



# ARQUITECTURA Y SISTEMAS OPERATIVOS

---

Docker

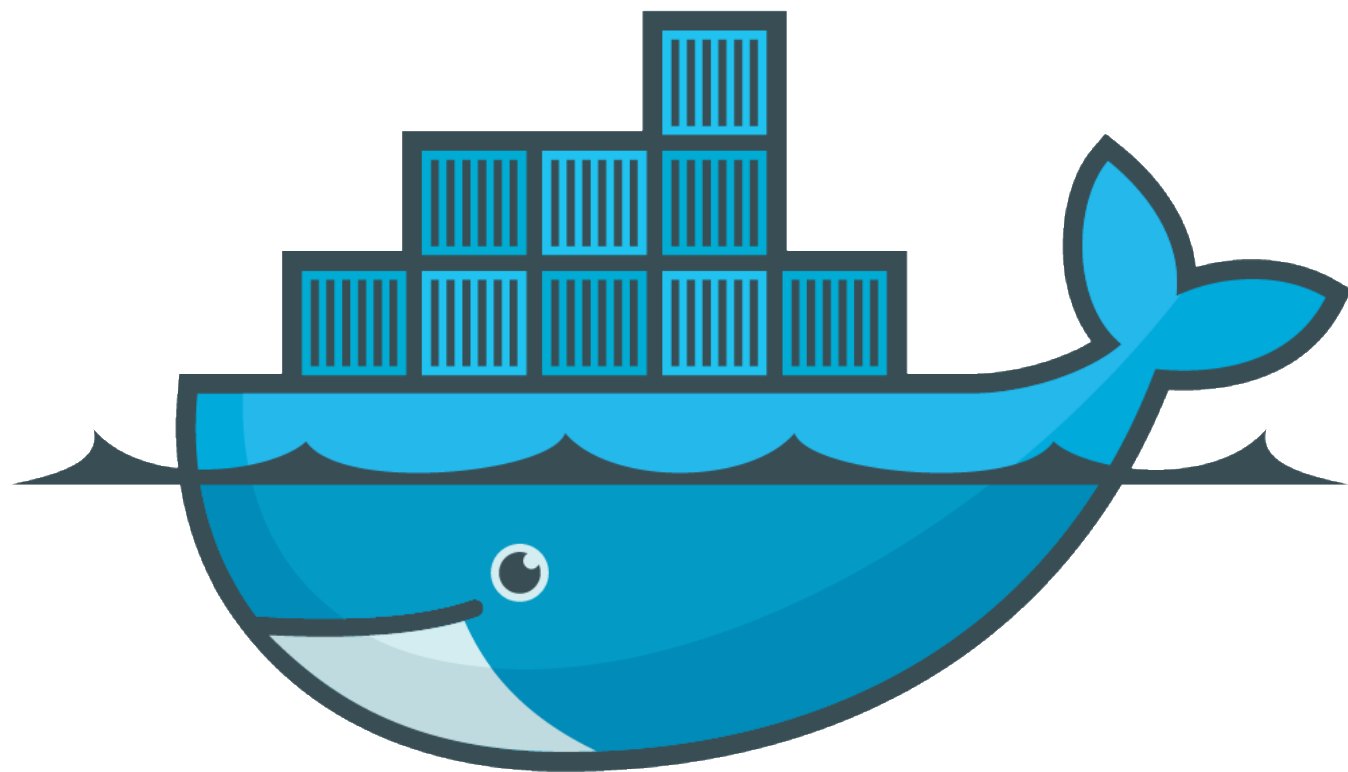
# PROFESORES



MARCOS PABLO RUSSO  
(MARCOSPR1974@GMAIL.COM)



MARTIN ISUSI SEFF  
(MARTINISUSI@GMAIL.COM)



docker

# ¿ QUE ES DOCKER ?

- Docker es una herramienta tanto para el sysadmin como developer.
- Ayuda a automatizar el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa de abstracción eliminando el overhead que provoca correr un Sistema operativo virtual para cada una sola aplicación.
- Además de la propia aplicación que gestiona los contenedores (a los que llamamos docker engine o docker), existe un repositorio de imágenes creadas por la comunidad, llamada docker hub.
- Tanto los developers como los sysadmin pueden compartir entornos de trabajo (imágenes).

