

# Docker From Zero To Hero

Noviembre 2017





# Algunas cosas antes de empezar

- <https://github.com/mercadolibre/workshop-docker-fullstack2017>
- Tener la vm instalada
- Instale docker antes de venir, lo puedo usar ?
  - Sí (aunque la vm tiene todo descargado)
- **Preguntar, Preguntar, Preguntar (en cualquier momento)**



# Agenda

- Que es Docker ?
  - Hola Mundo con Docker
- Docker for development
  - Comandos Básicos
  - Imágenes
  - Proyecto de desarrollo completo
  - Docker Compose
- Docker for production
  - Deployando la app en producción

**Hola !**



**Sebastian Schepens**  
**@sebas\_schep**



**Gabriel Eisbruch**  
**@geisbruch**

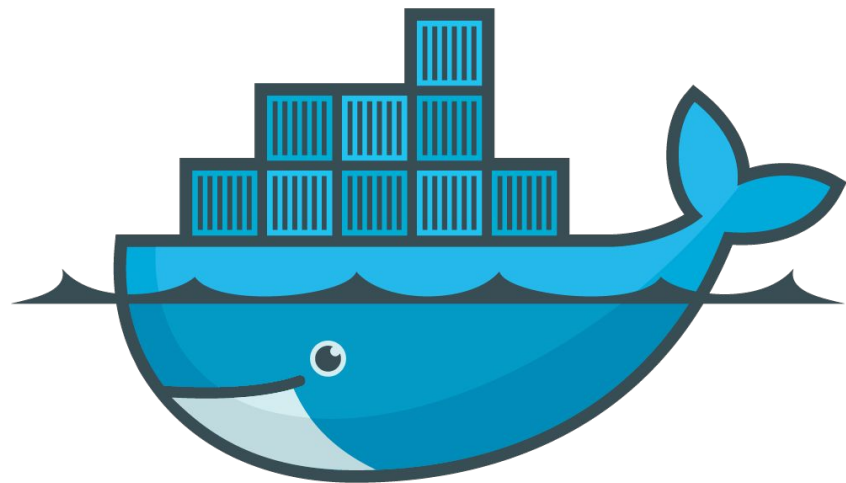


## Qué es Mercadolibre en cifras ?

- +4000 Búsquedas por segundo
- +100 Millones de usuarios
- ~ 140 millones de pagos
- ~ 90 millones de envíos
- +100 millones de items listados
- +5000 personas
- +1500 Ingenieros

## Qué es Mercadolibre en cifras de INFRA ?

- +36000 VM's
- +30 Millones de request por minuto
- +1000 deploys por dia
- 17000 Mensajes por segundo en colas
- 20 Tb de logs y 17000 millones de logs nuevos por dia



docker



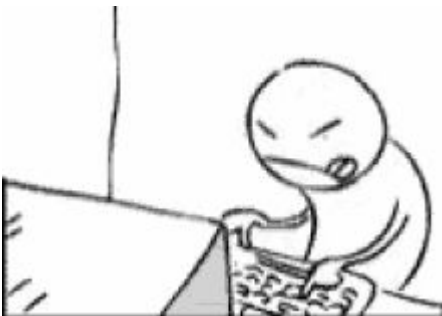


A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

Cómo lo hacemos sin containers?

