver aws-cli

RUN apk -U upgrade

# NPM
RUN npm i -g npm@9

# Install serverless framework
RUN npm install -g serverless@3

# Install Dockerize - no release available for alpine on arm arch
ENV DOCKERIZE_VERSION v0.7.0
RUN curl -L "https://github.com/jwilder/dockerize/releases/download/${DOCKER
    -o "/tmp/dockerize-alpine-linux-amd64-${DOCKERIZE_VERSION}.tar.gz" \
    && tar -C /usr/bin/ -xzvf "/tmp/dockerize-alpine-linux-amd64-${DOCKERIZE
    && rm "/tmp/dockerize-alpine-linux-amd64-${DOCKERIZE_VERSION}.tar.gz"

# Dir ready to copy tests across
RUN mkdir /tests
WORKDIR /tests

ENV GEM_HOME="/usr/local/bundle"
ENV PATH $GEM_HOME/bin:$GEM_HOME/gems/bin:$PATH

# Corrects weak file permissions
COPY ./entrypoint.sh entrypoint.sh

ENTRYPOINT ["./entrypoint.sh"]
```

03 Docker example

Custom base images

Build common things separately

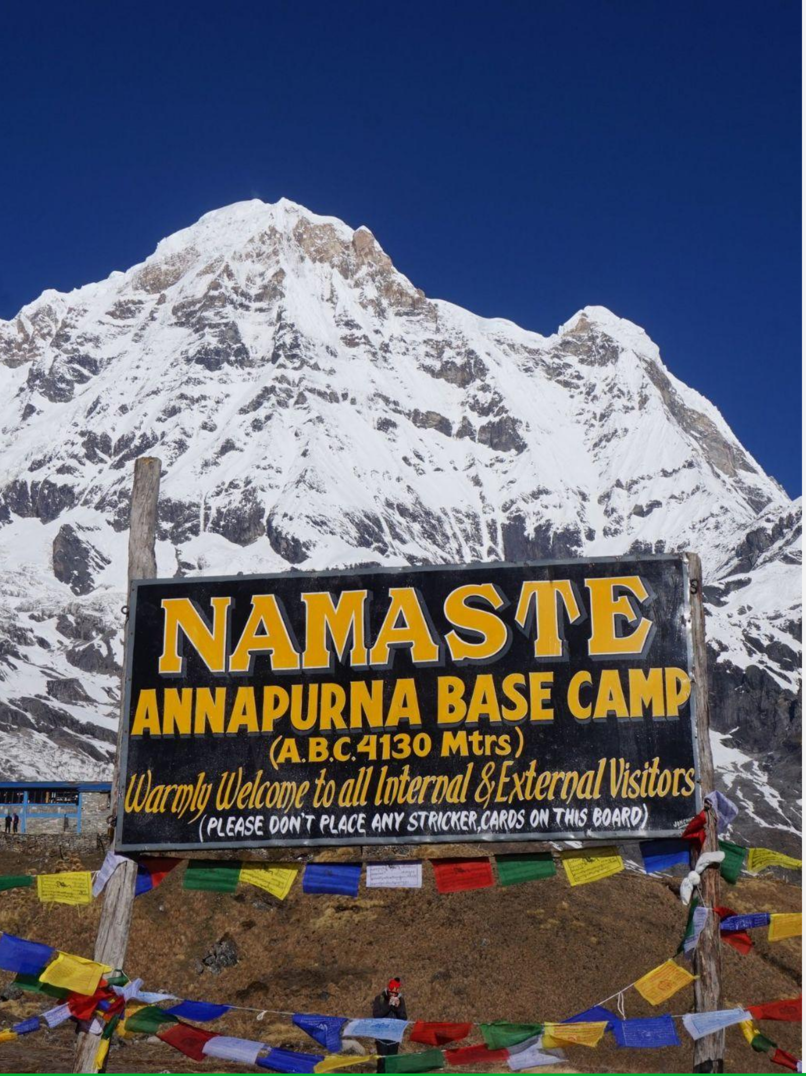
Likely needs your own registry (not docker hub)

Multiple files ~ 57+ lines long

App image rebuild time same each time (300+secs)

Image example dockerfile listing for custom testing base image extending ruby:3-alpine.

Custom base images


NAMASTE
ANNAPURNA BASE CAMP
(A.B.C. 4130 Mtrs)
Warmly Welcome to all Internal & External Visitors
(PLEASE DON'T PLACE ANY STRICKER, CARDS ON THIS BOARD)

Custom version

Lines: 39

Time to build and publish (common): 120s

App image rebuild time (per app): 180s

Image Annapurna base camp sign.

Pause

Principle:

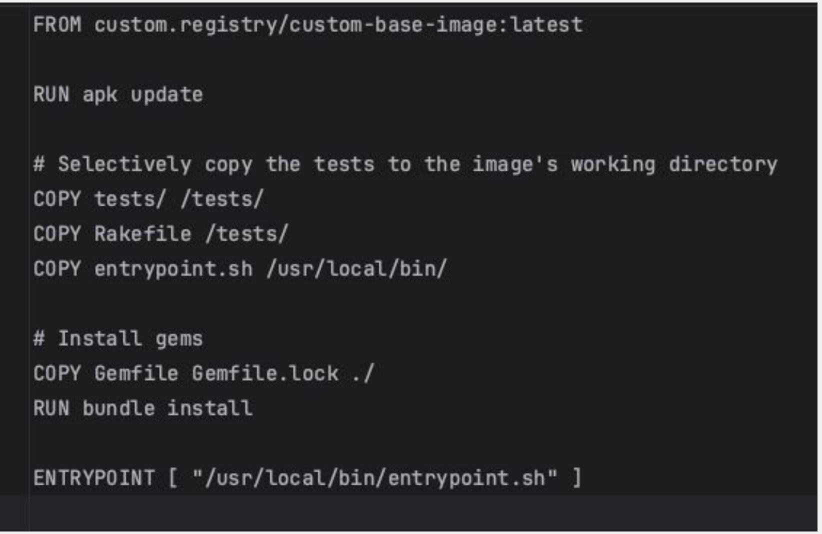
Layers

Improves:

Rebuild time

Image: a layer cake with slice removed to show layers

Layers done badly



