

# Docker & Wordpress

*“Juntos pero no revueltos”*



## ¿Quién soy?

- **Técnico de sistemas**
- **Formador**
- **Apasionado de la tecnología**



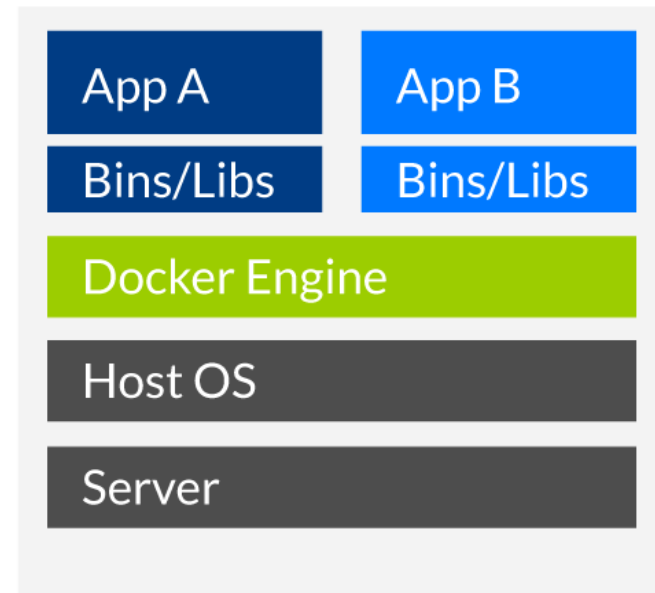
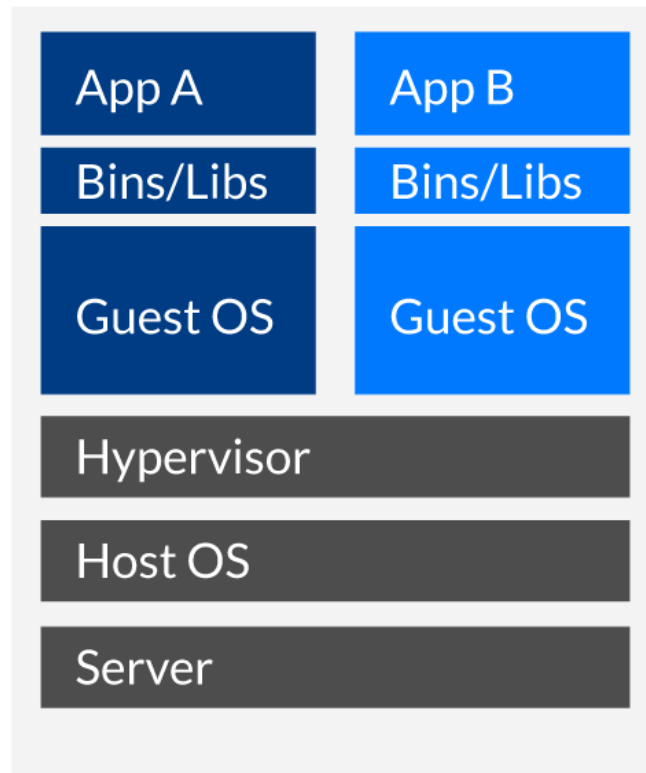
Mi trabajo se centra principalmente en la implementación de los sistemas y el asesoramiento técnico.

