DOCKER PARA WEB DEVS

CODETRY

GDG Neuquén

JUAN PABLO ORLANDO

- Estudiante UNCO
- Full-Stack developer @ codetry.io
- jp@codetry.io
- @juampiorlando94

MARTIN MOREIRA

- Estudiante UNCO
- Full-Stack developer @ codetry.io
- m@codetry.io
- @morexlt

1.

"Qué bueno, hay que deployar"

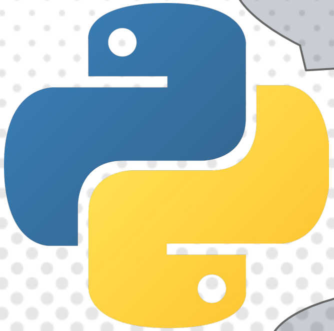
Dijo nunca nadie

TECNOLOGÍAS

Tecnologías

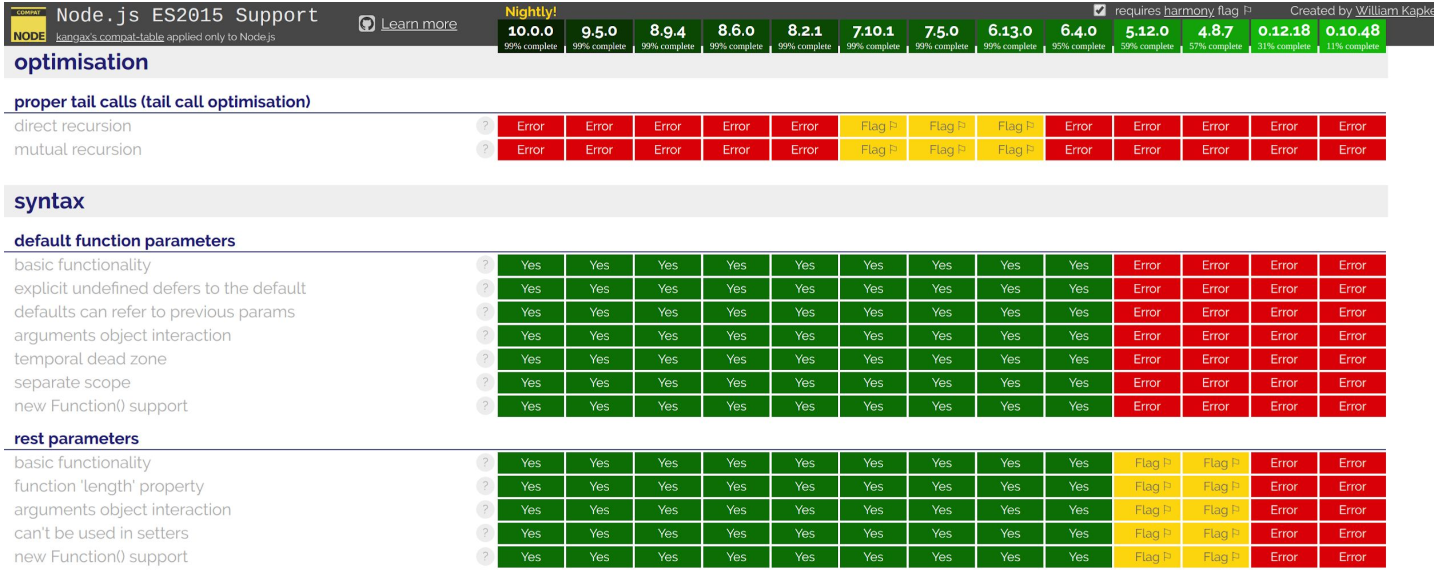
Versiones?

TECNOLOGÍAS



Versiones?
TM

ANALICEMOS UN EJEMPLO...



COMPAT NODE Node.js ES2015 Support

kangax's compat-table applied only to Node.js

Learn more

☑ requires harmony flag ⚐

Created by William Kapke

Nightly!

optimisation

proper tail calls (tail call optimisation)

	10.0.0 99% complete	9.5.0 99% complete	8.9.4 99% complete	8.6.0 99% complete	8.2.1 99% complete	7.10.1 99% complete	7.5.0 99% complete	6.13.0 99% complete	6.4.0 95% complete	5.12.0 59% complete	4.8.7 57% complete	0.12.18 31% complete	0.10.48 11% complete
direct recursion ?	Error	Error	Error	Error	Error	Flag ⚐	Flag ⚐	Flag ⚐	Error	Error	Error	Error	Error
mutual recursion ?	Error	Error	Error	Error	Error	Flag ⚐	Flag ⚐	Flag ⚐	Error	Error	Error	Error	Error