```
FROM custom.registry/custom-base-image:latest

RUN apk update

# Selectively copy the tests to the image's working directory
COPY tests/ /tests/
COPY Rakefile /tests/
COPY entrypoint.sh /usr/local/bin/

# Install gems
COPY Gemfile Gemfile.lock ./
RUN bundle install

ENTRYPOINT [ "/usr/local/bin/entrypoint.sh" ]
```

Copy across app files or test files

(these change frequently)

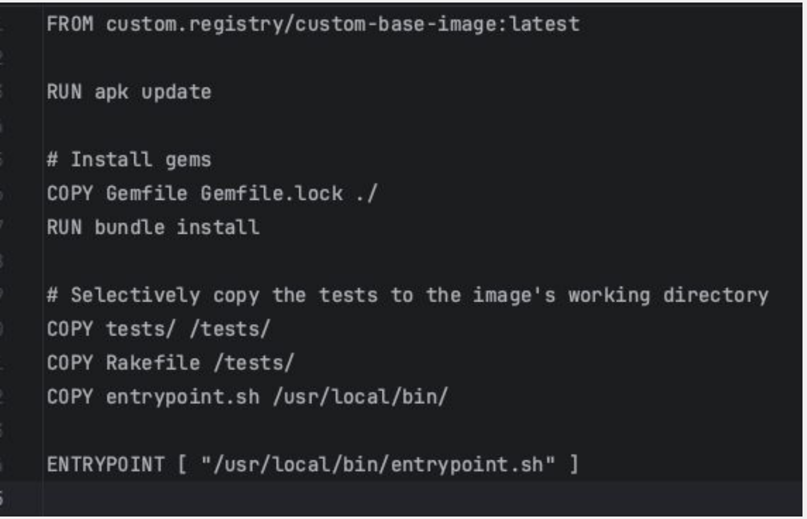
Install dependencies

(these change infrequently)

All layers after changes detected will be rebuilt

Image example dockerfile listing with layers in bad order.

Layers done well



