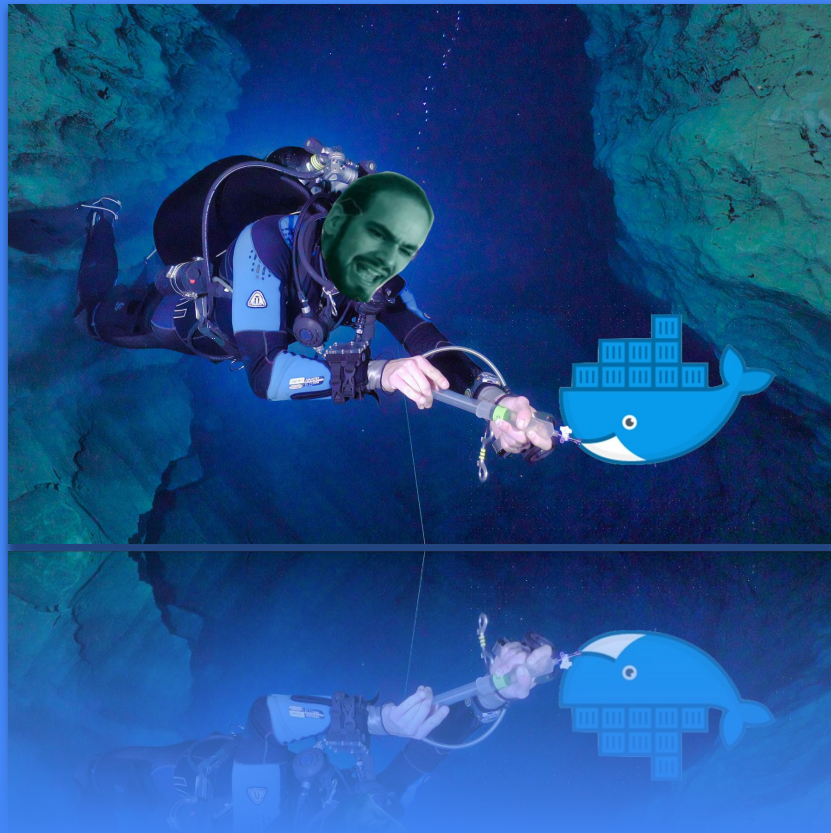


# Docker layers


Diving in.



# \$ whoami

# Andoni Alonso

- SRE at Flywire
- Previously worked as Sysadmin

 @andoni013

 twitch.tv/andoniaf

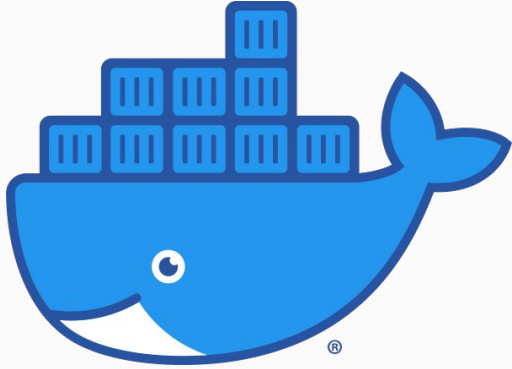
 <https://andoniaf.github.io>



# \$ ls agenda/

- Docker layers
- Layers Size
- Dive
- Layers Squash
- ONBUILD instruction

# Docker layers?



Original photo by [Sebastian Herrmann](#) on [Unsplash](#)

# Docker layers

- Container images consists of several layers.
- Each layer corresponds to certain instructions in your Dockerfile

```
File: VLCTechHub-api/Dockerfile
1 FROM ruby:2.6.5
2
3 RUN set set -ex \
4     && curl -sL https://deb.nodesource.com/setup_10.x | bash - \
5     && apt-get update -qq && apt-get install -qq --no-install-recommends \
6         nodejs \
7     && apt-get clean \
8     && rm -rf /var/lib/apt/lists/*
9
10 RUN gem update --system 3.0.6
11
12 ENV app /app
13 RUN mkdir $app
14 WORKDIR $app
15
16 COPY Gemfile Gemfile.lock $app/
17 RUN bundle install
18
19 COPY . $app/
20
21 ENV RACK_ENV=production
22 ENV PORT=80
23
24 EXPOSE $PORT
25
26 CMD bundle exec puma -C '-' -e $RACK_ENV -p $PORT
```