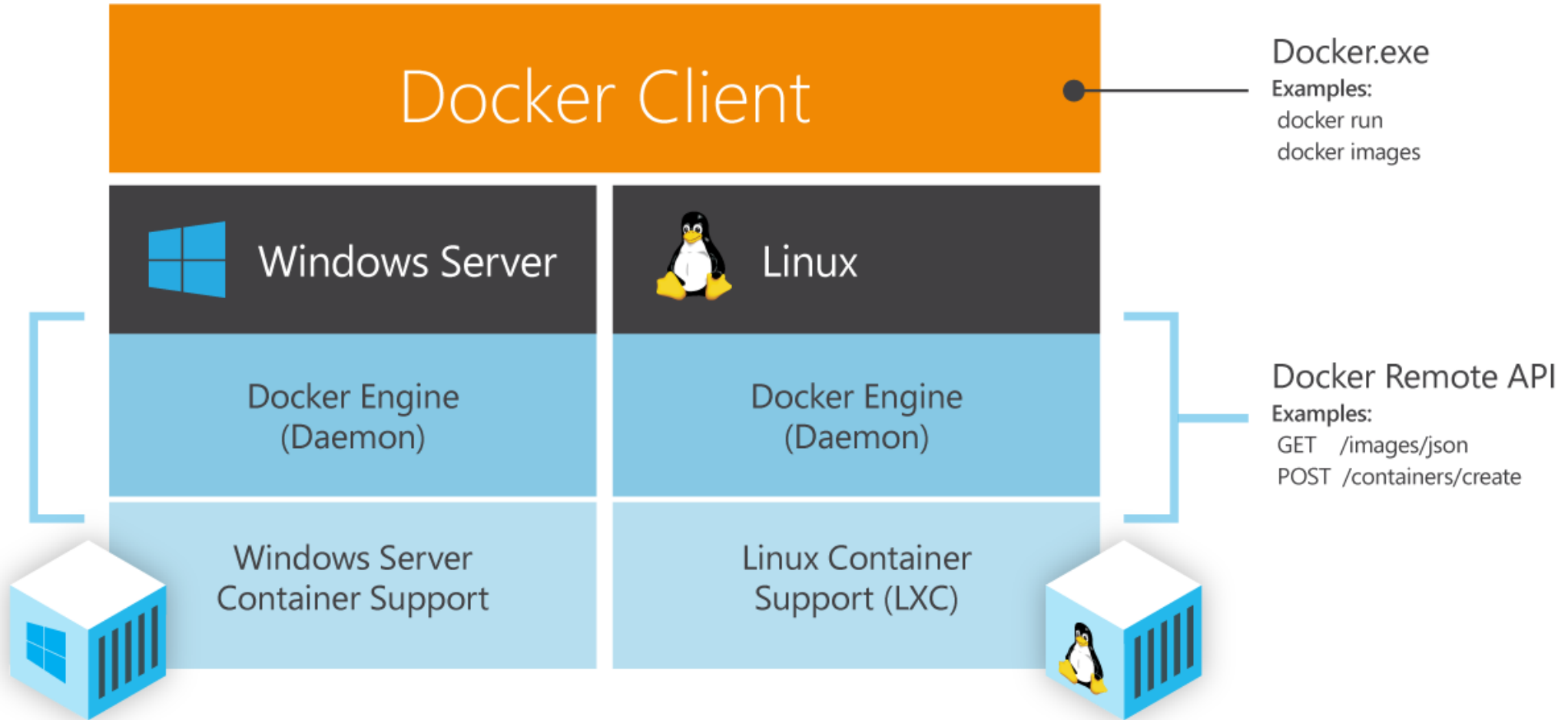
# ¿ QUE ES DOCKER ?