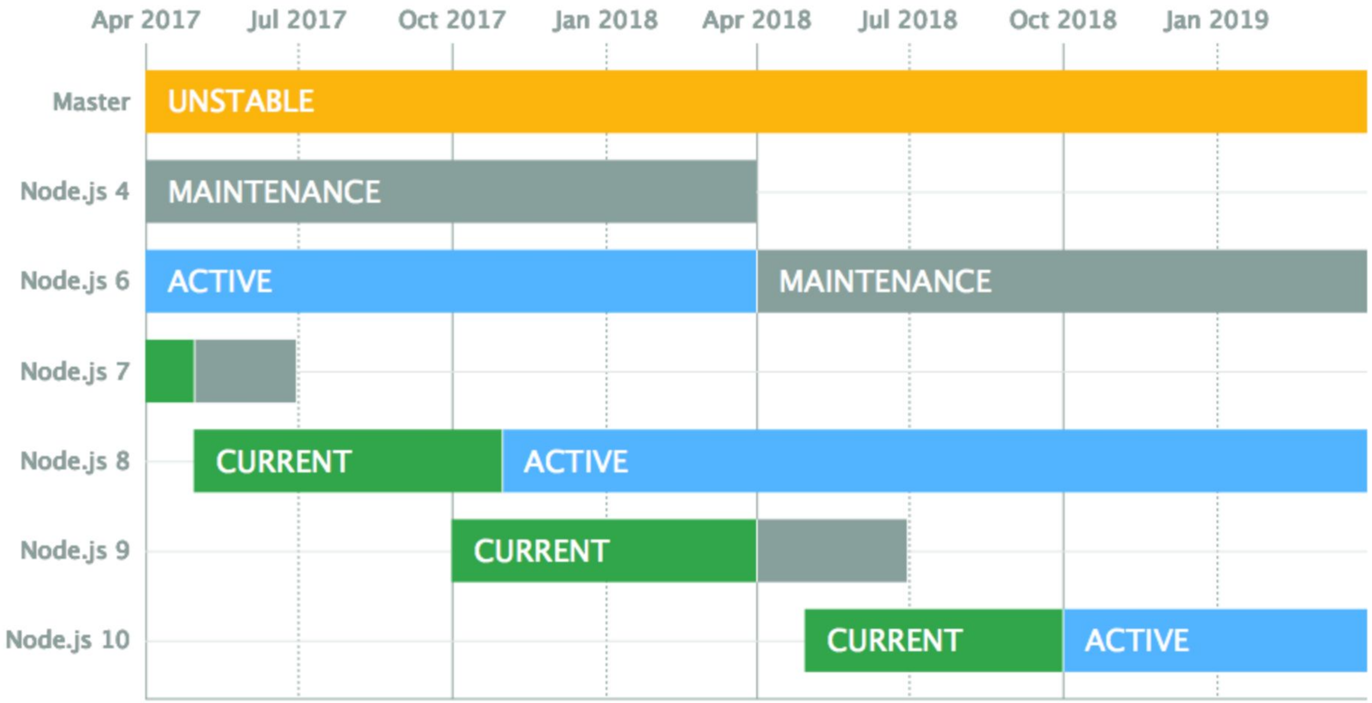
syntax

default function parameters

	10.0.0	9.5.0	8.9.4	8.6.0	8.2.1	7.10.1	7.5.0	6.13.0	6.4.0	5.12.0	4.8.7	0.12.18	0.10.48
basic functionality ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
explicit undefined defers to the default ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
defaults can refer to previous params ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
arguments object interaction ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
temporal dead zone ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
separate scope ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error
new Function() support ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Error	Error	Error	Error

rest parameters

	10.0.0	9.5.0	8.9.4	8.6.0	8.2.1	7.10.1	7.5.0	6.13.0	6.4.0	5.12.0	4.8.7	0.12.18	0.10.48
basic functionality ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Flag ⚐	Flag ⚐	Error	Error
function 'length' property ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Flag ⚐	Flag ⚐	Error	Error
arguments object interaction ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Flag ⚐	Flag ⚐	Error	Error
can't be used in setters ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Flag ⚐	Flag ⚐	Error	Error
new Function() support ?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Flag ⚐	Flag ⚐	Error	Error


Apr 2017
Jul 2017
Oct 2017
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Jan 2019
Master
UNSTABLE
Node.js 4
MAINTENANCE
Node.js 6
ACTIVE
MAINTENANCE
Node.js 7
Node.js 8
CURRENT
ACTIVE
Node.js 9
CURRENT
Node.js 10
CURRENT
ACTIVE

Y TENEMOS QUE ...

1. Demasiadas versiones
2. Diferentes soportes por versión
3. Diferentes features
4. Posibles problemas por != al deployar
5. ...

AMBIENTE DE DESARROLLO

Y TENEMOS QUE ...