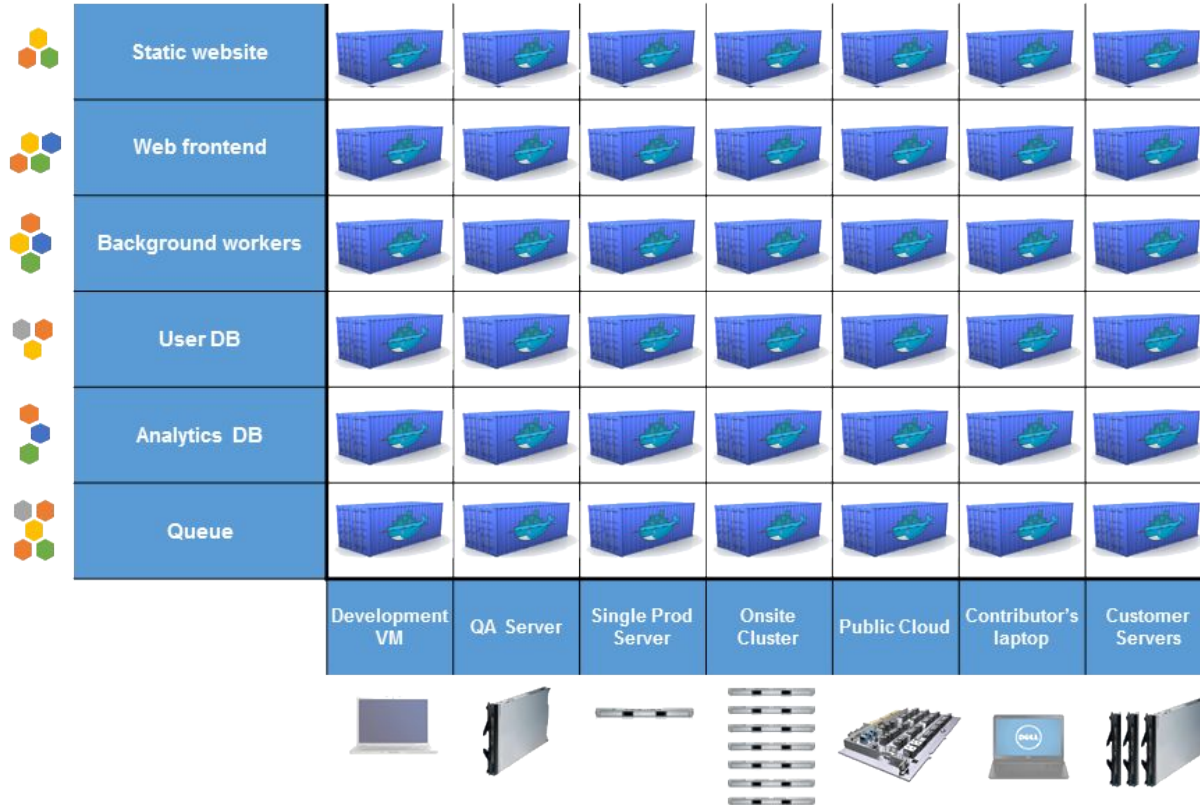


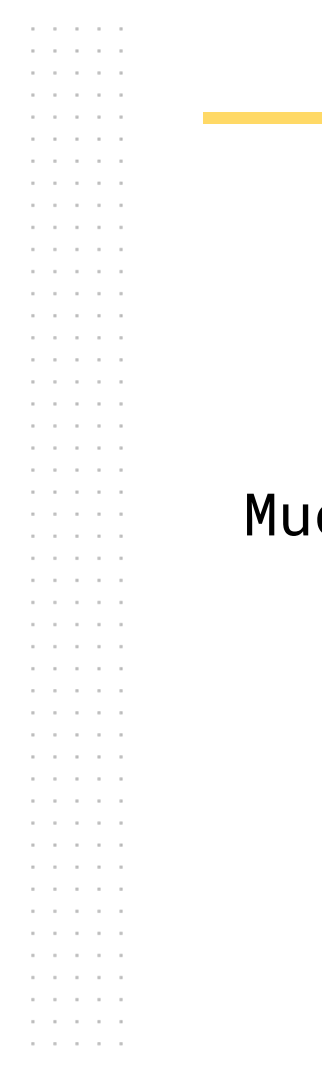
ANSIBLE



```
1 #!/bin/bash
2 #SAMPLE_NAME=11741
3 cd /Volumes/PhilDrive_DMS/TestDec7/sm_postprocess/
4
5 #getho.txt
6
7
8 echo "Debug level set for $DEBUG_LEVEL"
9 echo "log found in scripts directory"
10
11
12 cp $SCRIPT_DIR_OUT ./
13 cp $LOG_DIR_OUT ./
14 cp $SCRIPT_DIR_OUT ./
15 # echo $(cat $SCRIPT_DIR/run_somatic_mutation_analysis $1) no_false_mpf
16 if [ $DEBUG_LEVEL -gt 0 ]
17 then
18 echo "INFO: $SCRIPT_DIR/run_somatic_mutation_analysis.sh $SAMPLE no_false_mpf
19 basename $(ls $LOG_DIR_OUT) basename $(ls $SCRIPT_DIR_OUT)
20 $LOG_FILE $(ls $LOG_FILE)">$(LOG)
21
22 fi
23 $SCRIPT_DIR/run_somatic_mutation_analysis.sh
24
25 echo "End of somatic mutation analysis">> $LOG
```

# Que resuelven los containers?



A decorative vertical dotted pattern on the left side of the slide and a short orange horizontal bar near the top left.

Mucha teoria, me dijiste que esto iba a ser  
práctico

A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

Vamos con un ejemplo?

A decorative vertical dotted pattern on the left side of the slide and a short orange horizontal bar at the top left.

```
docker container run debian echo 'Hola Mundo!!'
```

# Que pasó acá ?


`docker container run debian echo 'Hola Mundo!!'`

Comando

Sub-Comando

Imagen

Comando a correr dentro de la imagen

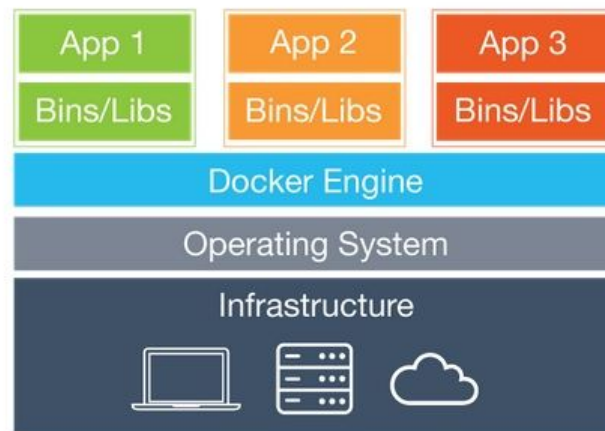
A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

Que pasó acá (por atras) ?





**Virtual Machines**



**Containers**

A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.


Vamos con algo más divertido



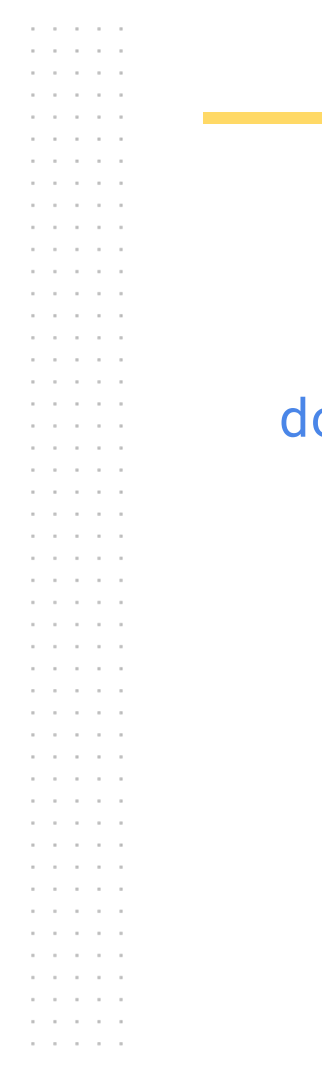
```
docker container run -ti debian bash
```




Terminal  
Interactiva




```
docker container run -ti \  
mercadolibre/nginx-workshop bash
```



```
docker container run -ti -p 9090:80 \
  --name main \
  --rm \
  mercadolibre/nginx-workshop bash
```



```
docker container run -ti -p 9090:80 \
  --name main \
  --rm \
  -v /home/workshop/html:/myhtmls
mercadolibre/nginx-workshop bash
```



```
docker container run -ti -p 9090:80 \
  --name main \
  --rm \
  --mount "type=bind,\
    source=/home/workshop/html,\
    target=/myhtmls" \
  mercadolibre/nginx-workshop bash
```

A decorative element on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

```
docker volume create test
```



A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

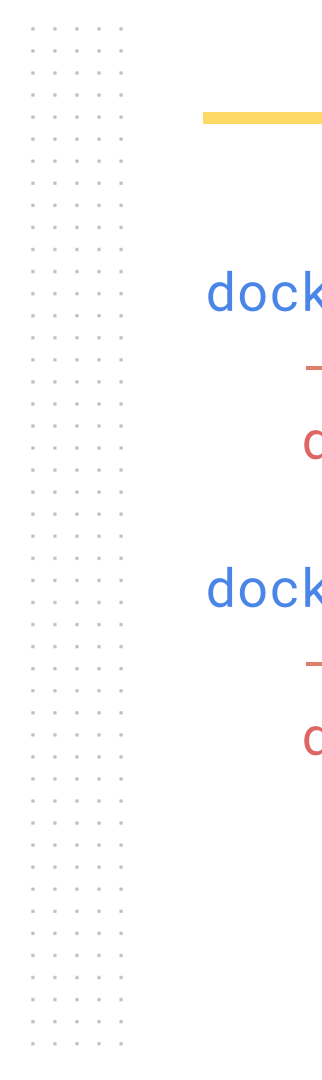
```
docker volume ls
```

```
docker volume inspect test
```

localhost:workshop geisbruch\$ docker volume inspect test1



```
[
  {
    "CreatedAt": "2017-11-25T14:20:21Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/test1/_data",
    "Name": "test1",
    "Options": {},
    "Scope": "local"
  }
]
```

Click to add speaker notes



```
docker container run \  
  --mount "source=test,target=/myhtmls" \  
  debian bash
```

```
docker container run \  
  --mount "source=test,target=/myhtmls,readonly" \  
  debian bash
```



```
docker container run -ti -p 9090:80 \
  --name main \
  --v test \
  debian bash
```

```
docker container run -ti -p 9090:80 \
  --volumes-from main \
  debian bash
```



