Using Docker for Fun and Profit

Martin Sereinig, 2018

about me

«zuagraster»

i really like ~~computers~~ web development

vp of engineering @ usersnap

twitter: **@srecnig**, linkedin: **Martin Sereinig**

docker

some definitions

~~wtf~~ what is docker, really?

»Docker is a computer program that performs operating-system-level **virtualization**, also known as “**containerization**".«

–https://en.wikipedia.org/wiki/Docker_(software)

»Docker Engine powers millions of applications worldwide, providing a **standardized packaging format** for diverse applications.«

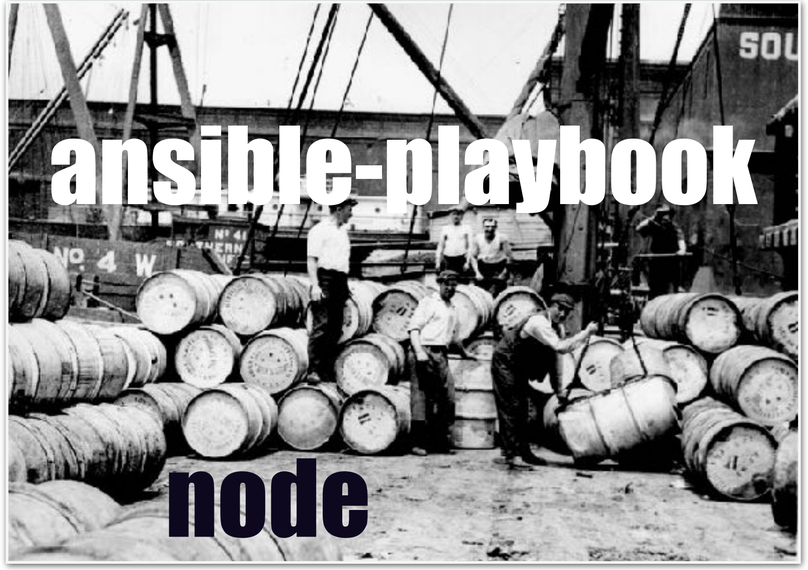
–https://www.docker.com/products/docker-engine

No. 4 W

ROTHENSTEIN

HANJIN
HYUNDAI
COSCO
YANG MING
K LINE
MOL
HAMBURG SÜD
MAERSK

shipping a new version



shipping a new version

docker

why docker?

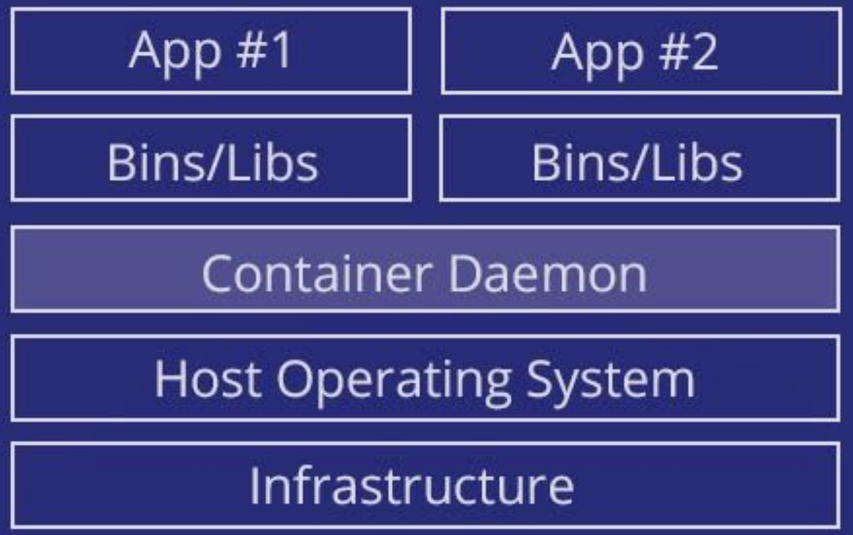
- apps are packaged in a standard way
- versions include everything they need
- containers are immutable (kind of)
- infrastructure as code (versioning!)