1. Diferentes SO’s entre desarrolladores
2. Diferentes versiones de cada SO
3. Diferentes soluciones (ej: timezone)
4. Diferencias con SO del servidor
5. ...

LLEGÓ DOCKER

docker

" INDEPENDENCIA VERDADERA ENTRE LAS APPS, LA INFRAESTRUCTURA Y LOS DESARROLLADORES. DESBLOQUEA SU POTENCIAL, Y CREA MODELOS PARA MEJOR COLABORACIÓN E INNOVACIÓN.

AGILITY

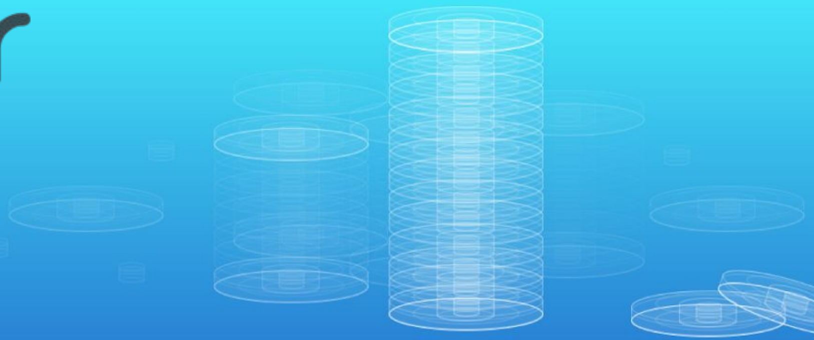
Accelerate software development and deployment by 13X and respond instantly to customer needs.

PORTABILITY

Eliminate the “works on my machine” once and for all. Gain independence across on-prem and cloud environments.

docker

SECURITY



COST SAVINGS

Shut up and take my money!

2.

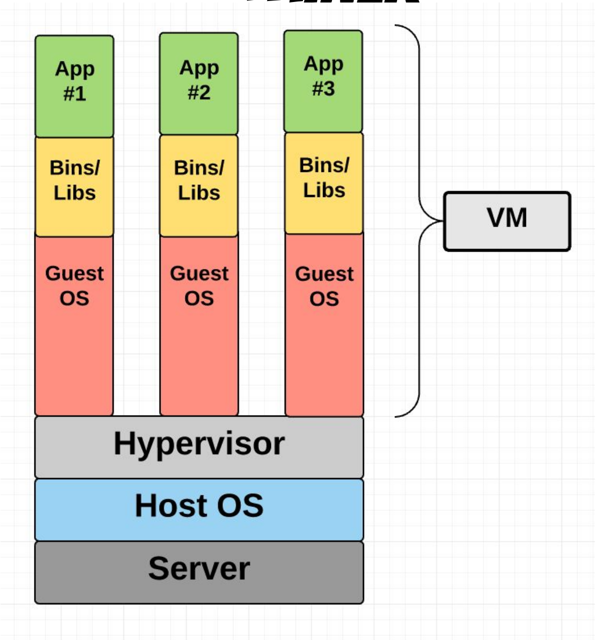
Pero ¿qué es Docker?

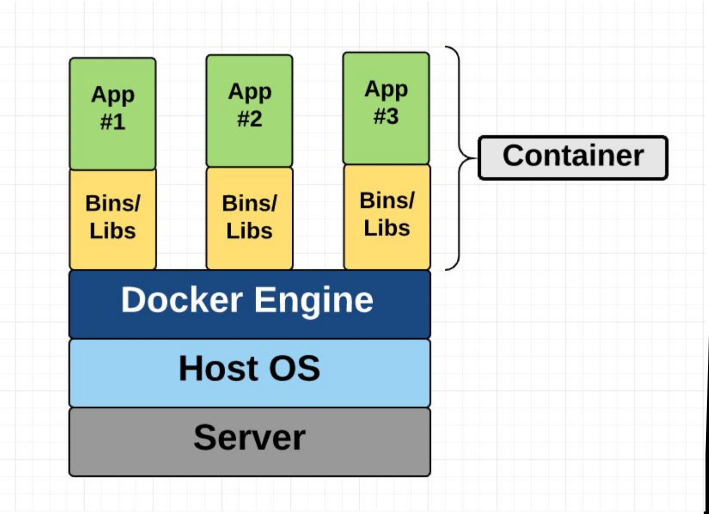
DOCKER

Docker es un servicio de gestión de contenedores.

"Develop, ship and run anywhere".