**Algunos comandos útiles antes de seguir ...**



```
docker container <subcommand> --help
```



docker container ps

A decorative graphic in the top-left corner consisting of a vertical grid of small dots and a short horizontal orange line.

docker container inspect





docker container start/stop/kill



docker container attach

A decorative graphic on the left side of the slide. It consists of a vertical grid of small dots on the far left, and a short, thick horizontal orange bar positioned near the top of the grid.

docker container exec

A decorative graphic on the left side of the slide. It consists of a vertical grid of small dots on the far left, and a short horizontal orange line positioned above the main text.

docker container logs

A decorative graphic on the left side of the slide. It consists of a vertical grid of small dots on the far left, and a short horizontal orange line positioned to the right of the dots, roughly in the upper third of the slide's height.

docker container cp

# Docker Images



```
docker container run -ti \  
mercadolibre/nginx-workshop bash
```



```
docker container run -ti \  
    --name my_container  
mercadolibre/nginx-workshop bash
```

```
vim /etc/nginx/conf.d/default.conf
```

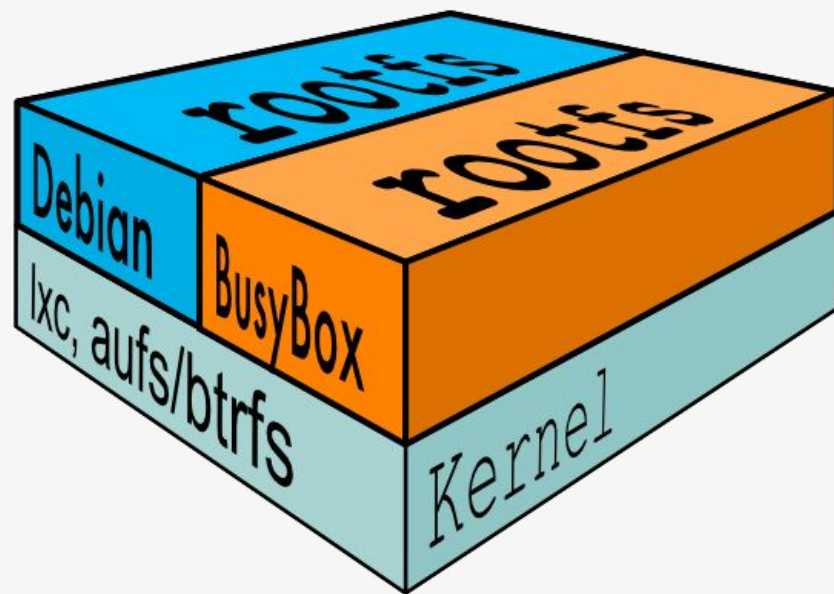


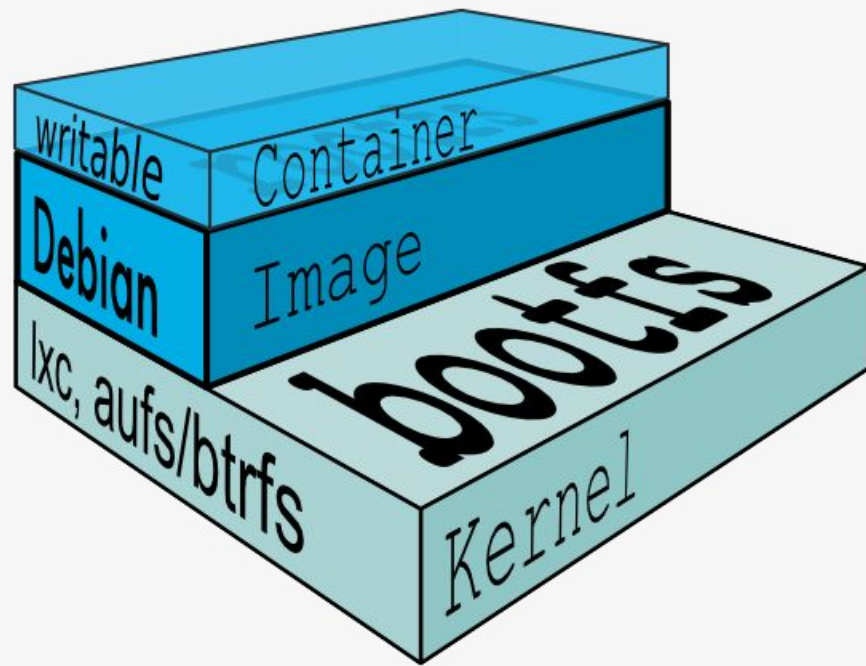


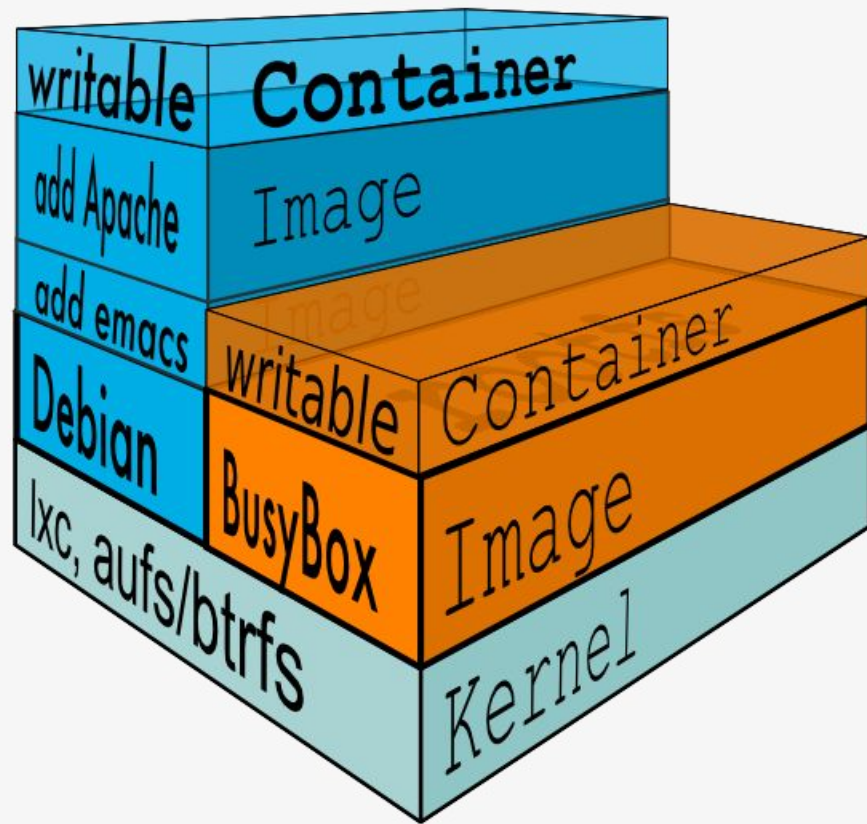
```
docker container diff my_container
```