# Docker layers

- Container images consists of several layers.
- Each layer corresponds to certain instructions in your Dockerfile

```
→ docker history python:3.8-slim
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
ec75d34adff9	2 weeks ago	/bin/sh -c #(nop) CMD ["python3"]	0B	
<missing>	2 weeks ago	/bin/sh -c set -ex; savedAptMark="\$(apt-ma...	8.42MB	
<missing>	2 weeks ago	/bin/sh -c #(nop) ENV PYTHON_GET_PIP_SHA256...	0B	
<missing>	2 weeks ago	/bin/sh -c #(nop) ENV PYTHON_GET_PIP_URL=ht...	0B	
<missing>	2 weeks ago	/bin/sh -c #(nop) ENV PYTHON_PIP_VERSION=20...	0B	
<missing>	7 weeks ago	/bin/sh -c cd /usr/local/bin && ln -s idle3...	32B	
<missing>	7 weeks ago	/bin/sh -c set -ex && savedAptMark="\$(apt-...	28.1MB	
<missing>	7 weeks ago	/bin/sh -c #(nop) ENV PYTHON_VERSION=3.8.5	0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) ENV GPG_KEY=E3FF2839C048B...	0B	
<missing>	7 weeks ago	/bin/sh -c apt-get update && apt-get install...	7.03MB	
<missing>	7 weeks ago	/bin/sh -c #(nop) ENV LANG=C.UTF-8	0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) ENV PATH=/usr/local/bin:/...	0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) CMD ["bash"]	0B	
<missing>	7 weeks ago	/bin/sh -c #(nop) ADD file:3af3091e7d2bb40bc...	69.2MB	



# Docker layers

- Image layers are read-only.
- When you launch a container from an image, it adds a thin writable layer.

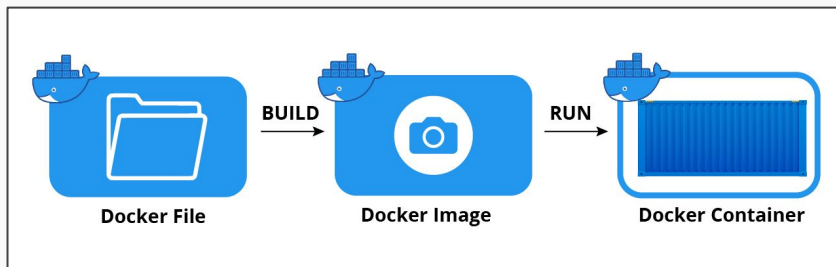


Image from [jfrog.com](http://jfrog.com)

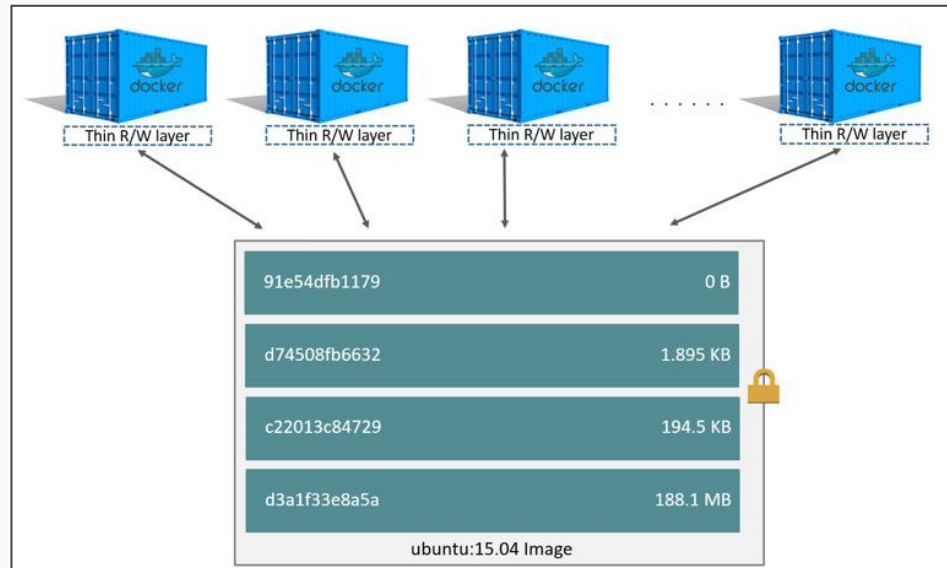


Image from [Docker Docs](https://docs.docker.com)

# Docker layers: Size