ARQ VM - CONTAINER

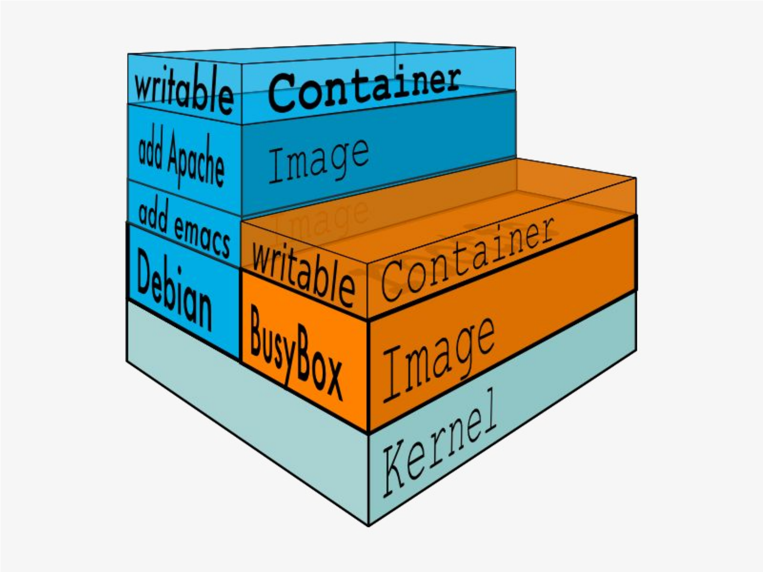




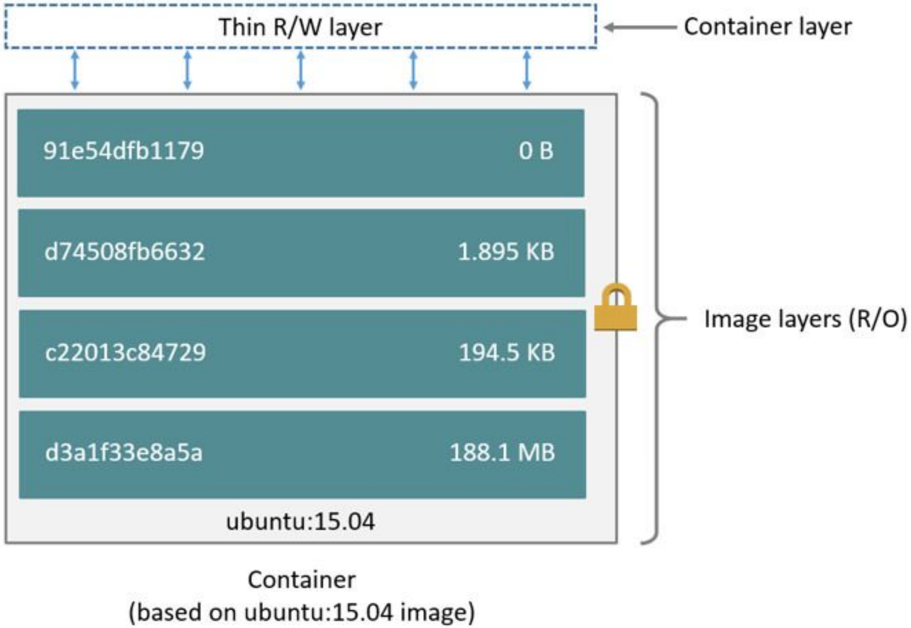
Docker images && containers

Una **imagen** es una combinación de un filesystem y parámetros que ejecuta el Kernel

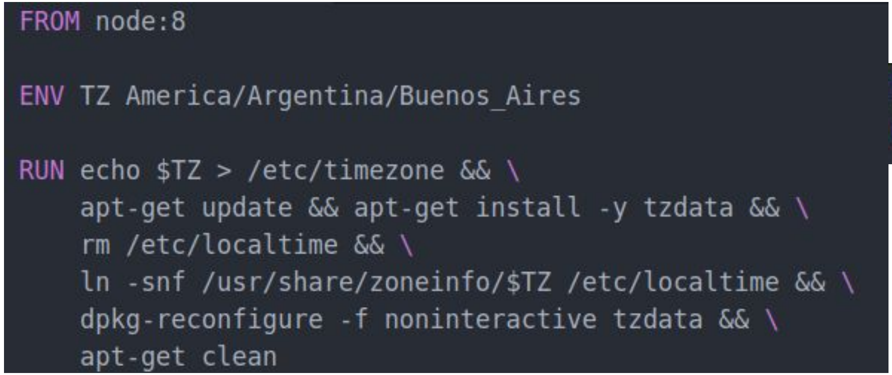
La puesta en ejecución de una de esas imágenes, lo transforma en un contenedor.