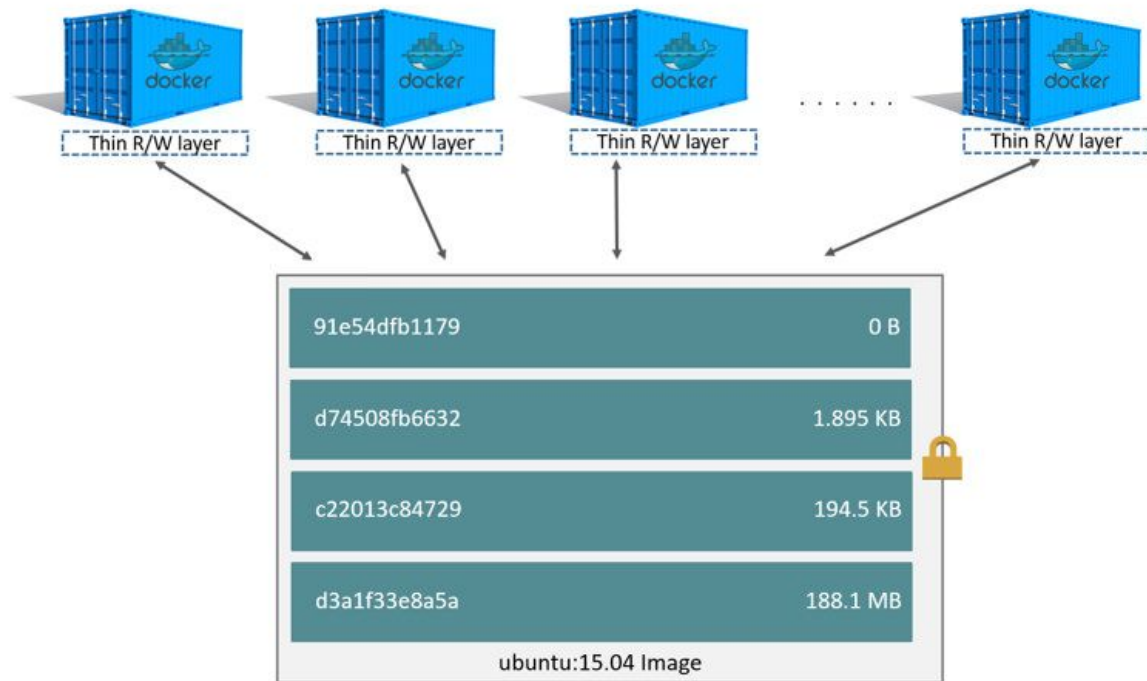
```
docker container commit my_container first_image
```

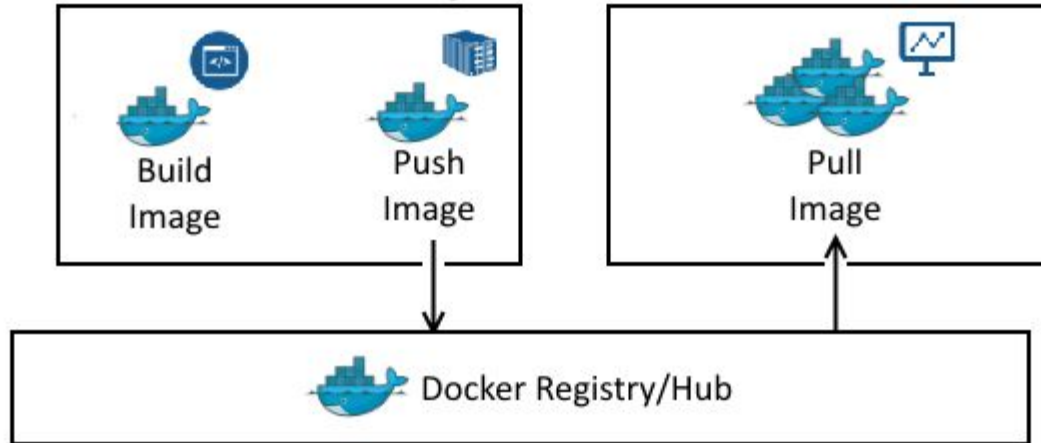
```
docker container run -ti first_image bash
```







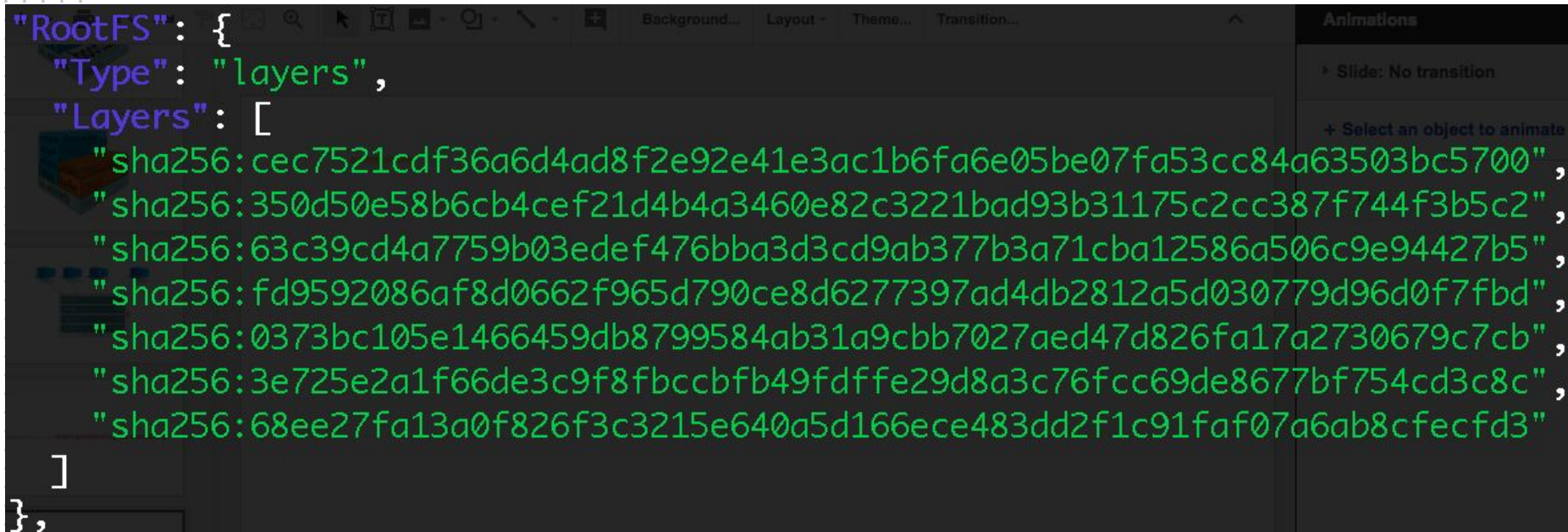






```
docker image inspect mercadolibre/nginx-workshop
```

```
https://hub.docker.com/r/mercadolibre/nginx-workshop/
```