- Corre en equipos de 64 bits.
- Gran parte de éxito de docker se basa en su fácil portabilidad y su ligereza.
- Soporta los sistemas operativos Windows, Mac y obviamente GNU/Linux.
- Docker es tanto cliente como servidor.

# ¿ QUE NO ES DOCKER ?

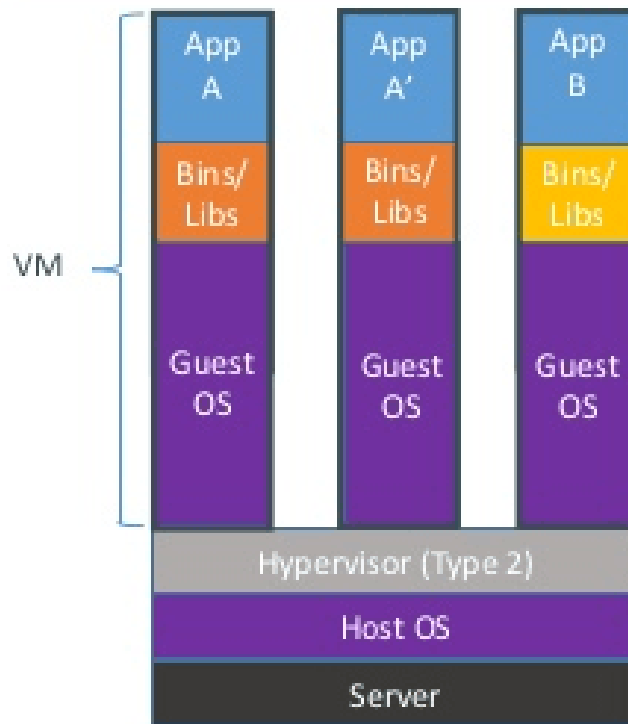
- Un gestor de entornos de desarrollo virtuales (vagrant).
- Un software de virtualización (hypervisor).
  - (kvm, xen, vmware, virtualbox, etc).
- Un gestor de configuración (puppet/chef).
- Un simple contenedor de software (lxc).
- Aunque se parece como servidor.