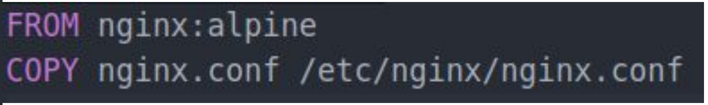
```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```



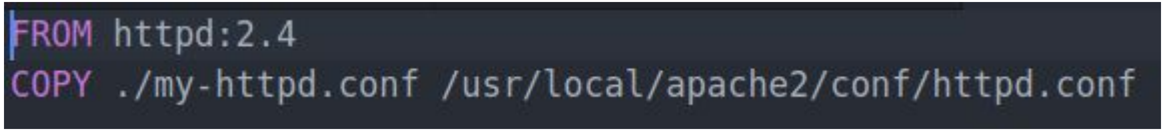
Ejemplo imágenes y contenedores



```
FROM node:8

ENV TZ America/Argentina/Buenos_Aires

RUN echo $TZ > /etc/timezone && \
    apt-get update && apt-get install -y tzdata && \
    rm /etc/localtime && \
    ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && \
    dpkg-reconfigure -f noninteractive tzdata && \
    apt-get clean
```

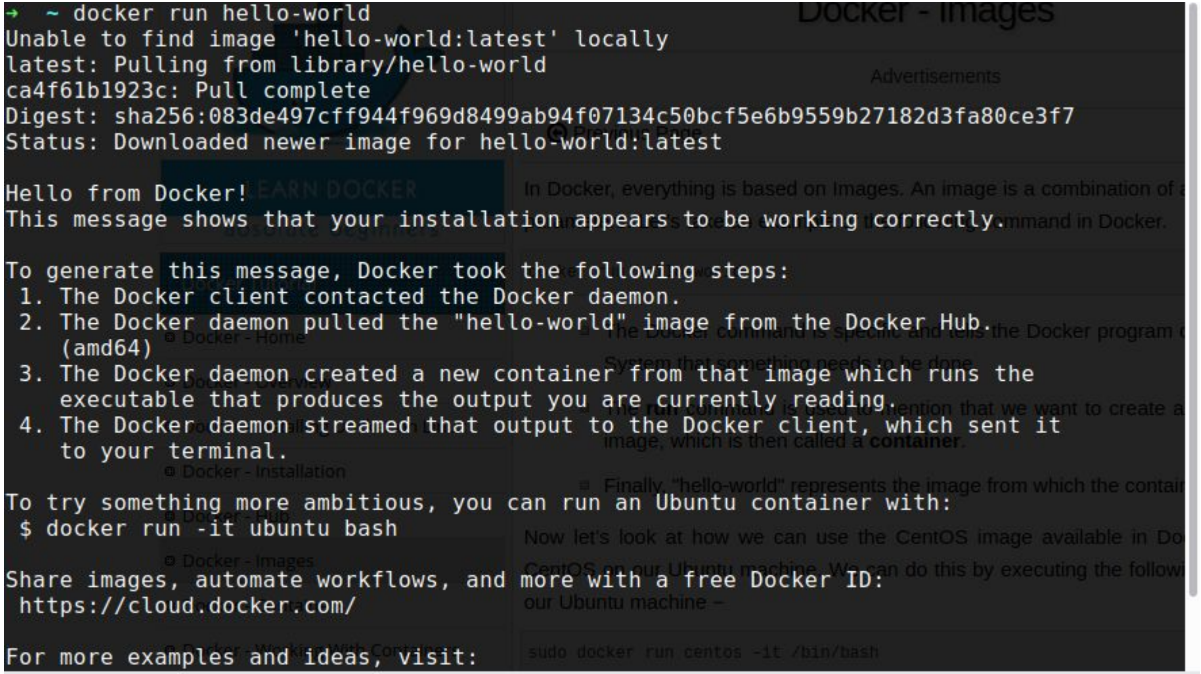