```
→ docker history matching_recon
IMAGE          CREATED          CREATED BY          SIZE
b24baa4bbfd7   7 weeks ago     RUN /bin/sh -c cp config/examples/*.yml $APP... 82.9kB
<missing>      7 weeks ago     COPY . /opt/recon # buildkit 4.62MB
<missing>      7 weeks ago     RUN /bin/sh -c bundle install -j 10 --withou... 154MB
<missing>      7 weeks ago     COPY Gemfile Gemfile.lock /opt/recon/ # buil... 10.1kB
<missing>      7 weeks ago     WORKDIR /opt/recon 0B
<missing>      7 weeks ago     RUN /bin/sh -c mkdir -p $APP # buildkit 0B
<missing>      7 weeks ago     WORKDIR /tmp 0B
<missing>      7 weeks ago     ENV LANGUAGE=en_US.UTF-8 0B
<missing>      7 weeks ago     ENV LANG=en_US.UTF-8 0B
<missing>      7 weeks ago     ENV LC_ALL=en_US.UTF-8 0B
<missing>      7 weeks ago     RUN /bin/sh -c sed -i -e 's/# en_US.UTF-8 UT... 3.61MB
<missing>      7 weeks ago     RUN /bin/sh -c apt-get update && apt-get ins... 220MB
<missing>      7 weeks ago     ENV BUILD_PACKAGES=build-essential curl git ... 0B
<missing>      7 weeks ago     ENV PHANTOMJS_PACKAGES=fontconfig libfreetyp... 0B
<missing>      7 weeks ago     ENV APP=/opt/recon 0B
<missing>      7 weeks ago     RUN /bin/sh -c chmod +x /bin/chamber # build... 17.7MB
<missing>      7 weeks ago     ADD https://github.com/segmentio/chamber/rel... 17.7MB
<missing>      18 months ago  /bin/sh -c #(nop) CMD ["irb"] 0B
<missing>      18 months ago  /bin/sh -c mkdir -p "$GEM_HOME" && chmod 777... 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV PATH=/usr/local/bundl... 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV BUNDLE_PATH=/usr/loca... 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV GEM_HOME=/usr/local/b... 0B
<missing>      18 months ago  /bin/sh -c set -ex && savedAptMark="$(apt-... 40.8MB
<missing>      18 months ago  /bin/sh -c #(nop) ENV RUBYGEMS_VERSION=3.0.3 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV RUBY_DOWNLOAD_SHA256=... 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV RUBY_VERSION=2.6.1 0B
<missing>      18 months ago  /bin/sh -c #(nop) ENV RUBY_MAJOR=2.6 0B
<missing>      18 months ago  /bin/sh -c mkdir -p /usr/local/etc && { e... 45B
<missing>      18 months ago  /bin/sh -c apt-get update && apt-get instal... 36.7MB
<missing>      18 months ago  /bin/sh -c #(nop) CMD ["bash"] 0B
<missing>      18 months ago  /bin/sh -c #(nop) ADD file:5ea7dfe8c8bc87ebe... 55.3MB
```

```
→ docker images | grep matching_recon
matching_recon
```

latest

b24baa4bbfd7

7 weeks ago

551MB



# Docker layers: Size

## How to Keep It Small?

- Small base images

python	3.8.3-slim	9d84edf35a0a	4 months ago	165MB
	3.8.3	7f5b6ccd03e9	4 months ago	934MB
	3.8.3-alpine	8ecf5a48c789	4 months ago	78.9MB

- Use .dockerignore files

	File: /tmp/.dockerignore
1	Dockerfile
2	node_modules
3	.git
4	deploy/env.list

176M	./git/s3s3mirror/.git
183M	./git/aseprite/.git
288M	./some/shitty/cms/.git
669M	./classic/monorepo/code/.git

# Docker layers: Size

## How to Keep It Small?

- Multistage builds

emo_getter	latest	14.1MB
golang	1.11.5-alpine3.9	311MB