# ¿ QUE ES DOCKER ?

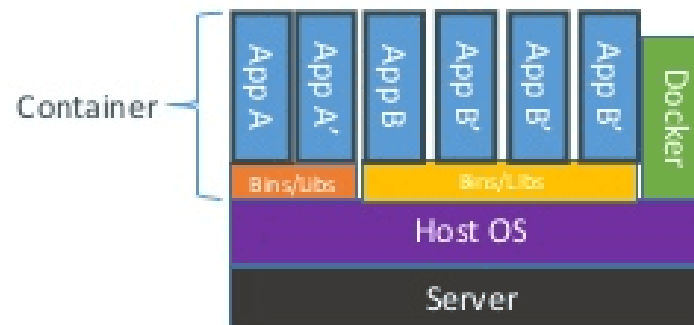


# ¿ QUE ES DOCKER ?

## Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries





# ¿ QUE ES DOCKER ?

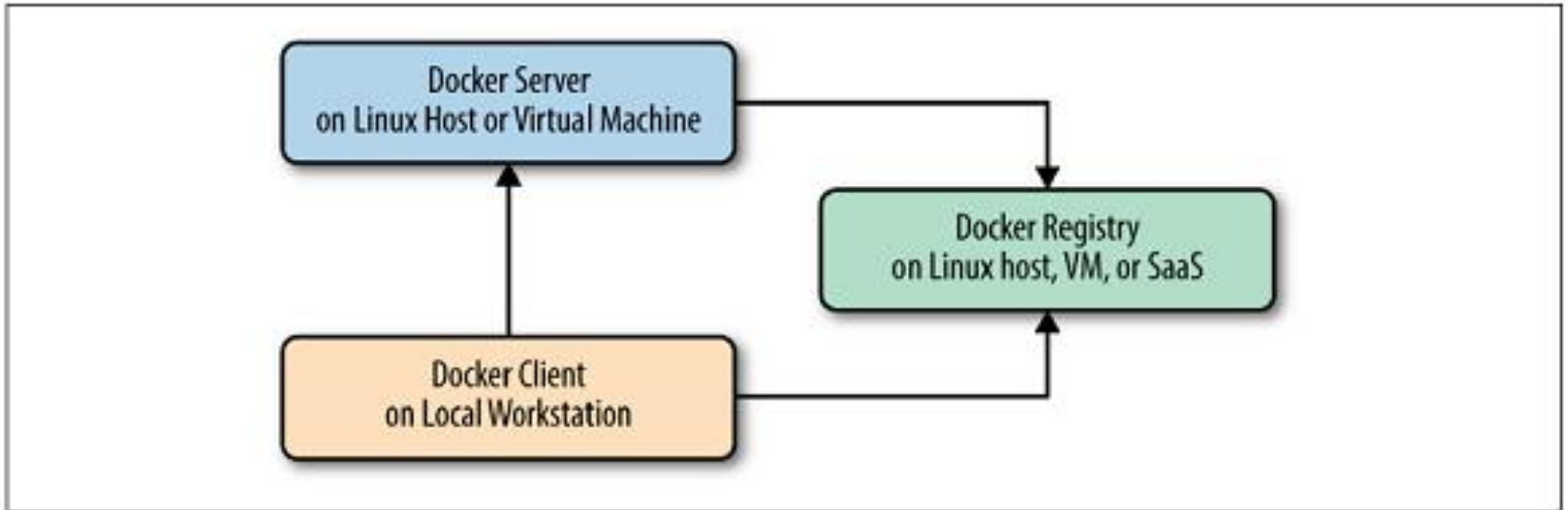
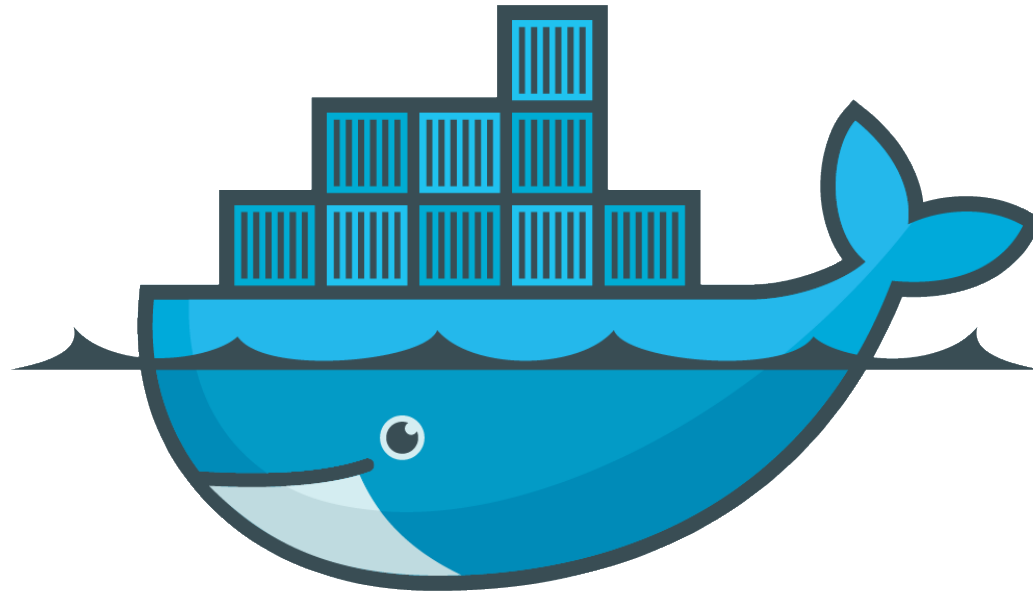