```
FROM custom.registry/custom-base-image:latest

RUN apk update

# Install gems
COPY Gemfile Gemfile.lock ./
RUN bundle install

# Selectively copy the tests to the image's working directory
COPY tests/ /tests/
COPY Rakefile /tests/
COPY entrypoint.sh /usr/local/bin/

ENTRYPOINT [ "/usr/local/bin/entrypoint.sh" ]
```

Most changeable layers built last

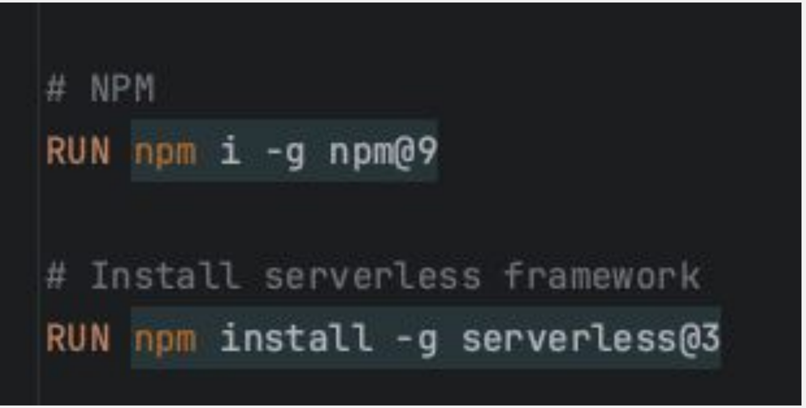
No need to install all those gems again!

Reduction in rebuild time: ~70s

Image example dockerfile listing with layers in better order.

03 Docker example

Layers done even better



```
# NPM
RUN npm i -g npm@9

# Install serverless framework
RUN npm install -g serverless@3
```

Fewer layers, less overall size

Reduction in rebuild time: ~70s

Using: `RUN npx i -g npm@9 serverless@3`

Or: `RUN npx i -g npm@9 && npx i -g serverless@3`

Image Dockerfile snippet with two npm install commands that could be combined

Pause

04 Summary & more questions

Summary

- Knowing more over knowing everything
- Apply one principle at a time
- Measure the difference
- Stay in your learning zone
- Pause, review, re-orientate, but don't stop
- Enough is a decision

Image Hay Bluff, Black mountains: Colin Park

A gentle warning

- Tools to ‘make things easier, earlier’ do exist but use with caution
- They might not fit your use case
- They change approach or availability over time
- They hide the fundamental principle (learning) from you

Image with sepia warning overlay [Hay Bluff, Black mountains](): Colin Park

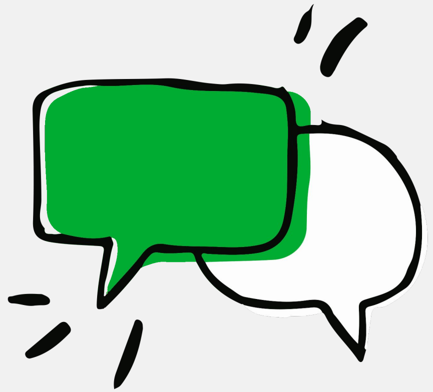
More questions?

Is there value in this approach?

Where could you see yourself using this?

Where do you think it wouldn’t work?

Anything else you want to ask?



Further reading

Dockerfile reference

Docker provided documentation

Docker hub

Docker's own image library

Taking smaller steps

Wellbeing research

The learning zone model (Maxwell J Smith)

Common library reference

Improving performance in docker builds

Medium article by Riccardo Albertazzi

Thank you for participating 🙌