```
File: Dockerfile
1 FROM golang:1.11.5-alpine3.9 AS build
2
3 RUN apk add --no-cache git
4
5 RUN addgroup -g 1000 -S gocker && \
6 adduser -u 1000 -S gocker -G gocker
7
8 WORKDIR $GOPATH/src/andoniaf/emo_getter/
9 COPY . .
10
11 # Get dependencies
12 RUN go get -d -v
13
14 # Build bin
15 RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o /go/bin/emo_getter
16
17 FROM scratch
18
19 # Copy unprivileged user
20 COPY --from=build /etc/passwd /etc/passwd
21
22 # Copy bin from build
23 COPY --from=build /go/bin/emo_getter /go/bin/emo_getter
24
25 USER gocker
26
27 ENTRYPOINT ["/go/bin/emo_getter"]
```

# Docker layers: Size

## How to Keep It Small?

- RUN statements

If you edit or remove a file, that file still exists inside your docker image.

We must clean temporal or useless files in the same RUN sentence to keep our images small.

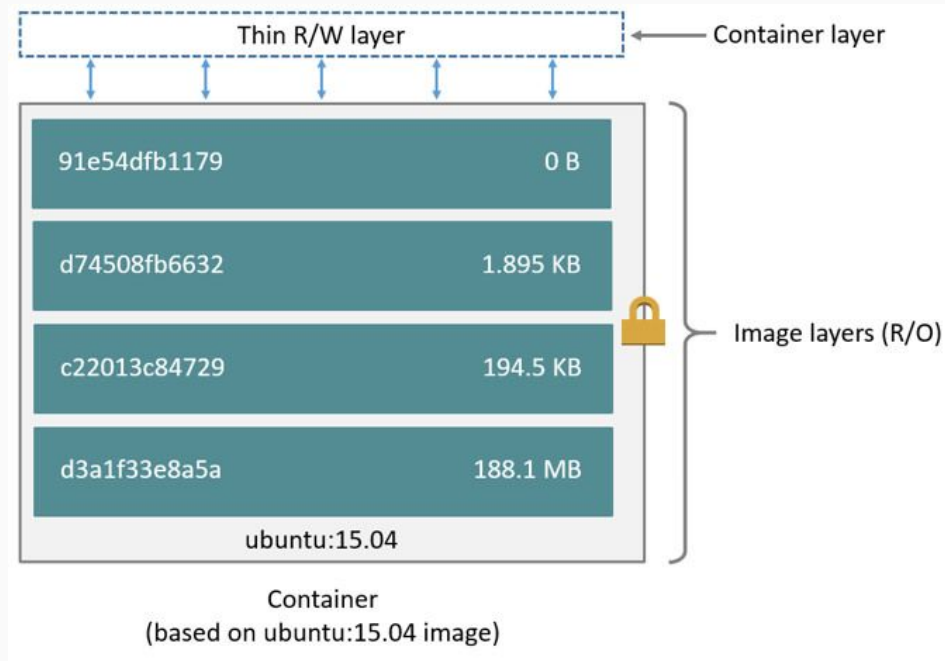


Image from [Docker Docs](#)

```
RUN set set -ex \  
&& curl -sL https://deb.nodesource.com/setup_10.x | bash - \  
&& apt-get update -qq && apt-get install -qq --no-install-recommends \  
    nodejs \  
&& apt-get clean \  
&& rm -rf /var/lib/apt/lists/*
```

# Docker layers: Dive

<https://github.com/wagoodman/dive>

- Show Docker image contents broken down by layer
- Indicate what's changed in each layer
- Estimate "image efficiency"
- CI mode
- Multiple Image Sources and Container Engines Supported:
  - docker
  - docker-archive (tar)
  - podman

README.md

## dive

go report A+ PASSED Donate PayPal

A tool for exploring a docker image, layer contents, and discovering ways to shrink the size of your Docker/OCI image.

The screenshot shows the Dive tool interface in a terminal window. The top section displays a list of layers with columns for Computed Image ID, Size, Command, Permission, UID:GID, and Size. The layers are listed in descending order of size. The bottom section shows the 'Image & Layer Details' for the selected layer, including the command used to create the layer and the file tree structure of the image. The file tree shows the root directory with subdirectories like bin, dev, etc, home, lib, media, cdrom, floppy, usb, mnt, proc, root, example, run, sbin, somefile.txt, srv, sys, tmp, usr, and var. The example directory contains somefile1.txt, somefile2.txt, and somefile3.txt.

```
1: Terminal -
[Layers]
Cmp Image ID      Size  Command
sha256:cd7108a72410606589 4.1 MB FROM sha256:cd7108a72410606589
sha256:f93b1ccbacace8c82e 2.1 kB #(nop) ADD file:63d4894bd0857354b
sha256:d2a26572cd4dc4358 0 B mkdir /root/example
sha256:9f50561518484bb62a 2.1 kB cp /somefile.txt /root/example/so
sha256:edf209d09263ab03c7 2.1 kB cp /somefile.txt /root/example/so
sha256:bb70c7aab83602e8a6 2.1 kB cp /somefile.txt /root/example/so
sha256:b0a712d3d08bc83c48 2.1 kB mv /root/example/somefile3.txt /r
sha256:05e8660b8f90e77b93 0 B rm -rf /root/example/

[Image & Layer Details]
Layer Command
/bin/sh -c cp /somefile.txt /root/example/somefile3.txt

Image efficiency score: 99 %
Potential wasted space: 6.2 kB

Count Total Space Path
2 4.2 kB /root/example
2 2.1 kB /root/example/somefile3.txt