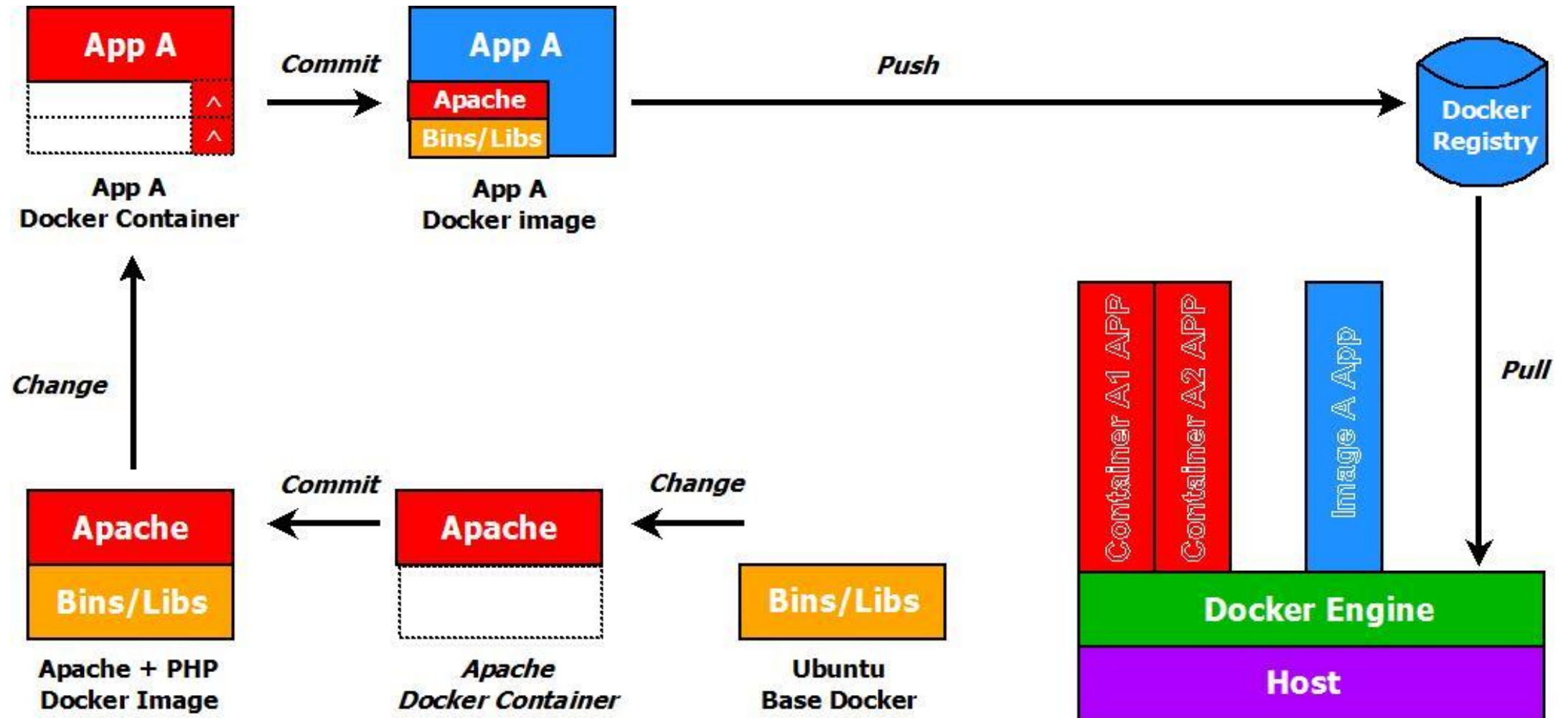
Figure 2-3. Docker client/server model

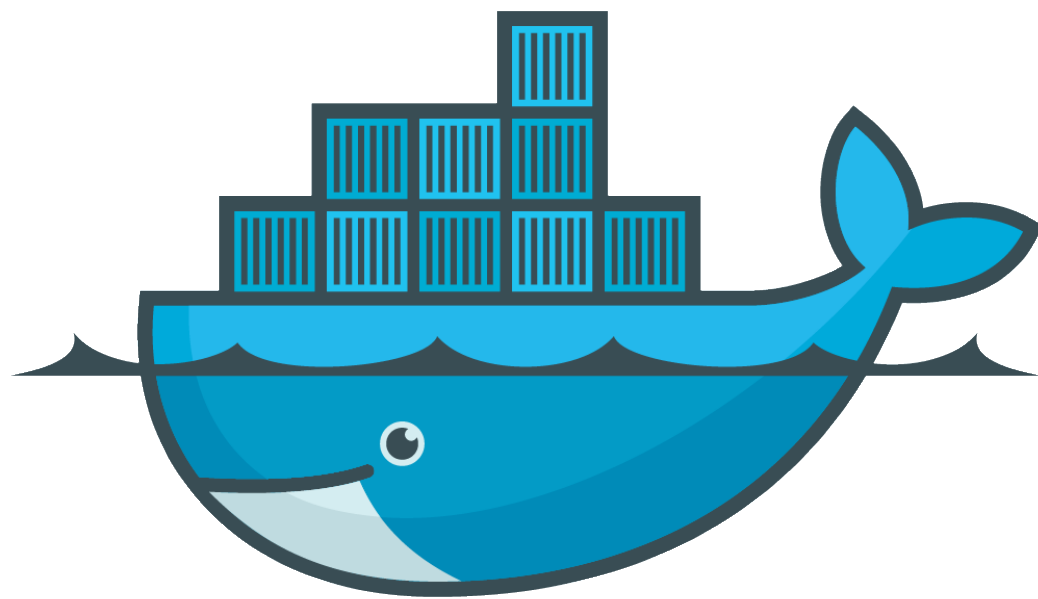


docker

CICLO DE VIDA

# CICLO DE VIDA





docker

WEB

# ARQUITECTURA DE DOCKER

- WEB

- <https://www.docker.com>

- HUB

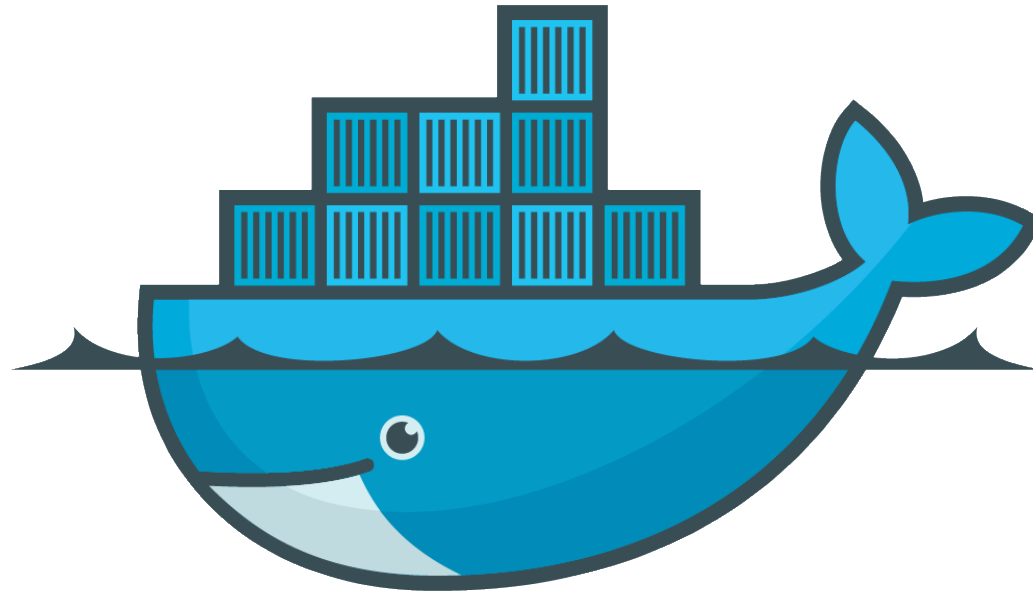
- <https://hub.docker.com>

- DOCUMENTACION

- <https://docs.docker.com>

- API

- <https://docs.docker.com/engine/api/>



docker

INSTALACION

# INSTALACION

- Ubuntu
  - apt-get install docker.io
- Centos/OS
  - yum -y install docker docker-commn container-selinux
- Debian
  - apt-get install docker-engine