## Agenda del día ...

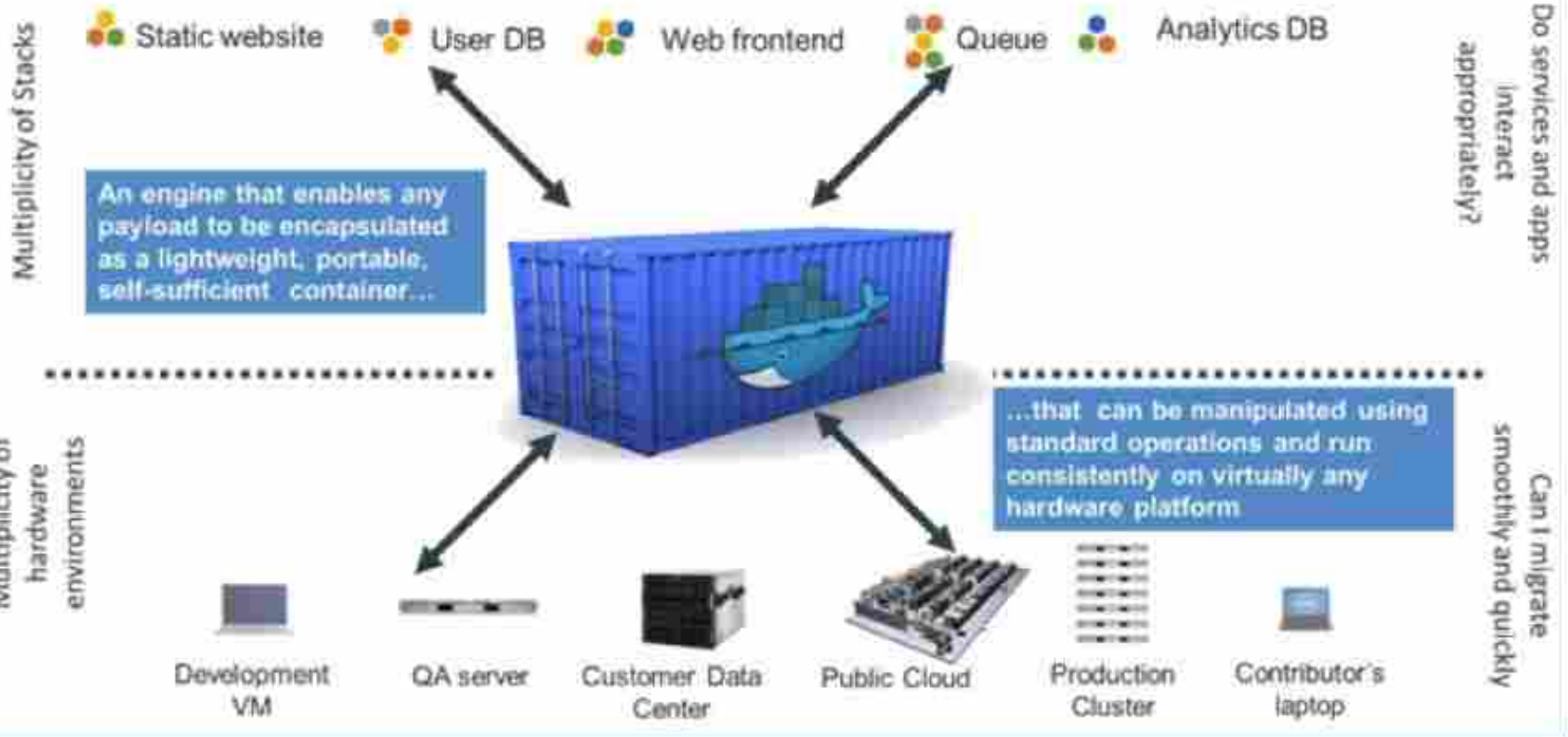
- **¿Qué es docker?**
- **Nuestra infraestructura en AWS**
- **Como desplegar servicios**
- **Multicontenedores con docker-compose**
- **Nuestro cluster con swarm**