DEMO

docker vs. vms

VIRTUAL MACHINES

CONTAINERS

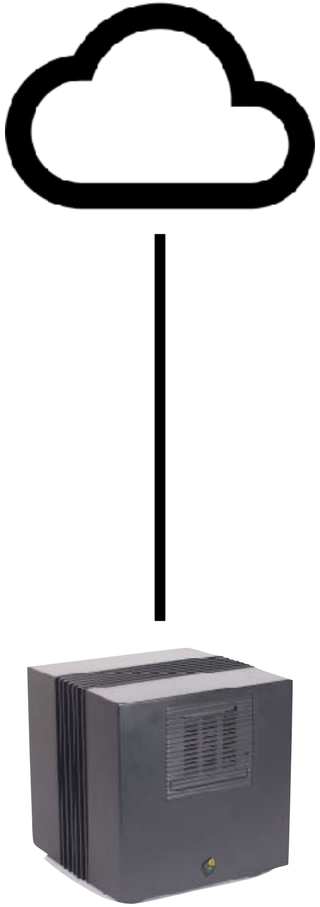


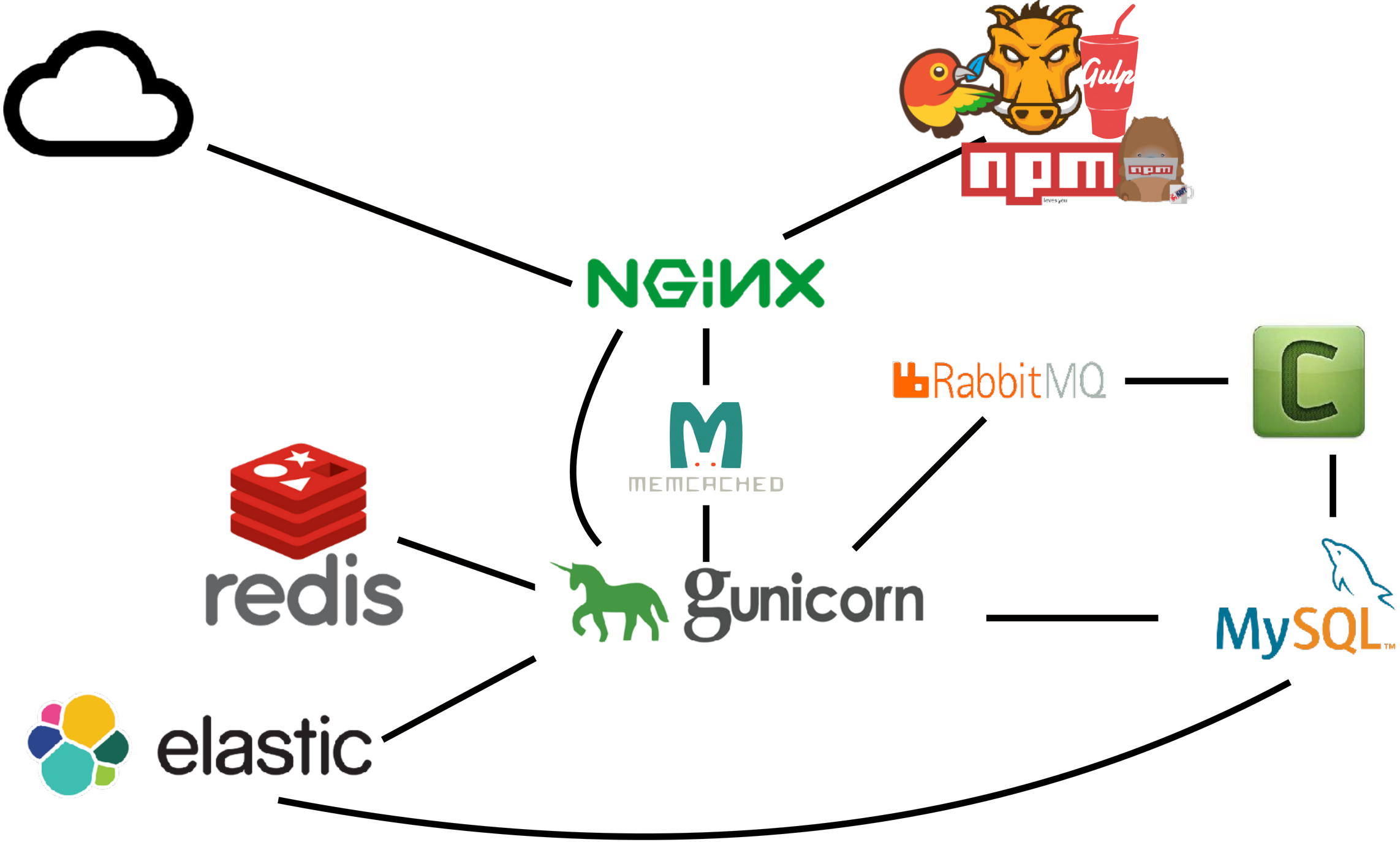
docker vs vms

- ease of use when building images
- really fast when building images
- really fast when starting images
- better performance
- security considerations

docker in development

web apps got really complicated




Gulp
npm
NGINX
RabbitMQ
C
MEMCACHED
redis
gunicorn
MySQL
elastic

problem: how do i replicate this in development?