<https://docs.docker.com/engine/installation/linux/>

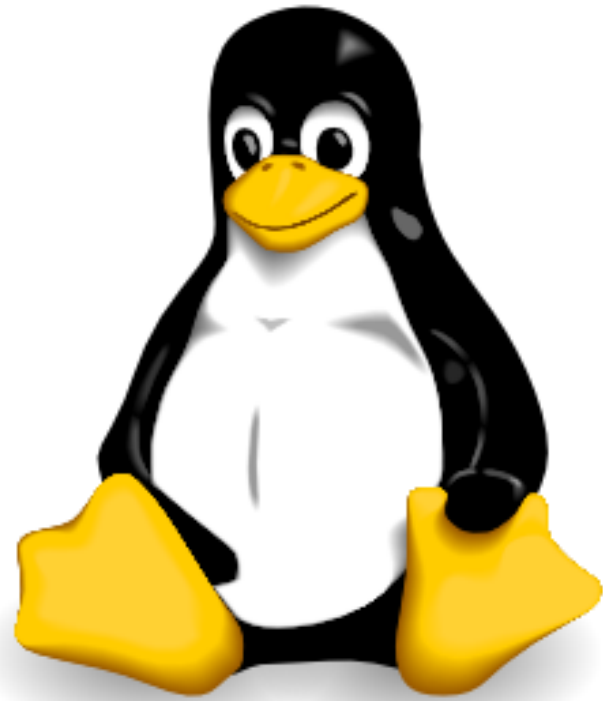
# INSTALACION

```
localhost:~# ps aux | grep docker
2106 root      0:21 /usr/bin/dockerd -p /run/docker.pid
2237 root      0:36 containerd --config /var/run/docker/containerd/containerd.toml --log-level info
2701 root      0:00 grep docker
```



# INSTALACION

```
# usermd -G docker -a USUARIO
```



GNU/Linux

# COMANDOS

# ifconfig

# route

```
localhost:~# ifconfig
docker0  Link encap:Ethernet  HWaddr 02:42:05:C7:D4:60
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0     Link encap:Ethernet  HWaddr 08:00:27:9D:FD:59
          inet addr:192.168.1.109  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe9d:fd59/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10072 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1517 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3094468 (2.9 MiB)  TX bytes:164694 (160.8 KiB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
localhost:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 192.168.1.1 0.0.0.0 UG 202 0 0 eth0
172.17.0.0 * 255.255.0.0 U 0 0 0 docker0
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
```