```
FROM nginx:alpine
COPY nginx.conf /etc/nginx/nginx.conf
```



```
FROM httpd:2.4
COPY ./my-httpd.conf /usr/local/apache2/conf/httpd.conf
```

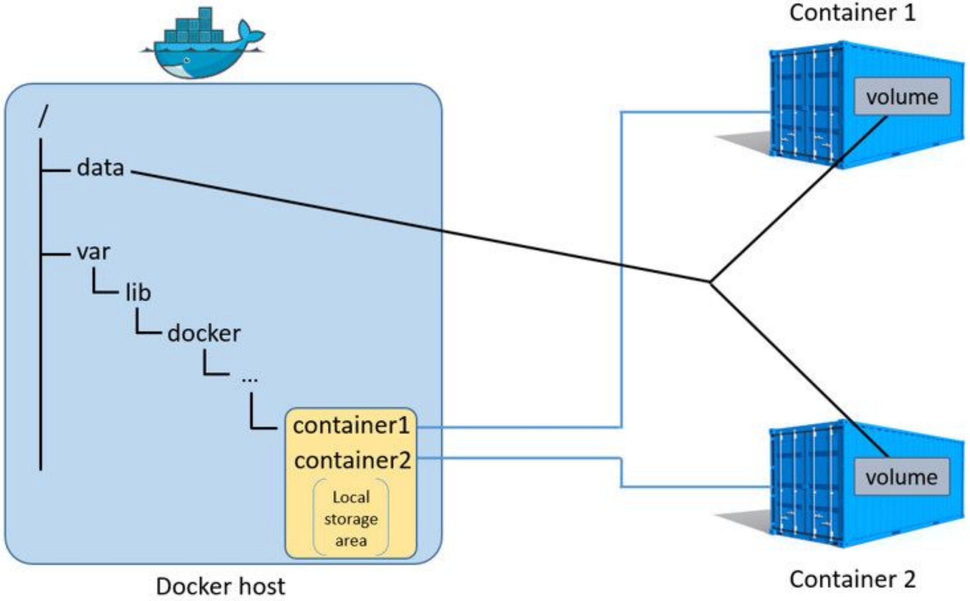
Dockerfiles



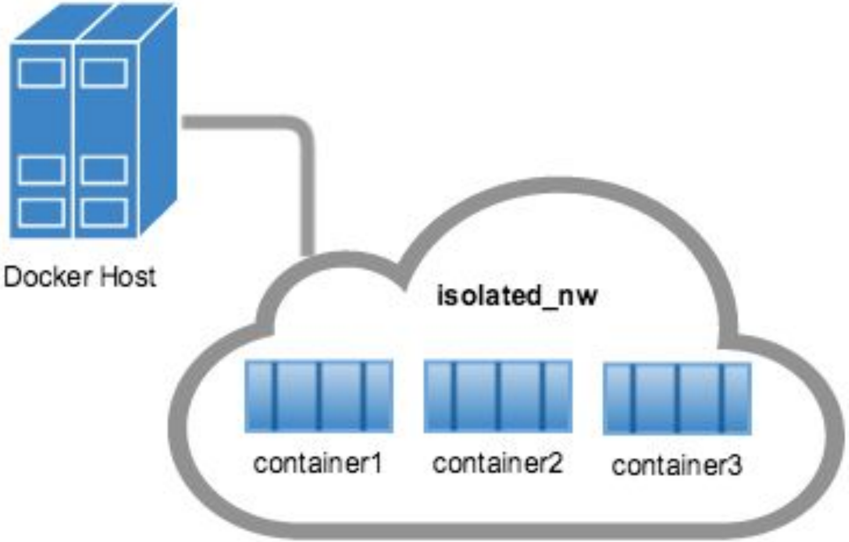
```
➜  ~ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:083de497cff944f969d8499ab94f07134c50bcf5e6b9559b27182d3fa80ce3f7
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://cloud.docker.com/

For more examples and ideas, visit:
```

Ejemplo de ejecución hello-world DEMO!!!

VOLUMENES


Container 1
volume
/
data
var
lib
docker
...
container1
container2
Local storage area
Docker host
volume
Container 2

DOCKER NETWORK



3.

¿CÓMO LO USAMOS PARA EL DESARROLLO?

Todos los desarrolladores, ahora

Docker run

```
docker run –rm \
   –p 443:443 –p 80:80 –name letsencrypt \
   –v "/etc/letsencrypt:/etc/letsencrypt" \
   –v "/var/lib/letsencrypt:/var/lib/letsencrypt" \
   quay.io/letsencrypt/letsencrypt:latest \
   certonly –n –m $LETSENCRYPT_EMAIL –d $DNSNAME –standalone –agree–tos
```

DOCKER RUN