## ¿Qué es Docker?

### VMs vs Docker



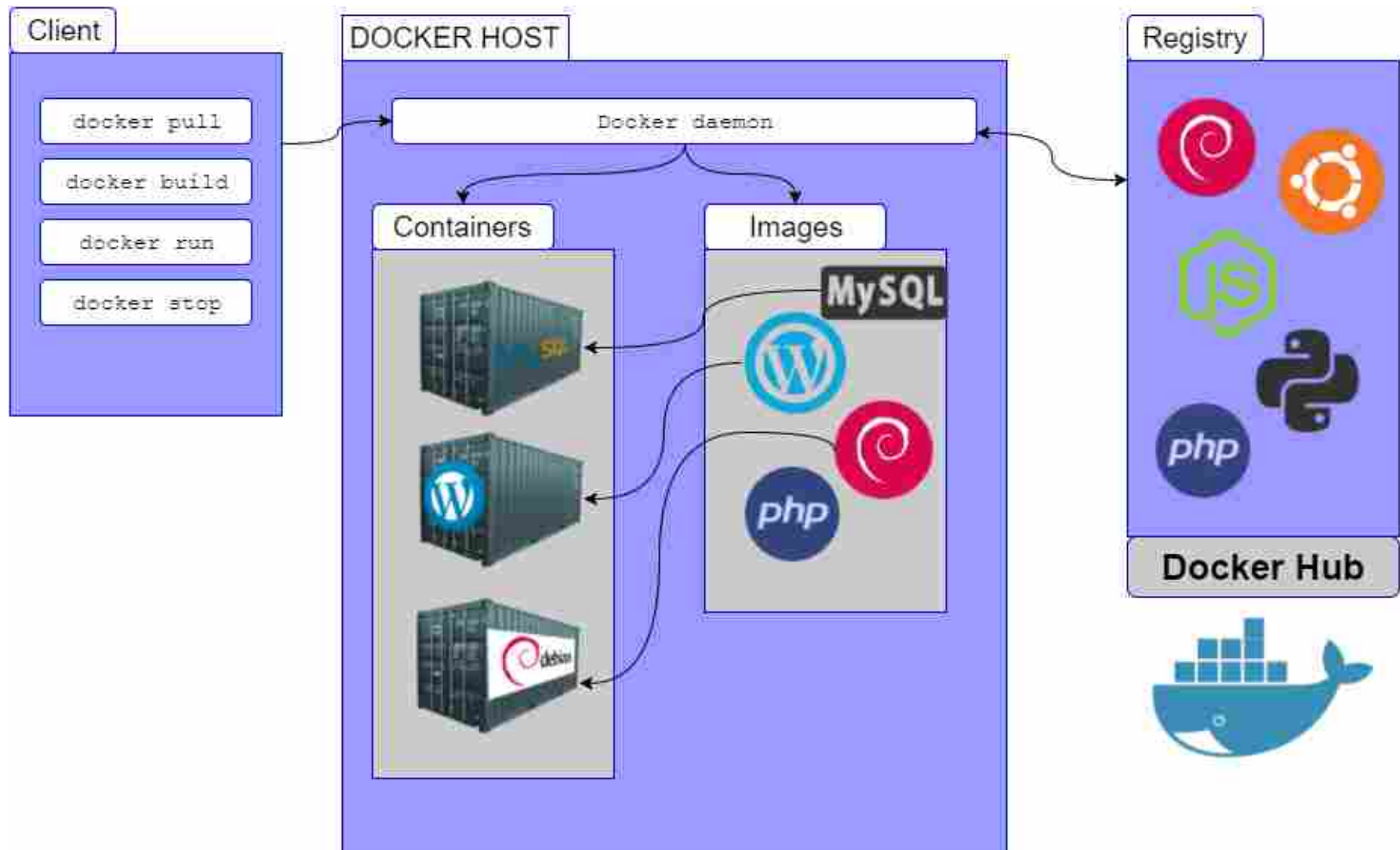
## Docker is a shipping container system for code