# COMANDOS

# iptables -L -n

```
localhost:~# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy DROP)
target     prot opt source                destination
DOCKER-USER all  --  0.0.0.0/0             0.0.0.0/0
DOCKER-ISOLATION-STAGE-1 all  --  0.0.0.0/0             0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0             0.0.0.0/0             ctstate RELATED,ESTABLISHED
DOCKER     all  --  0.0.0.0/0             0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0             0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0             0.0.0.0/0

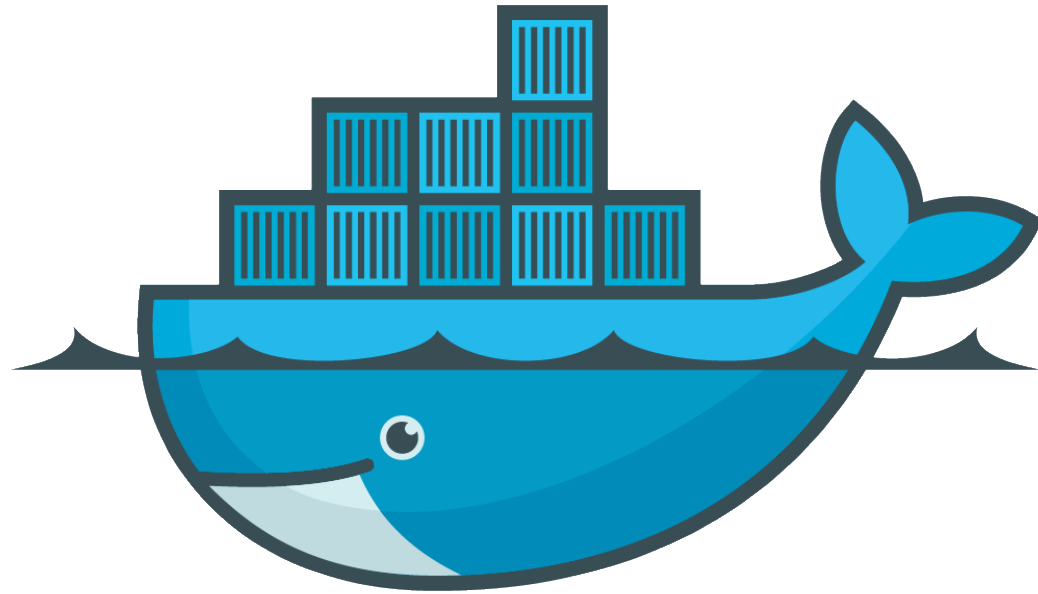
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain DOCKER (1 references)
target     prot opt source                destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target     prot opt source                destination
DOCKER-ISOLATION-STAGE-2 all  --  0.0.0.0/0             0.0.0.0/0
RETURN     all  --  0.0.0.0/0             0.0.0.0/0

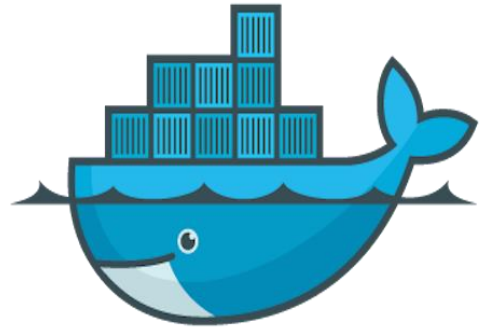
Chain DOCKER-ISOLATION-STAGE-2 (1 references)
target     prot opt source                destination
DROP       all  --  0.0.0.0/0             0.0.0.0/0
RETURN     all  --  0.0.0.0/0             0.0.0.0/0

Chain DOCKER-USER (1 references)
target     prot opt source                destination
RETURN     all  --  0.0.0.0/0             0.0.0.0/0
```



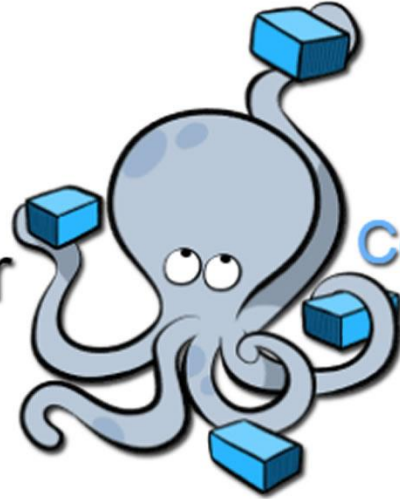
docker

Herramientas



docker

Docker



Compose



docker  
REGISTRY

TM & © 2015 Docker, Inc.



docker  
SWARM



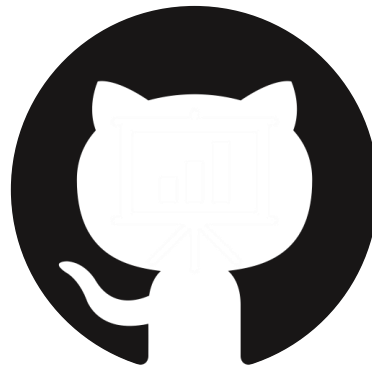
docker  
MACHINE

TM & © 2015 Docker, Inc.

# RECURSOS



SO-UTNFRA.SLACK.COM



[GITHUB.COM/MARTIN919191/ARQUITECTUR  
AYSISTEMASOPERATIVOS](https://github.com/MARTIN919191/ARQUITECTURA-SISTEMAS-OPERATIVOS)



SO.UTNFRA@GMAIL.COM