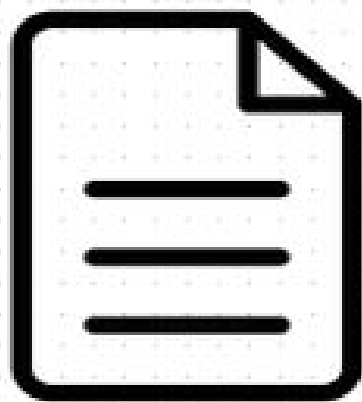


```

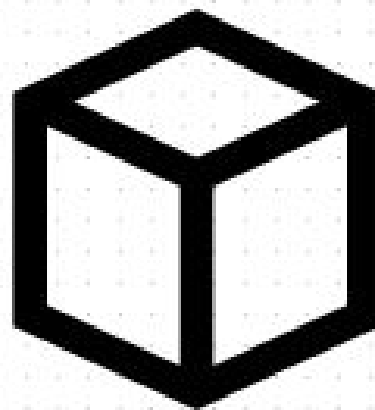
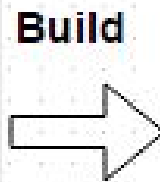
[localhost:nginx-workshop geisbruch$ docker history -H mercadolibre/nginx-workshop
IMAGE                CREATED              CREATED BY          SIZE
4cc598be279d        2 hours ago        /bin/sh -c mkdir /myhtmls          0B
aa17f1933c69        2 hours ago        /bin/sh -c #(nop) ADD file:1ef0b9e93976b59... 325B
0aee73ed8431        2 hours ago        /bin/sh -c ln -sf /dev/stdout /var/log/ngi... 22B
89c1bd98ecdc        2 hours ago        /bin/sh -c rm -fr /dev/stdout /dev/stderr    0B
53eca1ab7069        2 hours ago        /bin/sh -c apt update && apt install -y -f... 73.2MB
9e7424e5dbae        2 days ago        /bin/sh -c #(nop) CMD ["nginx" "-g" "daem... 0B
<missing>           2 days ago        /bin/sh -c #(nop) STOPSIGNAL [SIGTERM]        0B
<missing>           2 days ago        /bin/sh -c #(nop) EXPOSE 80/tcp              0B
<missing>           2 days ago        /bin/sh -c ln -sf /dev/stdout /var/log/ngi... 22B
<missing>           2 days ago        /bin/sh -c set -x && apt-get update && a... 53.2MB
<missing>           2 days ago        /bin/sh -c #(nop) ENV NJS_VERSION=1.13.7.... 0B
<missing>           2 days ago        /bin/sh -c #(nop) ENV NGINX_VERSION=1.13.... 0B
<missing>           2 weeks ago       /bin/sh -c #(nop) LABEL maintainer=NGINX ... 0B
<missing>           3 weeks ago       /bin/sh -c #(nop) CMD ["bash"]              0B
<missing>           3 weeks ago       /bin/sh -c #(nop) ADD file:45233d6b5c9b91e... 55.3MB

```

# Dockerfile



**Dockerfile**



**Image**



```
Dockerfile
```

```
1 FROM debian:stable
```


Diagram illustrating the components of the Dockerfile instruction `FROM debian:stable`:

- `FROM`: Instruccion
- `debian`: Imagen
- `:stable`: Tag



```
docker image build -t demo1:latest .
```

```
docker container run -ti demo1:latest  
bash
```

- 
- debian, ubuntu, centos, etc
  - scratch
  - alpine, busybox
  - Otros



Armemos una imagen que haga cosas copadas

Por ejemplo, que tire un **ping** !!!

```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f net-tools
6
7
8
9
```



```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools
6 #Change entrypoint
7 ENTRYPOINT ["/bin/ping"]
8
```



```
docker image build -t myping .
```

```
docker container run myping  
mercadolibre.com
```

```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools
6 #Change entrypoint
7 ENTRYPOINT ["/bin/ping"]
8 #Change cmd
9 CMD ["mercadolibre.com"]
```

10

A decorative graphic on the left side of the slide. It consists of a vertical grid of small dots on the far left, and a solid yellow horizontal bar positioned above the text.

docker container run myping



Pasemos a otra imagen un poco más compleja

```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools nginx
6 #Entrypoint nginx
7 ENTRYPOINT ["/usr/sbin/nginx"]
8 #CMD config file in /configs
9 CMD ["-c","/configs/nginx.conf"]
```

10

```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools nginx
6 #Entrypoint nginx
7 ENTRYPOINT ["/usr/sbin/nginx"]
8 #CMD config file in /configs
9 CMD ["-c", "/configs/nginx.conf"]
10 #Create app, logs and configs directories
11 RUN mkdir -p /app/logs /configs
12 #Add config
13 ADD ./nginx.conf /configs
14
```

```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools nginx
6 #Entrypoint nginx
7 ENTRYPOINT ["/usr/sbin/nginx"]
8 #CMD config file in /configs
9 CMD ["-c","/configs/nginx.conf"]
10 #Create app, logs and configs directories
11 RUN mkdir -p /app/logs /configs
12 #Add config
13 ADD ./nginx.conf /configs
14 #Add app
15 ADD ./app /app
16 |
```



```
1 FROM debian:stable
2 #Update apt packages
3 RUN apt-get update
4 #Install curl and net-tool commands
5 RUN apt-get install -y -f curl net-tools nginx
6 #Entrypoint nginx
7 ENTRYPOINT ["/usr/sbin/nginx"]
8 #CMD config file in /configs
9 CMD ["-c","/configs/nginx.conf"]
10 #Create app, logs and configs directories
11 RUN mkdir -p /app/logs /configs
12 #Add config
13 ADD ./nginx.conf /configs
14 #Add app
15 ADD ./app /app
16 #Re-link stderr && stdout
17 RUN ln -sf /dev/stdout /app/logs/access.log \
18     && ln -sf /dev/stderr /app/logs/error.log
```

```
8  #END config file in /configs
9  CMD ["-c", "/configs/nginx.conf"]
10 #Create app, logs and configs directories
11 RUN mkdir -p /app/logs /configs
12 #Add config
13 ADD ./nginx.conf /configs
14 #Add app
15 ADD ./app /app
16 #Re-link stderr && stdout
17 RUN ln -sf /dev/stdout /app/logs/access.log \
18     && ln -sf /dev/stderr /app/logs/error.log
19 #Port
20 EXPOSE 80
21
```



```
docker image build -t mynginx .
```

```
docker container run -P mynginx
```



**Mirar siempre Base Images !!!**

[https://hub.docker.com/\\_/nginx/](https://hub.docker.com/_/nginx/)

<https://github.com/nginxinc/docker-nginx>

Tengo muchas apps que usan nginx,  
copy/paste ???