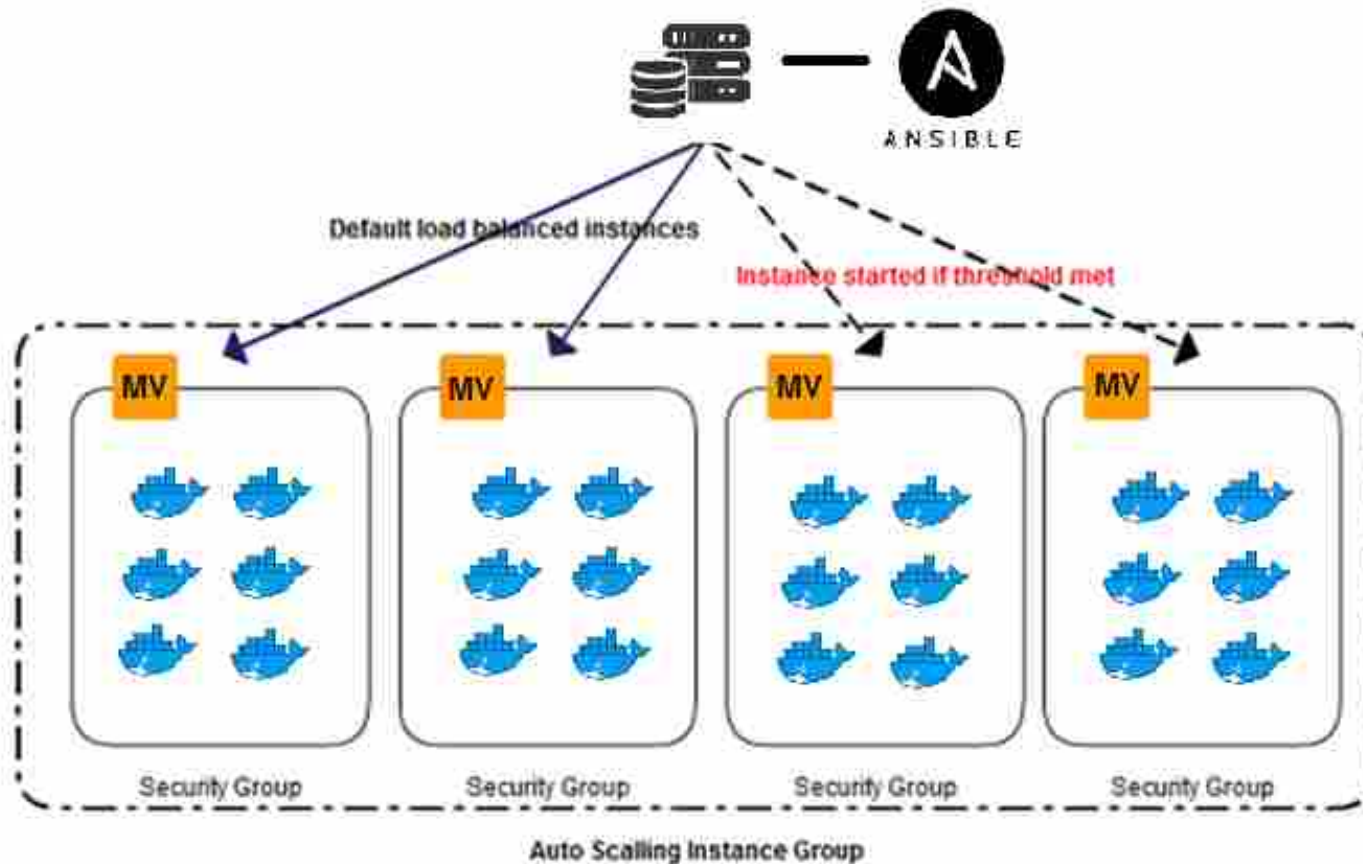
# Docker & Wordpress “Juntos pero no revueltos”

✉ hola@juanregueras.es

🐦 @ReguerasJuan



## Nuestra infraestructura en AWS



## ¿Cómo se administra todo esto?

### DockerUI



### Containers:

Actions ▾  Display All

🗑	Name	Image	Command	Created	Status
🗑	dockerui	abh1nav/dockerui:latest	./dockerui -e /docker.sock	Wed Sep 10 2014	Up About a minute
🗑	redis5	dockerfile/redis:latest	redis-server /etc/redis/redis.conf	Wed Sep 10 2014	Up About a minute



portainer.io

NAVIGATION

- Dashboard
- App Templates
- Containers
- Images
- Networks
- Volumes

Container list [Refresh](#)