[Aggregated Layer Contents]
Permission UID:GID Size Filetree
drwxr-xr-x 0:0 805 kB -- bin
drwxr-xr-x 0:0 0 B -- dev
drwxr-xr-x 0:0 251 kB -- etc
drwxr-xr-x 0:0 0 B -- home
drwxr-xr-x 0:0 2.7 MB -- lib
drwxr-xr-x 0:0 0 B -- media
drwxr-xr-x 0:0 0 B -- cdrom
drwxr-xr-x 0:0 0 B -- floppy
drwxr-xr-x 0:0 0 B -- usb
drwxr-xr-x 0:0 0 B -- mnt
drwxr-xr-x 0:0 0 B -- proc
drwxr-xr-x 0:0 6.2 kB -- root
drwxr-xr-x 0:0 6.2 kB -- example
-rw-r--r-- 0:0 2.1 kB -- somefile1.txt
-rw-r--r-- 0:0 2.1 kB -- somefile2.txt
-rw-r--r-- 0:0 2.1 kB -- somefile3.txt
drwxr-xr-x 0:0 0 B -- run
drwxr-xr-x 0:0 213 kB -- sbin
-rw-rw-r-- 0:0 2.1 kB -- somefile.txt
drwxr-xr-x 0:0 0 B -- srv
drwxr-xr-x 0:0 0 B -- sys
drwxr-xr-x 0:0 0 B -- tmp
drwxr-xr-x 0:0 176 kB -- usr
drwxr-xr-x 0:0 0 B -- var
```

# Docker layers: Dive



# Docker layers: Squash

```
docker build --squash
```

- Once the image is built, squash the new layers into a new image with a single new layer.
- Squashing does not destroy any existing image, rather it creates a new image with the content of the squashed layers.
- For most use cases, multi-stage builds are a better alternative, as they give more fine-grained control over your build, and can take advantage of future optimizations in the builder.



# Docker layers: Squash

```
→ docker run --rm -it -v /var/run/docker.sock:/var/run/docker.sock wagooodman/dive:latest --ci openvpn:squash_local

Using default CI config
Image Source: docker://openvpn:squash_local
Fetching image... (this can take a while for large images)
Analyzing image...
  efficiency: 100.0000 %
  wastedBytes: 0 bytes (0 B)
  userWastedPercent: NaN %
Inefficient Files:
Count Wasted Space File Path
None
Results:
  PASS: highestUserWastedPercent
  SKIP: highestWastedBytes: rule disabled
  PASS: lowestEfficiency
Result:PASS [Total:3] [Passed:2] [Failed:0] [Warn:0] [Skipped:1]

~ took 5s
→ docker run --rm -it -v /var/run/docker.sock:/var/run/docker.sock wagooodman/dive:latest --ci openvpn:squash_multi

Using default CI config
Image Source: docker://openvpn:squash_multi
Fetching image... (this can take a while for large images)
Analyzing image...
  efficiency: 100.0000 %
  wastedBytes: 0 bytes (0 B)
  userWastedPercent: NaN %
Inefficient Files:
Count Wasted Space File Path
None
Results:
  PASS: highestUserWastedPercent
  SKIP: highestWastedBytes: rule disabled
  PASS: lowestEfficiency
Result:PASS [Total:3] [Passed:2] [Failed:0] [Warn:0] [Skipped:1]
```

# Docker layers: Squash

Layers			Current Layer Contents	
Cmp	Size	Command		
	14 MB	FROM acbabd8622a109b	bin	
Layer Details			dev	
Tags: (unavailable)			etc	
Id: acbabd8622a109b5a971dc3dedab677ac9d0373692c141978b27fc39019bc82b			home	
Digest: sha256:7b2d418b821f9d3caf524a8e6ea3dc1b58491e93aca6b4ebf992e81256f8892d			lib	
Command:			media	
Image Details			mnt	
Total Image size: 14 MB			opt	
Potential wasted space: 0 B			proc	
Image efficiency score: 100 %			root	
Count	Total Space	Path	run	
			sbin	
			srv	
			sys	
			tmp	
			usr	
			var	

# Docker layers: ONBUILD



Questions?

Thoughts?

Thanks!



# \$ more info.lst

- [About storage drivers \[Docker Docs\]](#)
- [More info about docker layers, the copy-on-write \(CoW\) strategy. \[Docker Docs\]](#)
- [scratch docker image](#)
- [wagoodman/dive \[Github\]](#)
- [Squashing image's layers \[Docker Docs\]](#)
- [andoniaf/malicious-onbuild-image-demo \[Github\]](#)