Armemos una base image




```
12  #Add config
13  ADD ./nginx.conf /configs
14  #Re-link stderr && stdout
15  RUN ln -sf /dev/stdout /app/logs/access.log \
16      && ln -sf /dev/stderr /app/logs/error.log
17  #Nginx Port
18  EXPOSE 80
19  #Add app
20  ONBUILD ADD ./app /app
21
```

```
1  FROM mynginxbase:latest
```

```
2
```

A decorative graphic on the left side of the slide, consisting of a vertical grid of small dots and a horizontal yellow bar.

Vamos con un ejercicio ?

A decorative vertical dotted pattern on the left side of the slide and a short horizontal orange bar above the text.

Ok, esto esta genial, pero como hago  
si tengo otros servicios ?



# Docker Compose

```
1 version: '3'
2 services:
3   web:
4     build: .
5     command: start-dev
6     ports:
7       - "5000:5000"
8     environment:
9       REDIS_HOST: redis
10    depends_on:
11      - redis
12    volumes:
13      - ./nodeapp:/app
14  redis:
15    image: redis:4.0
16    command: --appendonly yes
17    volumes:
18      - ./data:/data
```

```
1 FROM debian:stable
```

```
2  
3 ADD ./node-v9.2.0-linux-x64.tar.xz /usr/local/bin
```

```
4 ENV PATH=$PATH:/usr/local/bin/node-v9.2.0-linux-x64/bin
```

```
5 RUN ln -s /usr/local/bin/node-v9.2.0-linux-x64/bin/node /usr/local/bin/node-v9.2
```

```
6 RUN mkdir /utils
```

```
7 ADD ./run_app.sh /utils
```

```
8 ENTRYPOINT ["/utils/run_app.sh"]
```

```
9 EXPOSE 5000
```

```
10 ADD ./nodeapp /app
```

```
11 WORKDIR /app
```

```
12 RUN npm install
```

```
13 ENV REDIS_HOST=redis
```

```
14
```

A decorative element on the left side of the slide, consisting of a vertical grid of small dots and a solid horizontal orange bar.

`docker-compose build`



`docker-compose up`



docker-compose ps



No expusimos el puerto de redis ?




docker network ls

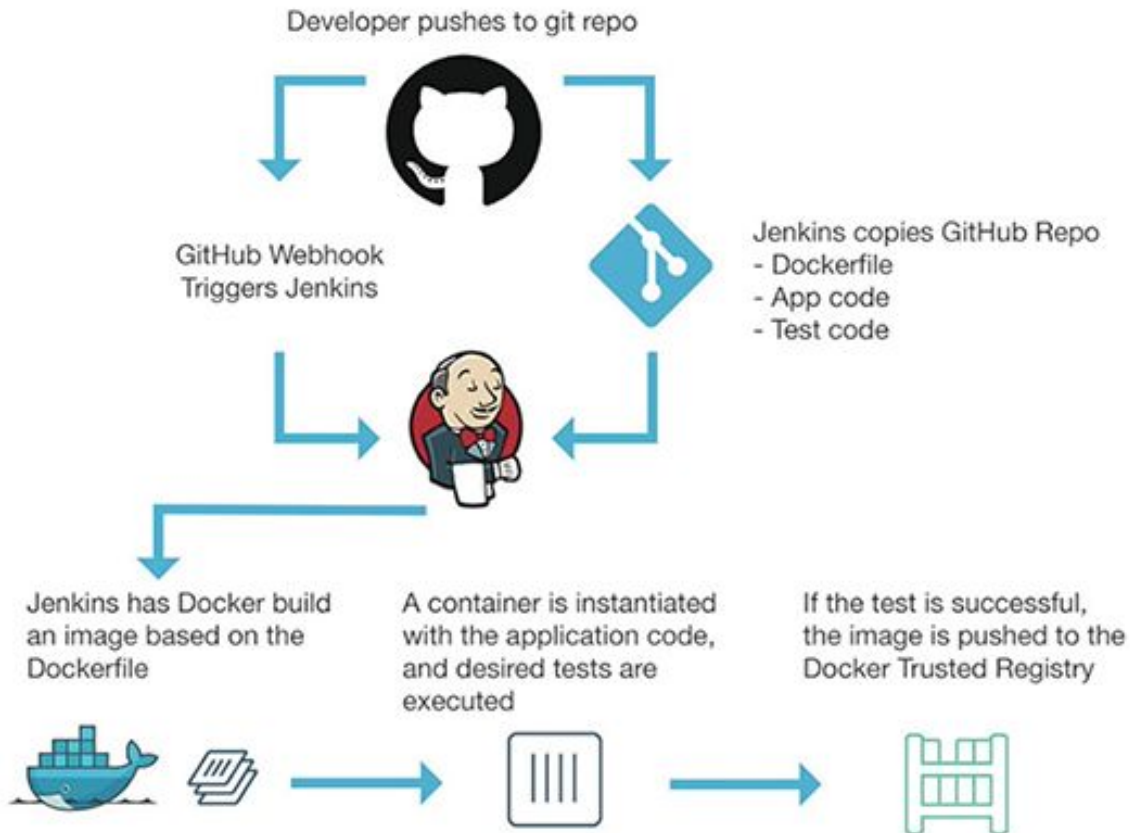
docker network inspect



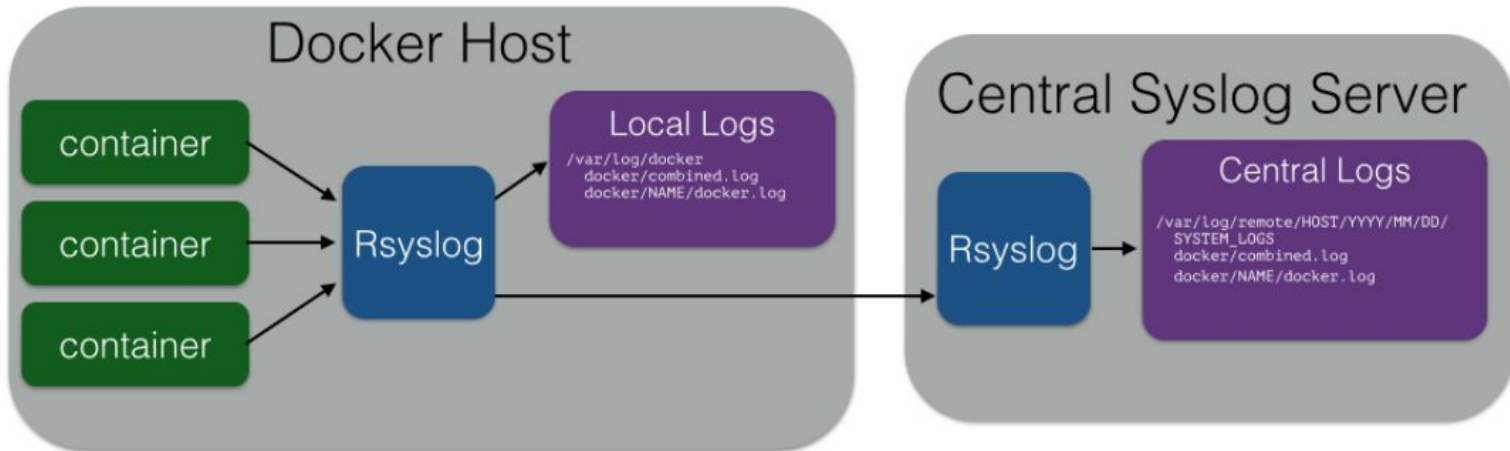
# Producción

- 
- Builds / Deployments?
  - Logs ?
  - Ops ?