Containers

Start Stop Kill Restart Pause Resume Remove Add container

State	Name	Image	IP Address
running	consul3	consul	-
running	consul2	consul	-



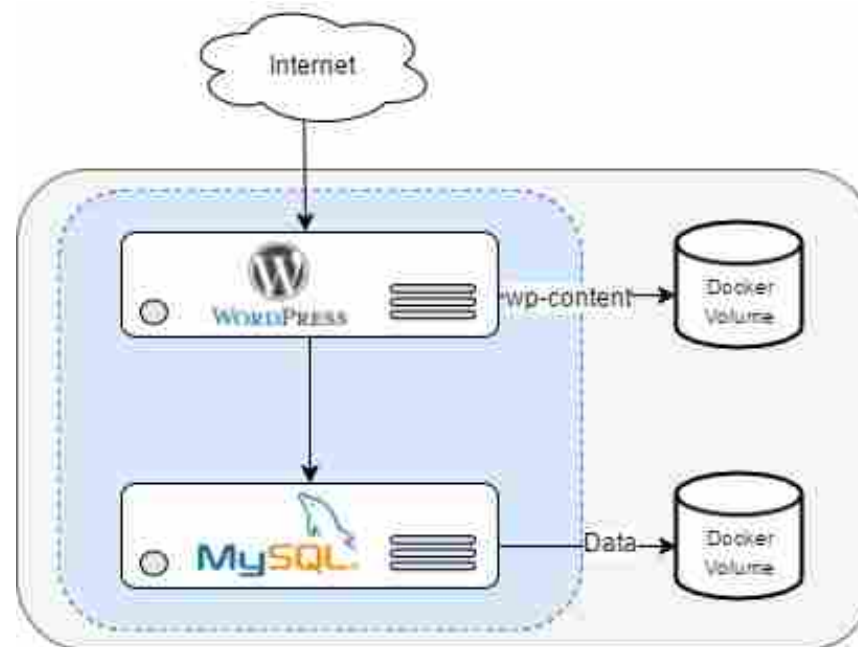
## Como desplegar servicios

### Creamos el contenedor con la base de datos

```
docker run -d -e MYSQL_ROOT_PASSWORD=admin --name bbdd mysql
```

### Creamos el contenedor con el wordpress

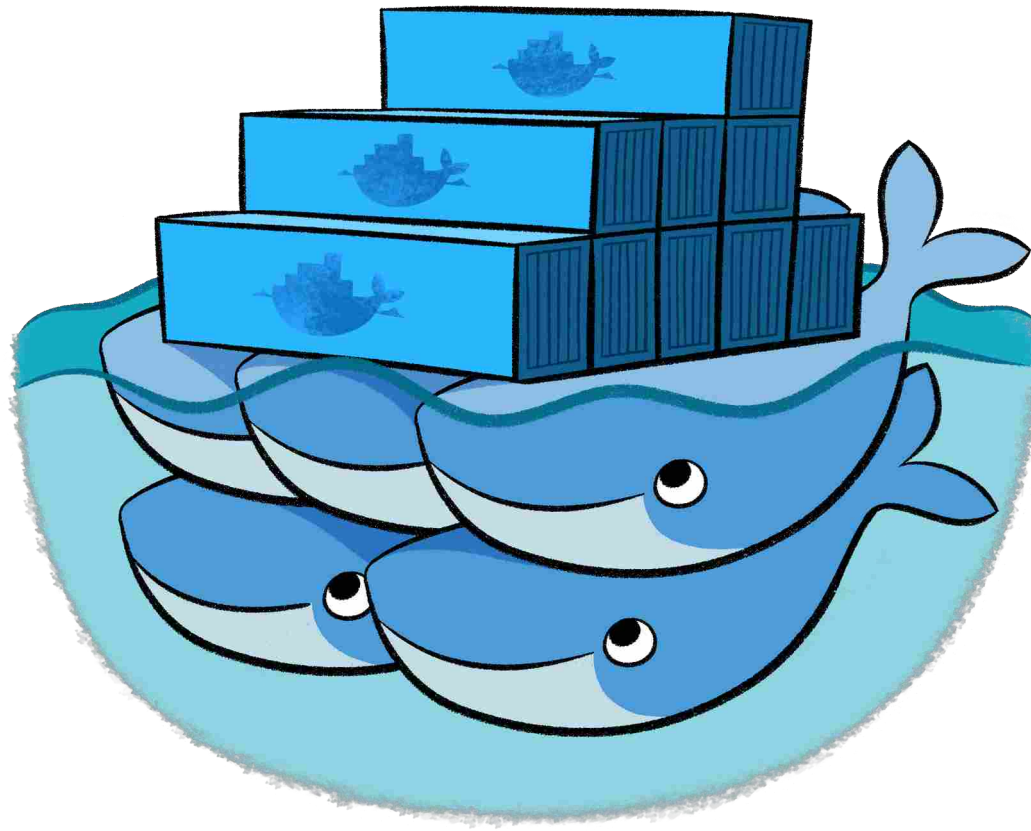
```
docker run -d --name web --link bbdd:mysql -p 90:80 -v wp_vol:/var/www/html wordpress
```



## Multicontenedores con docker-compose

```
version: '2'
services:
  db:
    image: mysql:latest
    volumes:
      - "./.data/db:/var/lib/mysql"
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    links:
      - db
    ports:
      - "8000:80"
```