```
docker run –rm \
  –p 443:443 –p 80:80 –name letsencrypt \
  –v "/etc/letsencrypt:/etc/letsencrypt" \
  –v "/var/lib/letsencrypt:/var/lib/letsencrypt" \
  quay.io/letsencrypt/letsencrypt:latest \
  certonly –n –m $LETSENCRYPT_EMAIL –d $DNSNAME –standalone –agree–tos
```

EASY (?

Esto se lo dejamos a los Sysadmins

DOCKER COMPOSE

"UNA HERRAMIENTA PARA DEFINIR Y EJECUTAR APLICACIONES COMPLEJAS CON DOCKER. CON COMPOSE, PUEDES DEFINIR UNA APLICACIÓN MULTI-CONTENEDOR EN UN SOLO ARCHIVO Y EJECUTARLA CON UN SIMPLE COMANDO"

DOCKER-COMPOSE WORKFLOW

1. Definir cada servicio en un Dockerfile (si es necesario)
2. Definir los servicios y su relación con el resto en un docker-compose.yml
3. Ejecutar **docker-compose up** para iniciar el sistema

Comandos:

docker-compose build

docker-compose up

docker-compose logs

376.663

Imagenes publicas (2016)

+10.000.000

Nginx pulls (2018)

6 BILLONES

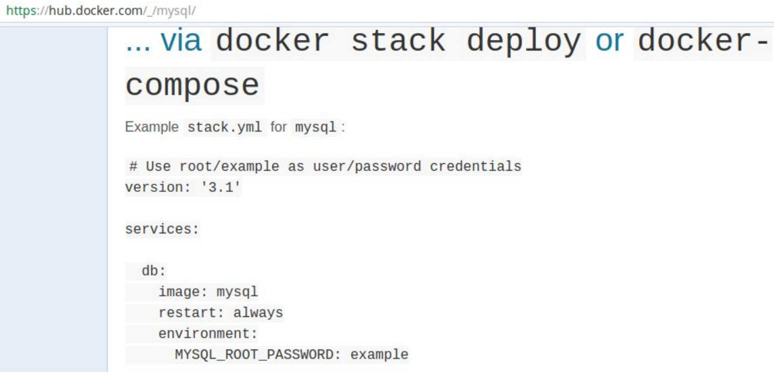
Total Pulls (2016)

387

Ubuntu pulls x min (2016)

DOCKER-COMPOSE Y DOCKER HUB (EJEMPLO)

https://hub.docker.com/_/mysql/

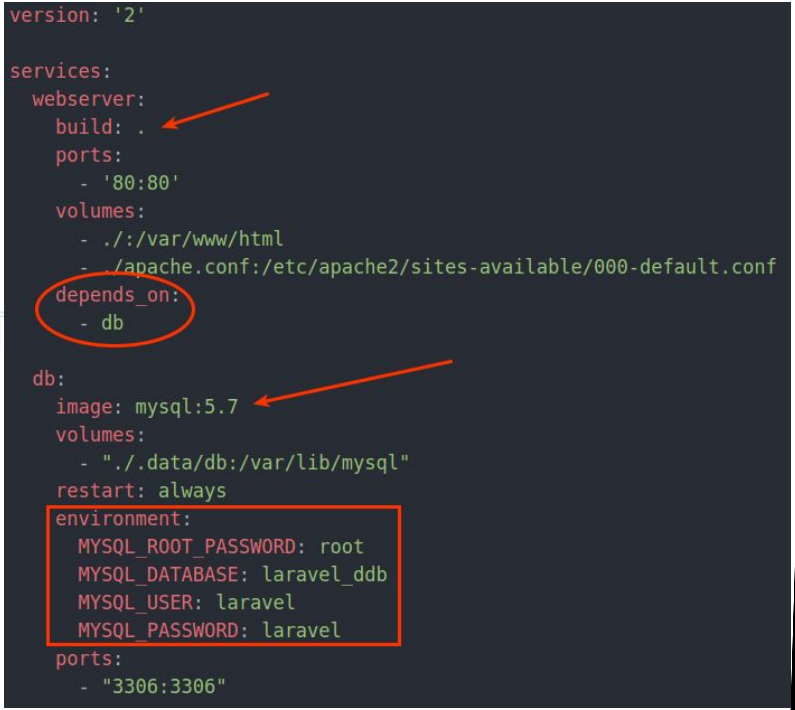


https://hub.docker.com/_/mysql/

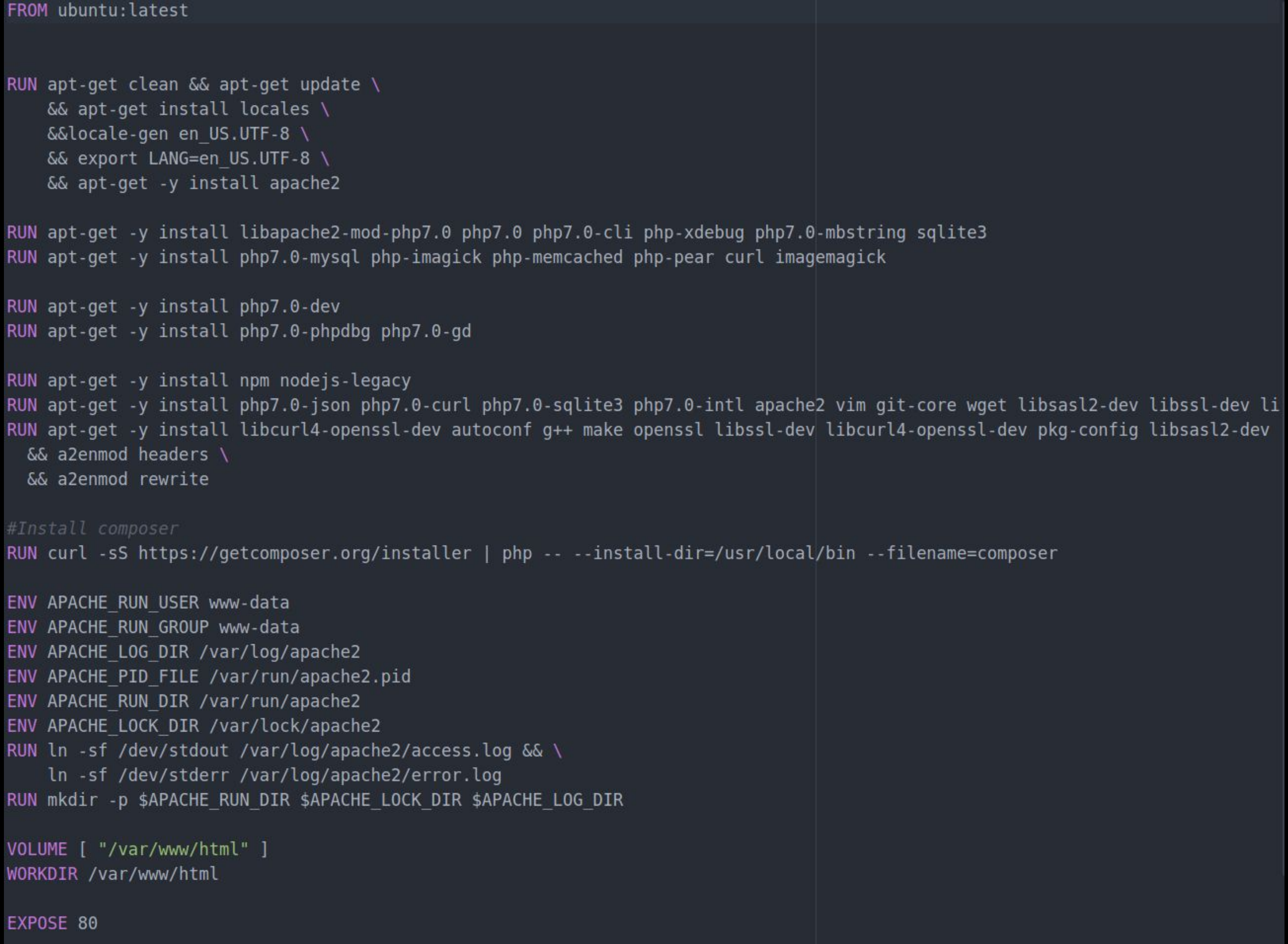
... via `docker stack deploy` or `docker-compose`

Example `stack.yml` for `mysql`:

```
# Use root/example as user/password credentials
version: '3.1'

services:

  db:
    image: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: example
```



```
version: '2'

services:
  webserver:
    build: .
    ports:
      - '80:80'
    volumes:
      - ./:/var/www/html
      - ./apache.conf:/etc/apache2/sites-available/000-default.conf
    depends_on:
      - db

  db:
    image: mysql:5.7
    volumes:
      - "./.data/db:/var/lib/mysql"
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: laravel_ddb
      MYSQL_USER: laravel
      MYSQL_PASSWORD: laravel
    ports:
      - "3306:3306"
```



```
FROM ubuntu:latest


RUN apt-get clean && apt-get update \
    && apt-get install locales \
    &&locale-gen en_US.UTF-8 \
    && export LANG=en_US.UTF-8 \
    && apt-get -y install apache2

RUN apt-get -y install libapache2-mod-php7.0 php7.0 php7.0-cli php-xdebug php7.0-mbstring sqlite3
RUN apt-get -y install php7.0-mysql php-imagick php-memcached php-pear curl imagemagick

RUN apt-get -y install php7.0-dev
RUN apt-get -y install php7.0-phpdbg php7.0-gd

RUN apt-get -y install npm nodejs-legacy
RUN apt-get -y install php7.0-json php7.0-curl php7.0-sqlite3 php7.0-intl apache2 vim git-core wget libsasl2-dev libssl-dev li
RUN apt-get -y install libcurl4-openssl-dev autoconf g++ make openssl libssl-dev libcurl4-openssl-dev pkg-config libsasl2-dev
  && a2enmod headers \
  && a2enmod rewrite

#Install composer
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
RUN ln -sf /dev/stdout /var/log/apache2/access.log && \
    ln -sf /dev/stderr /var/log/apache2/error.log
RUN mkdir -p $APACHE_RUN_DIR $APACHE_LOCK_DIR $APACHE_LOG_DIR

VOLUME [ "/var/www/html" ]
WORKDIR /var/www/html

EXPOSE 80
```

MODIFICAR EL ARCHIVO COMPOSE PARA PRODUCTION

- Remover los volumenes que montan el codigo de la aplicacion, entonces el código se encuentra dentro del container y no puede ser cambiado desde afuera
- Bindear diferentes puertos en la PC host
- Cambiar/Usar variables de entorno diferentes, ej. Para mostrar menos datos en los logs, o que la aplicación mande emails, etc.
- Especificar en el docker-compose.yml el restart: always

GRACIAS

GDG Neuquén

CODETRY