solution: docker-compose

docker-compose.yml

```yaml
version: '2'

services:
  wordpress:
    image: wordpress:4.8.2
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_PASSWORD: example
    volumes:
        - ../../usersnap:/var/www/html/wp-content/plugins/usersnap

  mysql:
    image: mariadb
    environment:
      MYSQL_ROOT_PASSWORD: example
```

dockerized developer onboarding

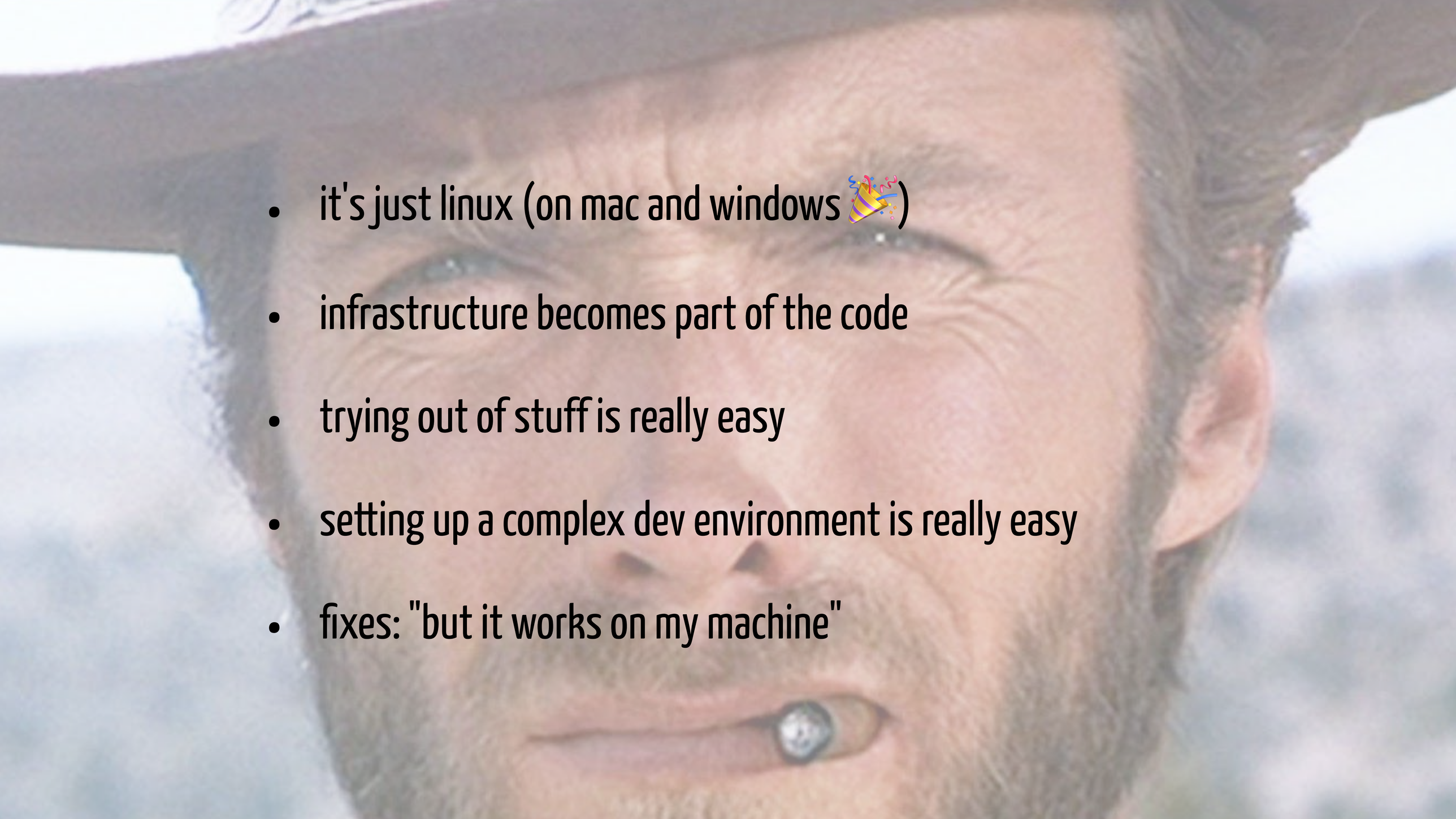
- install docker, docker-compose and git
- git pull
- docker-compose up
-
- start working

but wait, there’s more!