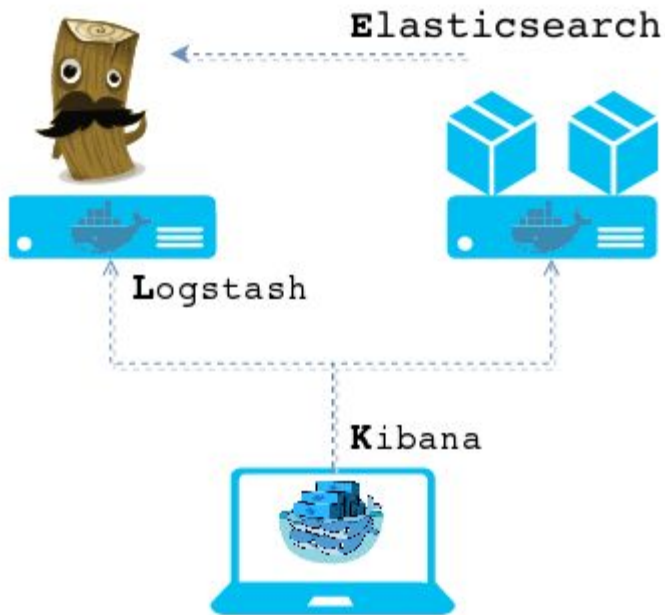
# Builds



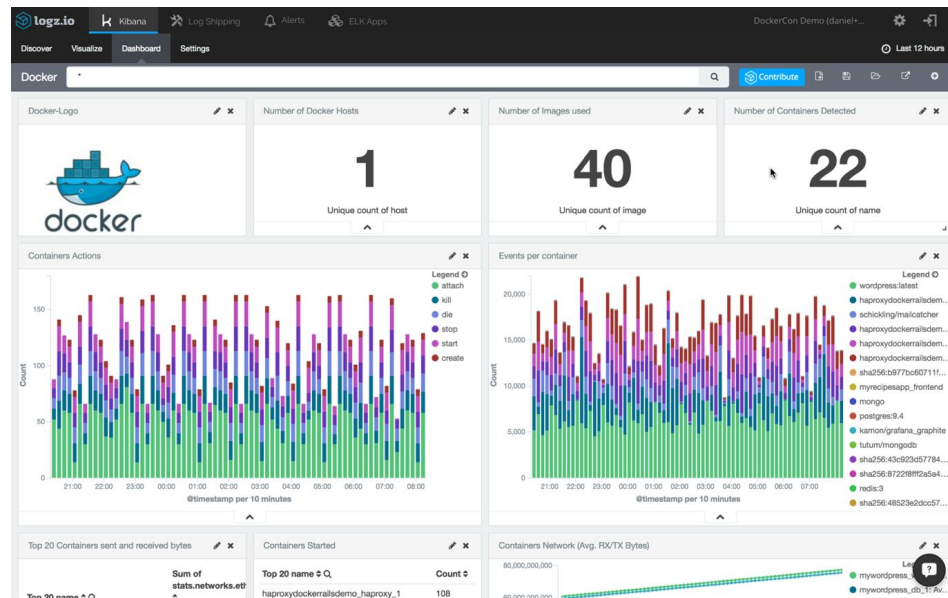
# Logs



# Logs



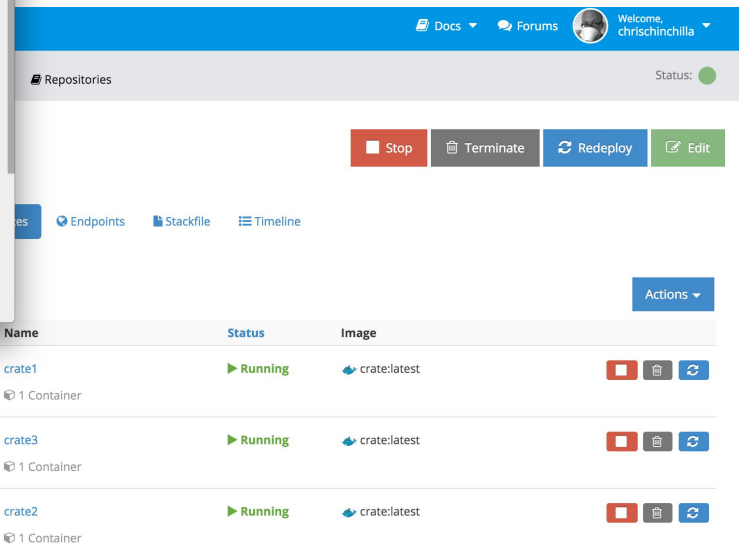
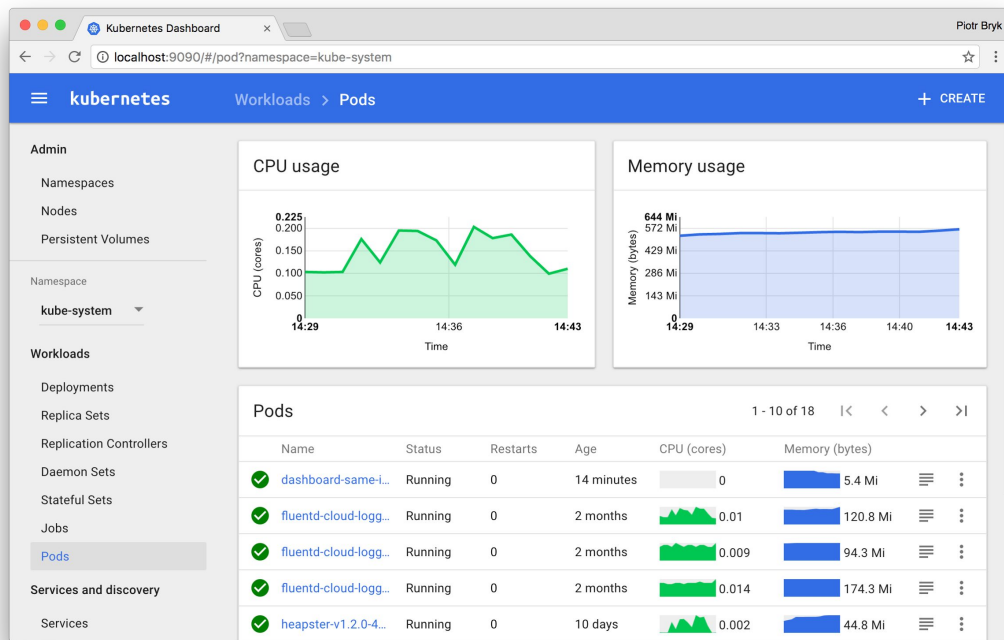
... running in a swarm of docker containers.