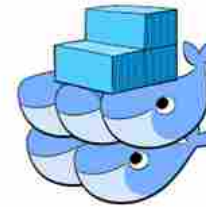
## Nuestro cluster con **Swarm**



# Docker & Wordpress “Juntos pero no revueltos”

✉ hola@juanregueras.es

🐦 @ReguerasJuan



DOCKER SWARM  
Manager  
192.168.10.1



Worker 1  
192.168.10.2



Worker 2  
192.168.10.3

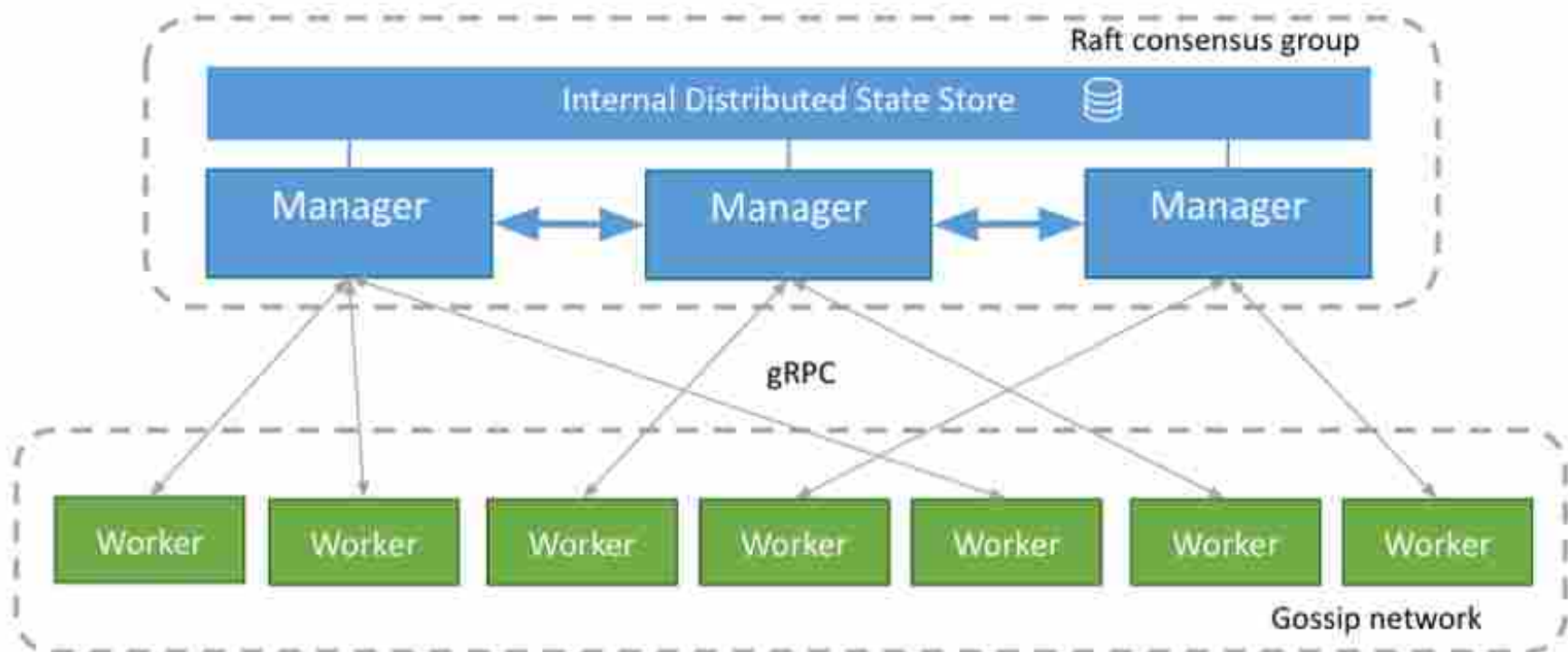


Worker 3  
192.168.10.4

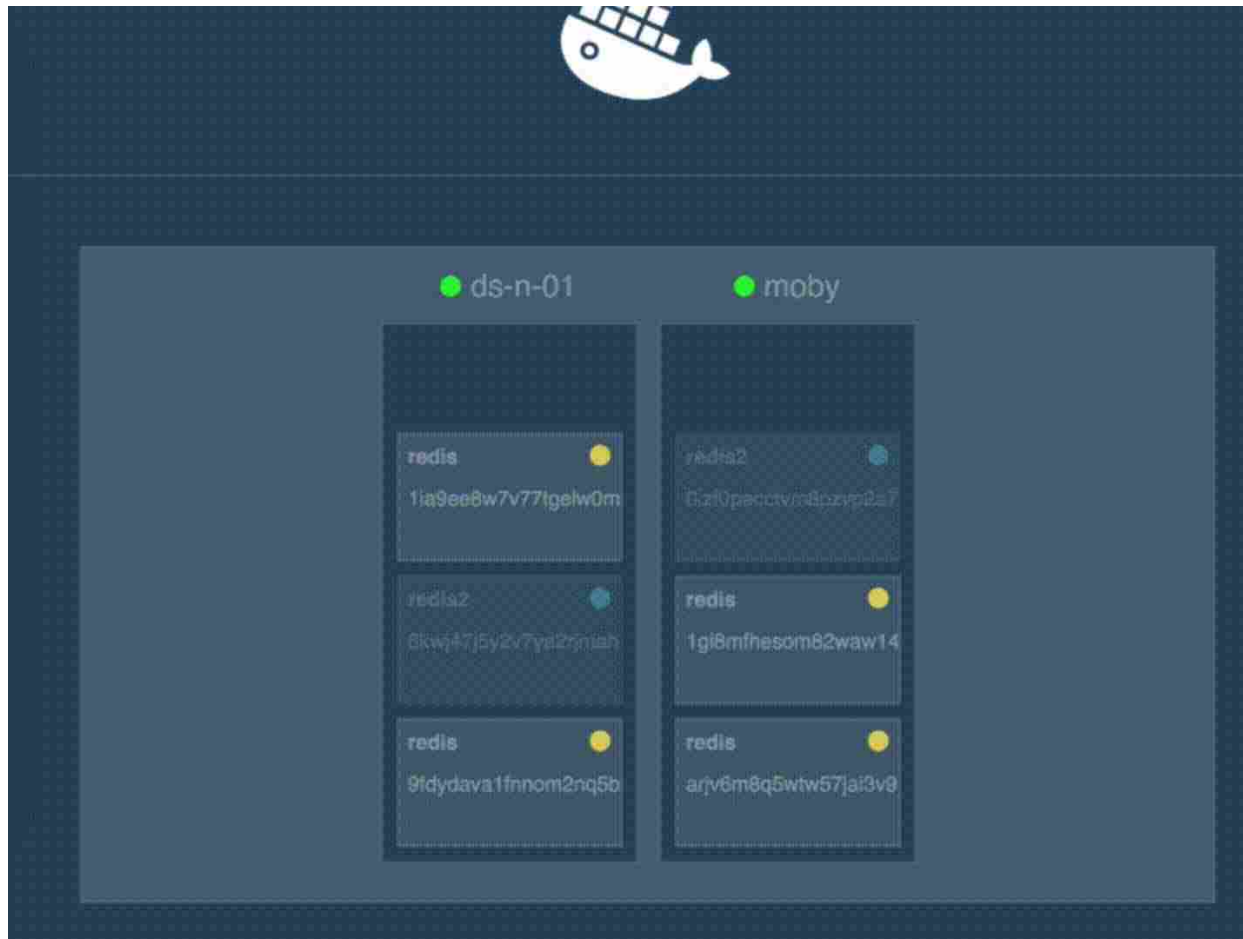
# Docker & Wordpress “Juntos pero no revueltos”

✉ hola@juanregueras.es

🐦 @ReguerasJuan



```
docker run -d -p 8080:8080 -v /var/run/docker.sock:/var/run/docker.sock julienbreux/docker-swarm-gui:latest
```



## Creación de un cluster y agregar nodos

```
$ docker swarm init
```

```
$ docker swarm join --token <id token> <ip manager:2377>
```

## Crear un servicio dentro del cluster

```
$ docker service create --replicas 1 --name web alpine ping docker.com
```

## Ver los servicios que se están ejecutando

```
$ docker service ls
```

## Borrar un servicio

```
$ docker service rm web
```

## Ver los procesos de un determinado servicio

```
$ docker service ps web
```

## Escalar un servicio

```
$ docker service scale helloworld=5
```

## Desplegar un servicio multicontenedor con docker-compose

```
$ docker stack deploy --compose-file docker-compose.yml stack
```

**Aun así ...**  
**¿Queda alguna duda?**