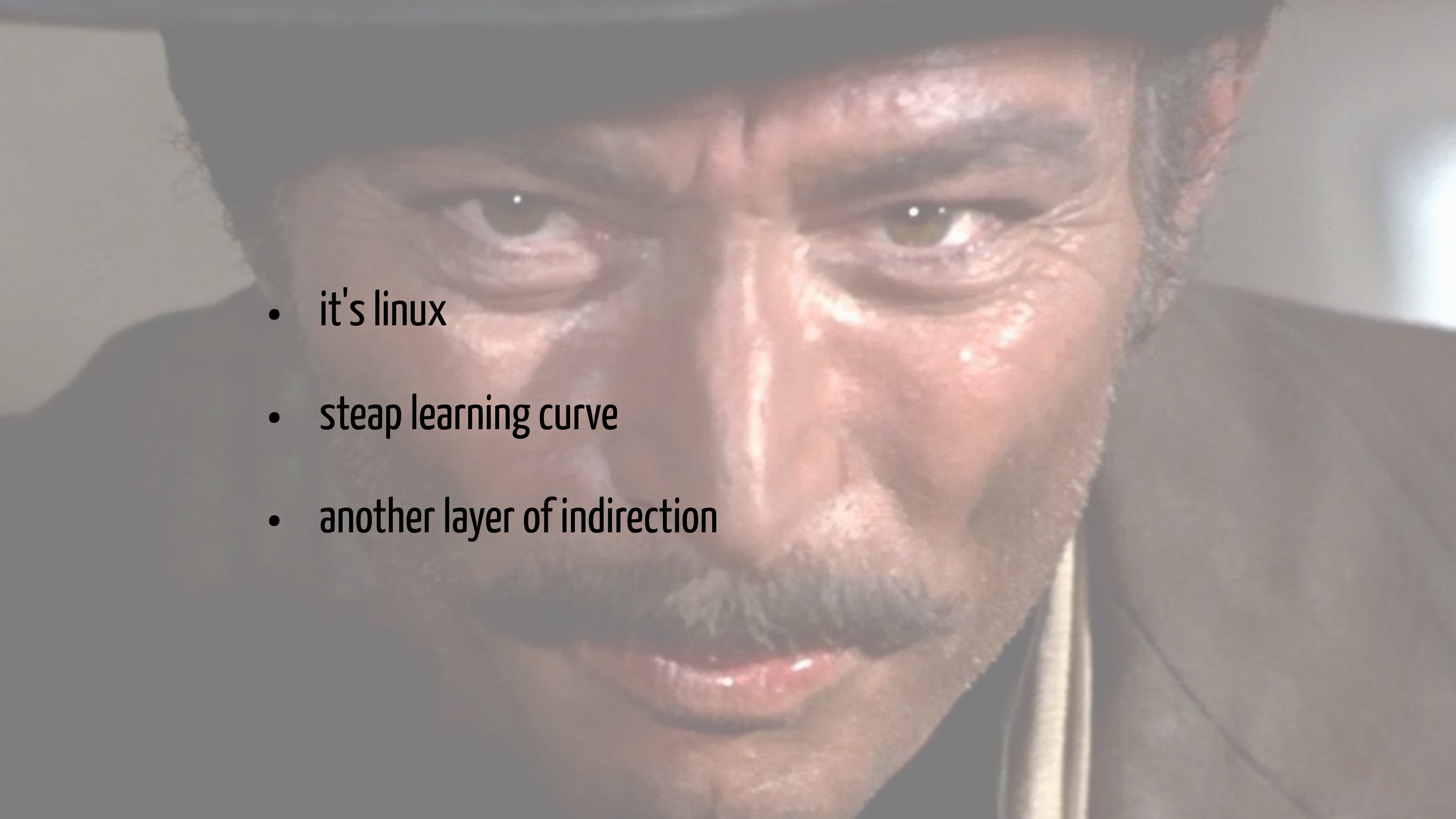
but wait, there's more!

- docker containers for testing
- continous integration based on docker
- shipping apps with docker

tying this up: the good, the bad and the ugly.



- it's just linux (on mac and windows 🎉)
- infrastructure becomes part of the code
- trying out of stuff is really easy
- setting up a complex dev environment is really easy
- fixes: "but it works on my machine"



- it's linux
- steap learning curve
- another layer of indirection

- mac support is meh

fun and profit?

thanks! questions?