# Ops

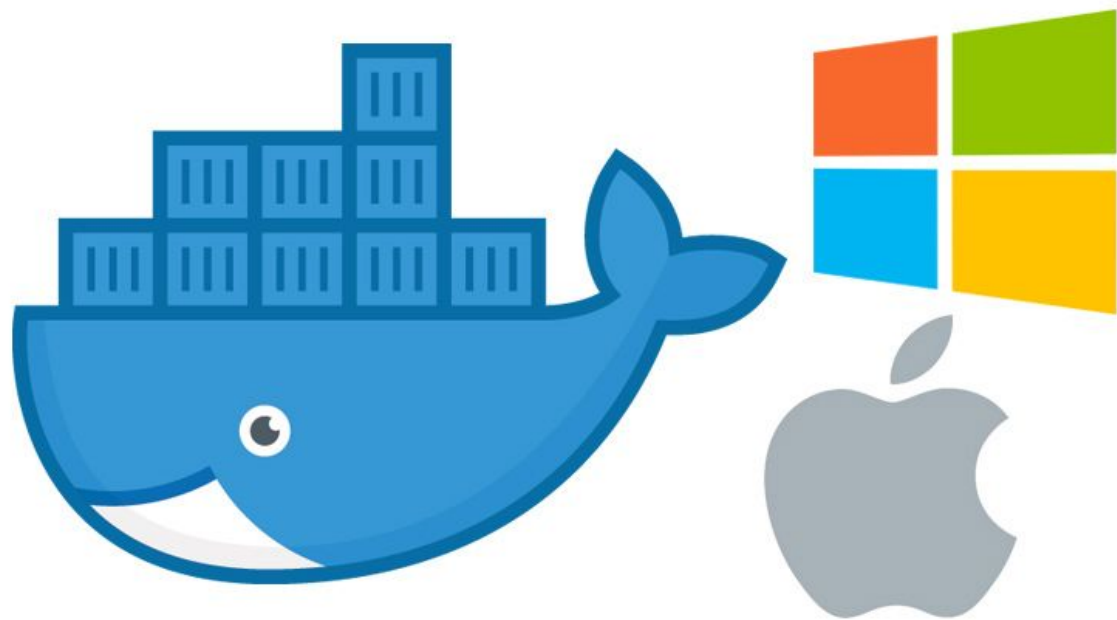
- Single instances
- Swarm / Kubernetes
- ECS/GCE Kubernetes/AKS / Docker Cloud

# Ops









# Como estamos usando docker

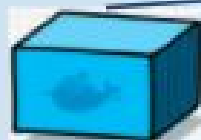
VPC-1

Subnet A app 1

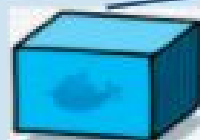
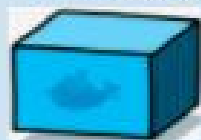
Server i-0XXXXXX

local-api

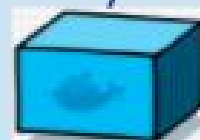
mercadolibre/docker-meetu  
p:0.0.1



rsyslog



JMX



datadog

**docker-compose.yml**



## Logs

Filter:

Quick filter logs

Scope:

Version:

timestamp

message



	deraul: pid=28378 comm= ps requested_mask= trace denied_mask= trace peer= docker-deraul
2015-08-07T18:30:54.692Z	[170653.175308] audit: type=1400 audit(1438972254.688:70312): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=28378 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:54.692Z	[170653.175364] audit: type=1400 audit(1438972254.688:70313): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=28378 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:55.048Z	[170652.749298] audit: type=1400 audit(1438972255.036:70362): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=28471 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:55.508Z	[168696.027856] audit: type=1400 audit(1438972255.504:69506): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=29213 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:55.508Z	[168696.027800] audit: type=1400 audit(1438972255.504:69505): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=29213 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:55.508Z	[168696.027739] audit: type=1400 audit(1438972255.504:69504): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=29213 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:55.508Z	[168696.027254] audit: type=1400 audit(1438972255.504:69503): apparmor="DENIED" operation="ptrace" profile="docker-default" pid=29213 comm="ps" requested_mask="trace" denied_mask="trace" peer="docker-default"
2015-08-07T18:30:56.967Z	[168696.155670] audit: type=1400 audit(1438972256.937:69507): apparmor="DENIED" operation="ptrace" profile="docker-

## Create scope

Application

apparel-home

Name

test1

Instance  
type



m3.medium (CPUs: 1, RAM: 3.75 GB, SSD: 1 x 4)

Min  
instances



2

Max  
instances



10

Version



0.0.2-test

Deploying from 2.1.0-brancha to 2.0.0

pending

creating

swapping

effective

Remaining to swap out ⓘ

1 / 1

Swapped in ⓘ

0 / 1

Swapping progress ⓘ

0%

Rollback

Finish

Metrics

The past 15 minutes ▾



Hide legends

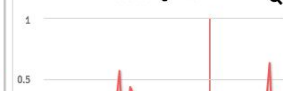
Request Count

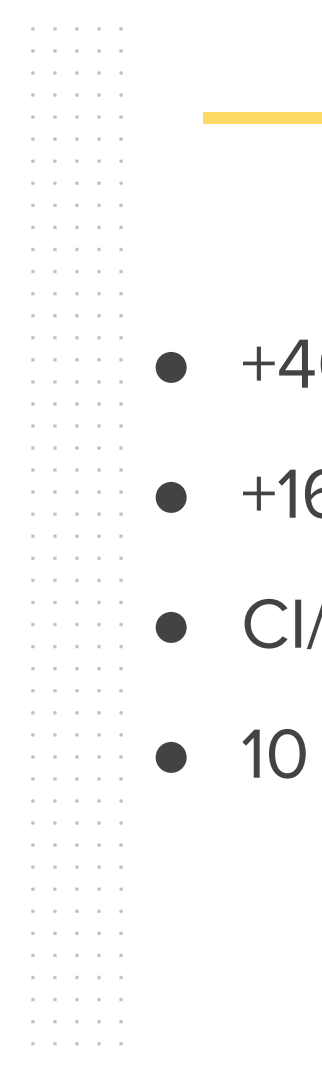


Errors Count





Processing Time



- 
- +40000 containers
  - +1600 apps
  - CI/CD
  - 10 Servicios integrados

# Algunas lecciones aprendidas

- 
- 
- Imágenes de tamaño pequeño
  - Docker registry propio
  - Imágenes inmutables (no instalar o hacer nada cuando levantan)
  - Variables de entorno para configuraciones
  - Armar imágenes base (aunque no hagan nada)





# We are hiring !!!

[gabriel.eisbruch@mercadolibre.com](mailto:gabriel.eisbruch@mercadolibre.com)

[sebastian.schepens@mercadolibre.com](mailto:sebastian.schepens@mercadolibre.com)

